University of Crete

Computer Science Department

AmI Game Floor: A Multimodal Ambient Intelligence Environment for playful learning

by

Haris Papagiannakis

Master's Thesis

Iraklio, February 2011

University of Crete

Computer Science Department

AmI Game Floor: A Multimodal Ambient Intelligence Environment for playful learning

by
Haris Papagiannakis

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Author

Haris Papagiannakis, Computer Science Department

Board of enquiry:

Supervisor

Constantine Stephanidis, Professor

Member

Anthony Savidis, Associate Professor

Member

Antonis Argyros, Associate Professor

Approved by:

Angelos Bilas, Associate Professor
Chairman of the Graduate Studies Committee

Iraklio, February 2011

ΠΕΡΙΛΗΨΗ

Πρόσφατες έρευνες για την εφαρμογή της Πληροφορικής στην εκπαίδευση τονίζουν την ανάγκη για την μαθητοκεντρική μεταστροφή της προς την πλήρη εκμετάλλευση του δυναμικού των εκπαιδευτικών εφαρμογών. Για τον σκοπό αυτό, έχει προταθεί η υιοθέτηση δραστηριοτήτων που διευκολύνουν την μάθηση μέσω της ενεργητικής ενασχόλησης των μαθητών σε ομάδες και της έντονης διαπροσωπικής τους αλληλεπίδρασης σε φυσικά περιβάλλοντα μάθησης.

Η παρούσα μελέτη παρουσιάζει το σχεδιασμό, την ανάπτυξη και αξιολόγηση ενός τεχνολογικού πλαισίου υποστήριξης για εκπαιδευτικές εφαρμογές, με την ονομασία AmI Game Floor. Το σύστημα συνθέτει ένα εκπαιδευτικό περιβάλλον Διάχυτης Νοημοσύνης (AmI) που στοχεύει στη δημιουργία διασκεδαστικών συνθηκών μάθησης εν μέσω αθλοπαιδιών. Συγκεκριμένα, παρουσιάζει μία καινοτομική προσέγγιση ενσωμάτωσης χαμηλού κόστους κιναισθητικής και συνεργατικής τεχνολογίας σε ένα φυσικό περιβάλλον μάθησης, συμπεριλαμβάνοντας τεχνικές μέτρησης αποδόσεως. Παρέχοντας δυνατότητες προσαρμογής για εκπαιδευτικές εφαρμογές σε επίπεδο υποδομής, το AmI Game Floor υποστηρίζει ένα μεγάλο εύρος αντικειμένων και θεματικών περιοχών.

Η υλοποίηση του συστήματος περιλαμβάνει ένα πεδίο δραστηριοτήτων, επαυξημένου με υπολογιστική όραση, υπό τη μορφή ενός πλέγματος θέσεων και ένα σύνολο παραμετροποιήσιμων διεπαφών χρήστη (UI) που προσφέρουν πολυτροπική διάδραση. Συνολικά, το AmI Game Floor παρέχει την απαιτούμενη υποστήριξη σε φυσική υποδομή, δικτύωση, 2D γραφικά, ήχο και χειρισμό (μέσω κιναισθητικής και συνεργατικής διάδρασης) για εφαρμογές εκπαιδευτικού περιεχομένου. Επίσης, υποστηρίζει λειτουργίες προφίλ χρήσης, εναλλαγής στυλ παιχνιδιού και ανάλυσης απόδοσης.

Προκειμένου να δοκιμαστεί και να αξιολογηθεί το AmI Game Floor, αναπτύχθηκε η εφαρμογή "Apple Hunt", η οποία προκαλεί τη αριθμητική σκέψη των (νεαρών) μαθητών μέσω του κιναισθητικού και συνεργατικού παιχνιδιού, υπό την επίβλεψη διακριτικής τεχνολογίας AmI. Προτεινόμενα σχέδια εφαρμογών, όπως η «Ακροστιχίδα», μία παραλλαγή του Scrabble™ και ο «Μαθηματικός Λαβύρινθος», περιγράφονται επίσης με σκοπό να αναδείξουν την ευελιξία του πλαισίου να υποστηρίξει την ανάπτυξη ποικίλων εφαρμογών.

Το "Apple Hunt" αξιολογήθηκε σύμφωνα με ένα συνδυασμό μεθόδων που απευθύνονται σε αξιολογητές μικρής ηλικίας, συμπεριλαμβανομένων των *παιδικών επιτροπών*, που αποδείχθηκαν μία υποσχόμενη προσέγγιση στην αξιολόγηση αλληλεπιδραστικών εφαρμογών με παιδιά. Τα αποτελέσματα της αξιολόγησης επιδεικνύουν το υψηλό δυναμικό του συστήματος να προάγει τη σκέψη και τη διασκέδαση, που πηγάζει από το συνεργατικό και πλήρως κιναισθητικό παιχνίδι των μαθητών. Τέλος, τα συμπεράσματα της μελέτης αναλύονται με γνώμονα τόσο τη σκοπιμότητα της εφαρμογής όσο και του πλαισίου υποστήριξης, ενώ παρουσιάζονται και σχετικές προτάσεις μελλοντικής εξέλιξης.

ABSTRACT

Today's research in computer-based education highlights the need for a learner-centred shift towards harnessing the full potential of learning applications. Knowledge-facilitating activities are suggested towards this achievement, involving the learners' active engagement in groups and intensive interpersonal interaction in real-world contexts.

This thesis reports the design, development and evaluation of a technological framework for learning applications, named AmI Game Floor, which constitutes an educative Ambient Intelligent (AmI) environment, aimed at creating challenging learning conditions through play and entertainment. AmI Game Floor represents an innovative approach of emphasizing low-cost kinesthetic and collaborative technology in a natural playful learning context, while embodying performance measurement techniques. Providing learning applications with infrastructure customization facilities, AmI Game Floor supports a wide breadth of educational subjects and concepts.

The system implements a vision-augmented playfield, shaped as a grid of game positions, and a set of customizable user interfaces (UI) offering multimodal interaction. Overall, the AmI Game Floor framework offers the appropriate physical infrastructure, networking, 2D graphics, sound and control (kinesthetic and collaborative interaction) support for learning applications. It also features profiling, various game modes and performance analysis.

In order to test and assess AmI Game Floor, the "Apple Hunt" application was developed, which engages (young) learners in arithmetic thinking through kinesthetic and collaborative play, overlooked by unobtrusive AmI technology behind the scene. Further, potential designs of the "Word-search game", a variation of Scrabble™ and "Math Labyrinth" are also discussed, so as to uncover the flexibility offered by the framework for accommodating the development of diverse applications.

"Apple Hunt" has been evaluated according to a combination of methodologies suitable for young testers, whereas *children committees* are introduced as a promising approach to evaluation with children. The obtained results demonstrate the system's high potential to encourage thinking and fun, deriving from the learners' full-body kinesthetic play and team work. Finally, the conclusions of this thesis are discussed with respect to both the game's and the framework's objectives, and future work is suggested.

Table of Contents

1. INTRODUCTION

This thesis investigates effective methods of technology-supported learning with respect to the learners' needs for physical activity, social interaction, participation and feedback. In this context, the development of an educational game framework, named AmI Game Floor, is described, that utilizes Ambient Intelligence[1] (AmI) technologies to support the development of playful learning[2] applications designated mainly for children. AmI Game Floor realizes a natural playfield in which players acquire learning by playing, moving, talking, helping or competing with each other. During this process, learners are free to experiment, make mistakes, quit and restart, even jump around or shout when things don't work out as desired.

The framework addresses the main objective of providing technological support in an Ambient Intelligent environment appropriate for accommodating and encouraging learning processes based on contemporary teaching methods (playful learning, learning by participation). Therefore, apart from providing an infrastructure for learning applications, the objectives of AmI Game Floor are to:

- Promote kinesthetic activity;
- Endorse collaboration among learners.

AmI Game Floor consists of an augmented natural playfield, capable of multiple player-tracking on a grid of game positions. A set of customizable graphical user interfaces illustrate the game's virtual world, designed to provide visual feedback, supplemented by a sound direction modality for acoustic feedback. Mobile phone controllers and a game manager complete the framework's infrastructure. Achieving seamless communication among its scattered components, the framework offers physical infrastructure, networking, graphics, sound and control support for applications.

In short, AmI Game Floor:

- Provides an educational framework, generic enough to support learning in various subjects;
- Augments physical play, allowing players to focus on strategy rather than on controls;
- Supplies performance feedback and analysis, oriented towards the learner and the educator accordingly. A performance record is also implemented for future reference;
- Consists of a low-cost vision-based installation;
- Supports unobtrusive multimodal interaction and provides an API for integration of supplementary modalities;
- Allows applications to customize its interaction patterns;
- Offers profile creation, stored in the respective record;
- Provides three different game modes: practice, individual and collaborative game.

[1] Ambient Intelligence refers to technology-equipped environments that are sensitive and responsive to the presence of people. The concept is analyzed in Section 1.2.

[2] Playful learning is a type of learning through play. Related information can be found in Section 1.4.

Applications provide the content of learning and set the logic of interaction, intending to create an atmosphere of *fun*, *challenge*, *engagement* and *learnability*. In order to evaluate the system's potential to facilitate playful learning through kinesthetic activity and collaboration, a prototype of "Apple Hunt" is developed. "Apple Hunt" is a learning application designated for elementary school children, which aims at exercising them in fundamental arithmetic operations. Also, the system's flexibility to accommodate diverse applications is demonstrated through three more application designs. A new evaluation methodology, involving the composition of children committees, is further applied in the evaluation of the Apple Hunt game. Finally, the realization of two evaluation sessions involving 9 children in total is described and its results are discussed in Chapter 6.

The remaining of this Chapter discusses the motivation behind this thesis and analyzes the related background. A brief introduction to the fundamental characteristics of Ambient Intelligence is provided, and the opportunities it offers in the domain of Education are discussed. The Chapter ends by briefly mentioning contemporary learning theories, with an emphasis on children education.

The thesis is structured as follows:

Chapter 2 discusses related work in interconnected areas of research. In particular, it goes through educational AmI environments, interactive floor projects, trends in kinesthetic console games and playful learning applications. The last Section summarizes the features of the referenced works and highlights the novel aspects of the work reported in the thesis. Chapter 3 relates to the design of AmI Game Floor. In particular, it describes the human-centred design methodology adopted, with an emphasis on the educational focus of the framework. In addition, an analysis of the system's features and functionality is provided, followed by an outline of the hardware setup.

Chapter 4 presents the system's implementation. The Chapter is prefaced by a brief analysis of its underlying infrastructure, followed by a detailed description of the software components that comprise the system. The typical software architecture of AmI Game Floor applications is also discussed, composing the image of a wider system. The Chapter closes with a description of the wider system's components communication.

Chapter 5 discusses application designs. Initially, it describes the implementation of "Apple Hunt", an arithmetic game application developed in order to demonstrate the framework potential and evaluate the usability of its applications. Moreover, the designs of other tow games, the "Word-search game", a variation of Scrabble™ and "Math Labyrinth" are also presented, in order to provide more insight in how games of various types can exploit the flexibility supported by the framework.

Chapter 6 describes the evaluation of AmI Game Floor by young testers. The Chapter begins with an introduction about game evaluation compared to the evaluation of other types of applications, as well as the challenges faced when involving children in the evaluation process. Then, the adopted evaluation process and the conducted experiment are described, followed by an overview of the obtained results. Finally, the chapter closes with a discussion of the findings.

Chapter 7 summarizes the conclusions of this thesis and suggests future work with respect to the current implementation.

1.1. Motivation

Information and Communication Technologies (ICT) have proven their ability to induce radical changes in human habits, activities and life. ICT have changed what it means for partners to cooperate, for friends to communicate and for families to obtain goods for living. It is not accidental that generation of today's children is characterized as the "Computer Generation". For this generation, ICT are changing the traditional way to read a book, play a game, or listen to music [13]. Whereas it is evident that ICT change human life and everyday activities, , this change does not always prove to be smooth. Technology has to be designed in such a way as to be respectful of life cycles, society structures, biological needs, ethics and particularities, so as to avoid the alienation of mankind from its natural course.

A fast transition has taken place through the expansion of the (Internet-enabled) Personal Computer in the private, working and educational environment, thus tying down people to a sedentary lifestyle. Despite many parents' inclination to early supply their children with computers for a successful professional future, several studies question the provisioned benefits of this view [9, 61]. In fact, they stress that the effects of prolonged computer use have to be seriously considered when it comes to children, as it may lead to serious consequences for a child's physical development. According to the same studies, hazards like musculoskeletal injuries, visual problems, obesity, social isolation and aggressiveness are only a subset of the wide-range potential effects that children may be faced with, if occupied extensively with computers. Figure 1 exposes a complete list of the related potential hazards.

Emphasizing computers in childhood may expose children to the risk of a broad range of developmental setbacks. Potential hazards include the following:

Physical Hazards
- Musculoskeletal injuries
- Visual strain and myopia
- Obesity and other complications of a sedentary lifestyle
- Possible side effects from toxic emissions and electromagnetic radiation

Emotional and Social Hazards
- Social isolation
- Weakened bonds with teachers
- Lack of self-discipline and self-motivation
- Emotional detachment from community
- Commercial exploitation

Intellectual Hazards
- Lack of creativity
- Stunted imaginations
- Impoverished language and literacy skills
- Poor concentration, attention deficits
- Too little patience for the hard work of learning
- Plagiarism
- Distraction from meaning

Moral Hazards
- Exposure to online violence, pornography, bigotry, and other inappropriate material
- Emphasis on information devoid of ethical and moral context
- Lack of purpose and irresponsibility in seeking and applying knowledge

Figure 1. Potential hazards from computer use in childhood (adapted from [9])

In contrast to physical activity, sedentary life is counterproductive for learning as well. Recent research in the field shows an association between low aerobic fitness and obesity with lower performance at school [59]. Moreover, child development experts emphasize that moving in three-dimensional space stimulates both sensory and intellectual development [9]. "As a child learns to put movements in order, brain areas are primed to put words and ideas into a logical sequence," Healy argues in [25].

While all grades of education are going through a wave of computerization, there has been no clear evidence of the educational benefits of ICT in terms of improvements in the students' performance [9]. An explanation of this fact possibly lies in what specialists call the computer-centred shift, as opposed to the emergence of a student-centred shift of education. According to Roschelle et. al [60], to induce positive effects on learning, technology must support its four fundamental characteristics: (i) active engagement, (ii) participation in groups, (iii) frequent interaction and feedback, and (iv) connections to real-world contexts.

In the above context, serious computer games have a great educational potential for children, as they are capable of combining all four characteristics in an entertaining vehicle for learning. Providing a fun, playful approach games facilitate learning in three ways [45]:

1. Learning as a result of tasks stimulated by the content of the games;
2. Knowledge developed through the content of the game;
3. Skills arising as a result of playing the game.

To date, there have been various praiseworthy examples of educational game software. Nevertheless, examining the interaction patterns used by the majority of them, physical skills development (corresponding to the third point above) is not only neglected, but also in declining course, hiding potential harmful effects. Repetitive long-lasting awkward moves of thumbs on a keyboard, mouse or control pad in a sedentary position is just one factor of the underlying danger.

Consequently, further efforts and experimentation are required towards thorough understanding of technology's negative effects and adopting healthy paradigms. Educational technology, in addition, needs to advance towards more effective models of learning-through-doing, rather than automating marginally effective models of presentational teaching [10]. This thesis explores the development of educational technology that promotes kinesthetic and collaborative activity aimed at creating challenging learning conditions through play and entertainment.

1.2. Ambient Intelligence (AmI)

Ambient intelligence (AmI) can be defined as the embodiment of technology into the built environment, so that people can focus on the activities they perform by using technologies, rather than on how they do it. AmI reflects the vision for ICT in the year of 2010 and beyond, that was conceived in the late 1990s [1, 58]. The notion of *ambience* in AmI corresponds to integration of technology in everyday life in a way that it is rendered unobtrusive during interactions with the environment. The notion *intelligence* reflects that the digital surroundings exhibit specific forms of social interaction [1]. In other words, the environment should be able to recognize inhabitants, adapt to them, learn from their behavior, and occasionally act upon their behalf. This leads to the following characteristics of AmI environments [1].

- *Unobtrusive* functionality embodied into the environment;
- *Context-awareness* through user, location, and situation identification;
- *Personalization* through interface and service adjustment;
- *Adaptation* through learning;
- *Anticipation* through reasoning;

Shifting towards natural, user-friendly and proactive interactions, the AmI paradigm aims at improving the quality of peoples' lives, by dictating technology to adapt to people, rather than forcing people to adapt to technology. Ambient Intelligence is rooted in three key technologies: Ubiquitous Computing, Ubiquitous Communication and Intelligent User Interfaces [2]. *Ubiquitous Computing* refers to the integration of microprocessors into everyday objects like furniture, clothes or toys. *Ubiquitous Communication* enables these objects to communicate with each other and with the user. *Intelligent User Interfaces* enable the inhabitants of AmI environments to control and interact with the environment in a natural (e.g., voice, gestures) and personalized way (preferences, context).

In the EU, research in this field is driven by a series of scenarios scripted on behalf of ISTAG [14] to provide a guide in the development efforts. The scenarios "offer provocative glimpses of futures that can be realized" and regard people as the "forefront of technology". The AmI technologies developed so far integrate, among others, location sensing, face, gesture and speech recognition, as well as speech synthesis, providing a physical means of interacting with them. Examples of related research projects are presented in Section 2.1.

1.3. The potential of Ambient Intelligence in education

Ambient Intelligence has a significant potential to contribute improving the educational environment. Embedding sensing and responsive technology into learning environments helps learners focus on the study subject, rather than on its procedural matters. Imagine the typical routine of a teacher collecting homework in paper-form from 20 students, having to check and evaluate it within the class time or after class. On the other hand, think of a system linking students' homes to school, capable of collecting homework from home and delivering results in class (together with overall statistics). Obviously, the teacher's job becomes far simpler in the latter scenario, allowing the teacher to focus on individual learners' needs instead of giving general advice.

While modern trends in Education include life-long learning or learning-by-doing, learning now builds not only on formal school environment, but also on social interactions, on television or the Internet. The ISTAG Scenarios for Ambient Intelligence [14] include a scenario involving AmI educational technologies, - the Ambient for Social Learning (ASL) [14]. In the scenarios, ASL is defined as a learning environment which serves various ages and knowledge levels, by customizing subjects to the learners' capacity. Besides, the learners are important source of knowledge, as they provide an important input for the learning situations of each other. ASL is a good example of how AmI technologies can potentially support greater empowerment of users, allowing them to follow their own tasks and processes in a learning environment. AmI further contributes by providing instant personalized feedback, resulting from the ability to record and review experiences of past and present participants.

A great ally of AmI in learning environments is educational gaming. In particular, augmented educational games are capable of transforming the conventional classroom into an educational world of physical activity. Varying from “intelligent” building blocks [43] to interactive dragon avatars [29], augmented educational games provide a means of connecting learning with the natural world. This learning approach facilitates the development of physical skills, which play a significant role in the overall learning process [9].

1.4. Contemporary Learning Theories

Learning is generally defined as the lifelong process of transforming information and experience into knowledge, skills, behaviors, and attitudes [74].Learning theories fall under three main categories [16]:

a. **Behaviourism**: According to this theory, learning is manifested by the establishment of new behaviour through conditioning of environmental events, characterized by contiguity and frequent iterance. Consequently, learning is influenced by the frequency of external events which impact education.

b. **Cognitivism**: In opposition to Behaviourism, this theory looks beyond behavior to examine learning. Cognitivism relates learning to human memory and regards the brain as an organized processor of information. The main difference of Cognitivism lies in the perceived locus of control over the learning procedure: the learner, rather than the environment, plays the protagonist role.

c. **Constructivism**: Regards learning as an individual involvement in real-world situations, in which the learner actively constructs knowledge based upon current or past experiences. According to this theory, learning is a personal endeavor performed in practical contexts. Social constructivism further suggests that knowledge can be the result of social interaction and collaborative activities motivated by shared problems or tasks. Moreover, Constructivism sees the educator (teacher) merely as a facilitator between the learner and knowledge, rather than the transmitter of educational content. Teaching evolves from simply imparting knowledge towards creating opportunities for students to learn.

Although it is not a recent theory, Constructivism has widely influenced the contemporary view of learning. Two of its expressions, the Kinesthetic and Collaborative types of learning, are discussed in the following paragraphs with an emphasis on children education.

Kinesthetic learning

Kinesthetic learning is a learning type in which the learner is engaged in physical activity, as opposed to reading a book or following a lecture. In this type of learning, knowledge is the result of a part or the whole body's movement, which stimulates the sensory system. Body movement is a crucial learning factor for children [18], as apart from health matters, it also involves their sensory, physical and social development. Moreover, Moser [48] extends this theory by arguing that playing relates to sociality and this combination of sociable play can, in turn, lead to improvement of speech.

Playful learning, as field experts define kinesthetic learning through play, has received much research attention in the last years. In addition to fun, scientists believe that playful learning encompasses the following core interrelated learning activities [54, 56]:

- Self-motivation
- Exploration through interaction
- Engagement and responsibility
- Reflection
- Imagination, creativity and thinking at different levels of abstraction
- Collaboration

Although little related research has emerged (Section 2.2), technology-supported playful learning seems to bear a great potential to benefit the physical development of children.

Collaborative learning

Collaborative learning (CL) is an umbrella term for a variety of approaches in education that involve joint intellectual effort by students or students and teachers [11]. Collaborative learning refers to methodologies and environments in which learners engage in a common task in which each individual depends on and is accountable to each other. Groups of students work together in searching for understanding, meaning or solutions or in creating an artifact of their learning such as a product. Teachers who use collaborative learning approaches tend to think of themselves less as expert transmitters of knowledge to students, and more as expert designers of intellectual experiences for students - as coaches or mid-wives of a more emergent learning process.

The benefits of CL are extensively discussed in literature. Vygotsky argues that students are capable of performing at higher intellectual levels when asked to work in collaborative situations than when asked to work individually [62]. Group diversity in terms of knowledge and experience contributes positively to the learning process. According to Johnson and Johnson [31], there is persuasive evidence that cooperative teams achieve at higher levels of thought and retain information longer than students who work quietly individually. According to Smith and MacGregor [73], CL has more to offer on educational goals by:

- Learner involvement
- Community sense
- Civic responsibility

Appropriate use of computers can have an impact in promoting collaborative activity among children. Höysniemi et. al argue that, since “children naturally gather in groups, especially to play games, collaborative computer technologies are a thriving new field in children’s computer use and focus on teaching new social skills to children” [29]. Benford et. al [6] complement that despite the underlying potential, “neither learning nor collaboration will occur simply because two students share the same computer. Numerous factors must be addressed”. Various related factors have been discussed in Section 1.1.

In conclusion, while the learner’s and the educator’s roles are actively evolving within the information society, the possibilities offered by ICT towards playful learning can play an important role in the development of children. AmI Game Floor is the result of experimentation through a platform for this kind of activities, appropriate for contemporary methods of teaching as well as for individual experimental learning through play.

2. RELATED WORK

The essence of AmI Game Floor is related to a handful of different domains, thus constituting an interdisciplinary concept. The proposed framework is rooted in Ambient Intelligence technologies to create a contemporary learning environment. Comprising an interactive playfield for kinesthetic play, it is governed by human-computer interaction principles specialized for children. Figure 2 illustrates a concept map of AmI Game Floor, illustrating its relations to various domains.

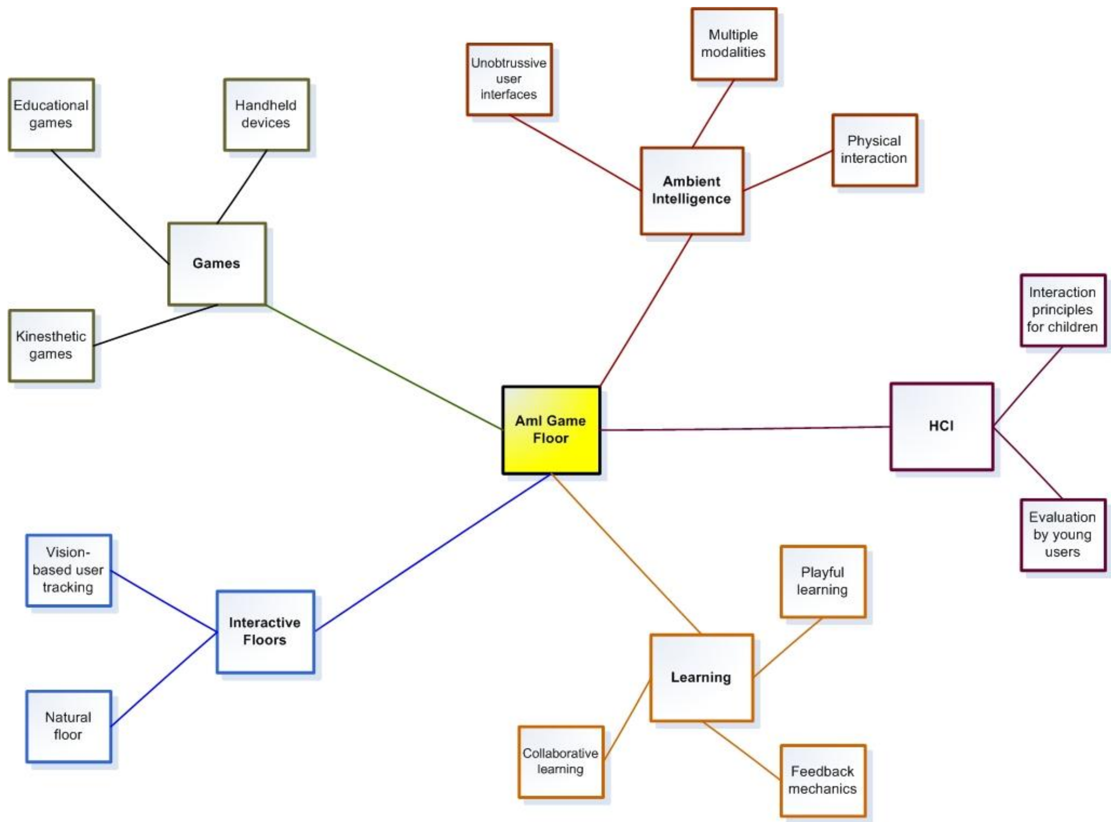


Figure 2. Concept map of the AmI Game Floor

This Chapter discusses related work from interconnected areas of research. In particular, it goes through educational AmI environments, interactive floor projects, trends in kinesthetic console games and playful learning applications. The last Section summarizes the features of the referenced works and highlights the novel aspects of the work described in this thesis.

2.1. Educational AmI Environments

Recently, there has been a demand for developing tools to support education and entertainment in a seamless manner. As pointed out in Section 1.1, the digital essence of such applications is not adequate to induce their successful implementation in learning environments. On the contrary, a great advantage of AmI applications is their potential to facilitate non-skilled learners and educators in any type of learning or teaching. This renders AmI applications appealing to learning environments. The following paragraphs refer to research achievements in this field.

Karime et al. [33] propose an ambient edutainment system for very young children. Based on communication and multimedia technologies, the system provides learning and entertaining capabilities for children between the ages of 1 and 4. One of the important features it presents is the transformation of the children's narrative activities into visual representations. Based on a microphone, used to recognize the children's spoken utterances, and a camera used to sense the speaker's mood, input flows trigger visual representations of related images gathered and filtered through the Web (Figure 3). Children, supervised or unsupervised, have both the capability of improving their cognitive skills and learning about new objects and entities through the use of visual presentations provided by the proposed system. Experimental results obtained show that the system creates a joyful environment for the children during their experiments.

Figure 3. A multimedia-driven AmI edutainment system for young children [33]. A visual response to a child's utterance of the word "Apple"

COHIBIT [49] is an edutainment exhibit (oriented for theme parks) in an ambient intelligence environment. The visitors can use instrumented 3D puzzle pieces to assemble a car. The key idea of this edutainment framework is that all actions of a visitor are tracked and commented by two "life-like" guides (Figure 4). Visitors get the feeling that the anthropomorphic characters observe, follow and understand their actions and provide guidance and motivation for them. While the visitor can fully engage in the manipulation of tangible, real world (car) pieces, the agents remain in the background, commenting on visitor activities, assisting where necessary, explaining interesting issues of car technology at suitable moments and motivating the visitor if s/he pauses for too long. In summary, the system's mixed-reality installation provides a tangible, (via the graspable car pieces), multimodal, (via the coordinated speech, gestures and body language of the virtual character team) and immersive (via the large-size projection of the "life-like" characters) experience for a single visitor or a group of visitors.

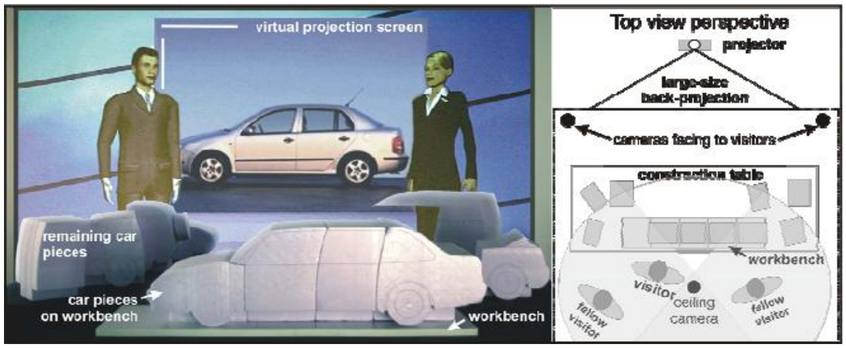


Figure 4. Overview of COHIBIT [49]. The system integrates a tangible user interface interacting with embodied agents on screen

Marti and Lund [43] explore the possibilities of activating the children's physical (body) and functional (brain) structure through engaging into a tangible programming activity. They argue that their "programming by building" scheme enables non-skilled everyday people to program. Rectangular LEGO DUPLO bricks (Figure 5, called "I-BLOCKS", are augmented by sensory technology, so as to be used in building tangible structures with various functionalities. Each building block contains serial two-way connections and receives energy power from a battery building block via connectors in the corners on its bottom. I-BLOCKS are divided into input and output building blocks. Input building blocks may include LDR sensors, IR sensors, microphones, switches, potentiometer, whereas output building blocks may include servo motor, DC motor, IR emitter, LEDs, sound generator, etc.

Figure 5. I-BLOCKS, augmented building blocks for programming through a tangible user interface. © H. H. Lund, 2002.

2.2. Playful Learning Applications

Playful Learning activities (Section 2.2) are acknowledged to offer engagement, self-motivation and entertainment during the learning process. Playful learning systems, depending on their implementation, might constitute a superset of educative AmI environments. Their main distinctive characteristic is the game approach they base on. Recent research trends in this topic are briefly overviewed in the following paragraphs.

The "hunting of the Snark" [54] is a playful application involving physical artifacts that are electronically augmented and enhanced to trigger various digital events to happen. Its designers envisioned an adventure game, where pairs of children have to discover as much as they can about a virtual imaginary creature called the Snark, through collaboratively interacting with a suite of tangible objects. They argue that this kind of tangible interaction leads children to a form of natural learning. Snark uses a room-based interface with handheld devices and has also been played outdoors. In general, the binding is predominantly static, the game uses spatial and relational interactivity, and the embodiment appears to be local and environmental.

The game contains several activities. The first requires the children to find some hidden invisible clues in a physical room using a handheld device. The device is a Jornada PocketPC interfaced with an ultrasonic indoor positioning sensor and an electronic compass, and is used to identify hidden virtual objects responding to physical tokens (embedded with RFID tags). The physical tokens are then combined with the proper artifacts, triggering digital animations and sounds. As soon as the clues are collected, the children are offered more forms of interaction with Snark. For example, they can move their feet in various combinations (in a constrained space) to make Snark appear, or they can flap their arms as if flying—again to change the Snark's behaviour. While interacting with Snark the children have to capture as much of the Snark's behaviour, morphology and personality using a special device called a Snarkcam (Figure 6). At the end of the adventure the children can download what they have captured and also revisit all the places where they encountered Snark. The underlying idea is to trigger reflections about the novel forms of causality and also to get the children to articulate what it meant to them to engage in playful learning. Evaluation of the game regarding the children's excitement, engagement, reflection, creativity and collaboration yielded successful results [54].

Figure 6. The "hunting of the snark" combines multiple interaction techniques in a playful learning context

"Qui Qui's giant bounce" [26] is a game providing an immersive and physically engaging alternative to traditional computer games by making use of computer vision and hearing technology. The game's perceptive user interface is wireless and does not require any contact with input devices during play. In particular, it involves a fire-dragon which mimics the user's movements and breathes fire when the user shouts. The game works on a PC computer equipped with practically any low cost microphone and webcam. It is targeted for uncontrolled real-life environments such as homes and schools and is oriented towards 4 to 9 year old children. The dragon avatar activates children into using their bodies and thus develops their physical abilities, such as coordination skills, spatial recognition and balance. Although this system was mainly designed for testing collaborative design with children, its designers conclude in that vision-based intuitive user interfaces that activate the children's whole body are beneficial for their physical development [29].

2.3. Interactive Floors

Interactive floor surfaces, firstly introduced as dance platforms, have emerged to more sophisticated versions, some of which are learning-oriented. Regardless of their purpose, interactive floors demonstrate a potential to promote kinesthetic interaction and collaboration. The prototypes described below are sensor intensive environments for tracking of people's movements of feet and fall under two main categories: Sensor-based and Vision-based interactive floors. Sensor-based interactive floors are mostly used as dance settlements. Apart from the floor-sensors paradigm, the Magic Carpet [52] enhances sensitivity by Doppler-radars to perform tracking of the upper body and arms movements. On the other hand, vision-based interactive floors present more customized interactivity advantage, offering a variety of possible activities.

iGameFloor [21] is an interactive floor platform ideal for educational applications involving body-kinesthetic learning. Its concept lies in vision-based limb tracking on an interactive floor – a 12 m^2 glass surface with bottom projection. iGameFloor is oriented towards schools' leisure environments, in which students engage in sociable activities and play. As its designers argue, "kinesthetic interaction is fun and motivating thus encourages children to explore and learn". In iGameFloor, kinesthetic interaction on a large display surface supports collaborative, co-located play and learning through communication and negotiation among the participants. iGameFloor supports three types of applications: collaborative games, Kknowledge sharing applications and simulations, which provide a framework for a variety of learning concepts.

iGameFloor is built into the physical floor of a school department square (Figure 7). It consists of a 3 m deep hole covered with a projection surface. The projection surface is 3x4 m glass of approximately 9 cm thickness divided into four tiles. The projection is created by 4 projectors with a resolution of 1024x768 pixels. They are placed vertically, each one covering their tile of glass. Each projector is associated with a Web cam tracking limb contact points on the tile covered by the given projector. The system uses two separate PC, one analyzing the vision input and one managing the graphics distribution among the projectors. It also supports sound through ceiling-mounted loudspeakers and a nearby-placed subwoofer.

Figure 7. Left: Action on iGameFloor. Right: The system's hardware setup

BodyGames [41] is a prototypical interactive floor consisting of a set of tangible tiles for physically activating children in their play. Each tangible tile functions as a building block by containing processing power, sensors, actuators, and communication capabilities. This component design allows the flexible settlement of the blocks in various shapes (Figure 8), which accommodates a wide variety of applications, such as “ping pong”, “hopscotch” and “colour race”. Two prototype iterations of the tangible tiles have been tested. Their first implementation has a soft surface and each measures 40 x 40 cm^2. Inside each tile, there is a force sensitive resistor and the actuation consists of 9 red LEDs and 9 blue LEDs distributed equally on the tile in a 3x3 matrix. With this simple tile, it is possible for example to switch from blue to red or from red to blue every time someone jumps the tile. The second version is a smaller building block, measuring 21 x 21 cm^2, and the actuation consist of one light source supporting 8 different colours. Further, the second version of the tile is made out of rubber and integrates sound output. The tangible tiles are initially utilised in 2D on the ground, but are also extended with wireless handheld units in order to develop activities where children can interact with virtual and/or physical elements in 3D.

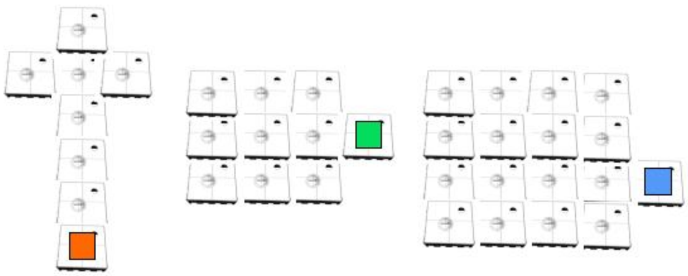


Figure 8. Implementing various playfield settlements with BodyGames' building blocks

iFloor [36] leads the notion of interactive floors towards community interaction between collocated people. Installed in a library environment, iFloor proposes the use of vision-based user tracking for browsing and selecting content as a collaborative process, during which mobile phones can be used for posting messages onto the floor. It is governed by the aim to facilitate a space for communication and collaboration, as well as to experiment with collaboration on interactive floor surfaces with no need of special input devices apart from being present in the physical library. The system consists of a remote server for receiving and handling sms and emails and administering questions and answers. Furthermore, a projector mounted on the ceiling is connected to a local computer for the display on the floor (Figure 9). Bright daylight visibility problems are bypassed by using a powerful projector to project the graphic interface onto thin white PVC boards on the floor. The floor interaction works on the basis of a video tracking system software analysing the rim of the interface based on a video feed from a web-cam mounted on the ceiling. The tracking of people’s position is translated into magnetic forces attracting the cursor. Furthermore, the users’ physical proximity to the floor display is translated and used to orient the interface accordingly. The system’s tracking precision is enough to keep track of 15 people at one time in a 5 x 4 m^2 rectangle.

Figure 9. Using iFloor

Litefoot [20] is an interactive floor space that tracks dancers' steps, and converts the steps into auditory and visual display. The system can also record steps, for further analysis for use in dance research programs, choreographic experimentation and training. The PodoBoard, a sensor-based dance floor, was developed to facilitate the extension of the use of clackage, a form of seated dance that normally takes place on a small wooden floor. The PodoBoard provides an accurate set of coordinates for the position of feet contact with the floor, which is a matrix of 1 inch square aluminium tiles. The reported reaction time of the floor to footsteps is very fast. However, the system is critically dependent upon good electrical contact between shoe and floor, as it works through using the shoes to complete an electrical circuit – the shoes have metal contacts at toes and heel.

Another related project is the Magic Carpet [52], a dance floor on which the position and pressure of the performer's feet are measured together with upper-body and hand motion to create a musical environment, where any kind of body motion is directly and immediately converted into expressive sound. A 16 x 32 grid of piezoelectric wires, running across the carpet at a roughly 4" inter-wire pitch, is used to sense foot pressure and position. Whenever a peak in pressure is detected, the processor sends out a MIDI Note-On event, with the note number corresponding to the particular wire generating the data, accompanied by a 7-bit pressure value sent as the note velocity. Corresponding Note-Off events are sent when the pressure value from a formerly active wire decays back to the baseline. Further, the system uses two Doppler radars for tracking the movement of the arms and upper body.

Doppler-shifted reflections from a performer moving within the beam return to the antenna, where they are mixed with the transmitted signal in a hot carrier diode. This produces beat frequencies in the range of 0-5 kilohertz (kHz) that directly represent the performer's dynamic state (the frequency is a function of velocity, and the beat amplitude is a combined function of the size and distance of the reflecting object). The sensor hardware is connected to a PC running music software written in Visual C++. The music generated by the system consists of a low voice, a middle voice, and a high voice. Finally, the direction of movement detected by the radar units controls the chord on which everything is played. The overall, combined effect of the entire installation is a relaxing soundscape that responds to subtle movements on the part of the performer. The sound mappings allow for a good deal of expression, yet they are intuitive and simple enough for players to appreciate the connection between their movements and the sound produced.

The Lumetila project [40] aims at developing a natural user interface where the user uses body movements to control a computer game. Their designs describe a floor sensor system, a real time 3D graphics engine and special effects devices for creating the envisioned immersion and experience in the Virtual Space, animated in an orthogonal display. The room of the Virtual Space is equipped with a 3D sound system and light effect devices. Supporting multi-player game, the system does not depend upon any virtual reality input devices, like data glasses or gloves, which induces immersion and easier collaboration among the players.

The Lambent Reactive [34] is an audiovisual environment that bases upon an interactive illuminated floor (Figure 10). By interfacing a network of pressure sensitive, light-emitting tiles with a 7.1 channel speaker system and requisite audio software, many avenues for collaborative expression are offered. By giving users light and sound cues that both guide and respond to their movement, the Lambent Reactive creates a rich environment that playfully integrates the auditory, the visual, and the kinetic sense into a unified interactive experience.

Figure 10. The Lambent Reactive interactive floor responding to a user's movement

The hardware system for the Lambent Reactive consists of a grid of 36 networked 16”x16” square tiles and a computer simultaneously running Ableton Live, Native Instruments’ Battery, and proprietary Lightspace software. Each tile of the reactive floor surface is composed of 16 individually addressable 4”x4” pixels, and is capable of transmitting pressure data from four sensors embedded in the intersections between the central and outer layers of pixels. In this way, individual position and pressure can be detected down to a resolution of two pixels, or 8”, less than the length of all but a child’s foot. The system converts each tile’s sensor data into MIDI signals, which are routed to the audio software to trigger the sound sources. A timing system is also built to enable linear sequencing of visual and audio events, as well as rhythmic detection and recording of dance steps. Since the system’s introduction, several applications have been developed, demonstrating the system’s flexibility to accommodate a wide range of game requirements, not limited to dance-oriented applications.

2.4. Kinesthetic Console games

Kinesthetic console games represent a more natural way of combining technology with games. Since the last decade, gaming companies have started to invest in this concept, because of its potential to attract even people skeptical against technology. However, kinesthetic console games are not a very recent idea, as in 1998 Konami firstly introduced the Dance Dance Revolution™ (DDR) game [78], a music video game series. Since its introduction, DDR has been the pioneering series of the rhythm and dance genre in video games, and has been the inspiration for many variations of this kind. DDR innovation at that time lied in that it was one of the first games to involve player motion in the gameplay. To interact with the game, players stand on a stage and hit colored arrows, laid out in a cross, with their feet, responding to musical and visual cues. Players are judged by how well they time their dance to the patterns presented to them on screen and are allowed to choose among a variety of music songs to play to if they receive a passing score.

Figure 11. Original DDR coin-op machine

The dance stage (Figure 11) is divided into 9 sections, 4 of which are in the cardinal directions and contain pressure sensors for the detection of steps. During normal gameplay, arrows scroll upwards from the bottom of the screen and pass over a set of stationary arrows near the top (referred to as the "guide arrows" or "receptors", officially known as the Step Zone). When the scrolling arrows overlap the stationary ones, the player must step on the corresponding arrows on the dance platform. Dancing feedback is also shown on screen, where players are given a judgement for the accuracy of their dancing steps. Longer green and yellow arrows referred to as "freeze arrows" must be held down for their entire length, either producing a "O.K." if successful, or a "N.G." (no good) if not. Successfully hitting the arrows in time with the music fills the "Dance Gauge", or life bar, while failure to do so drains it. If the Dance Gauge is fully depleted during gameplay, the player fails the song, usually resulting in a game over. Hoysniemi [28] in a large scale international survey shows that DDR play has a positive effect on the players' physical and social lives. Additionally, DDR "creates a setting where new friends can be found".

Wii™ [76] is a home video game console released by Nintendo on November 19, 2006. What initially discriminated Wii from its rival consoles (Xbox, Playstation 2) is its wireless controller, the Wii Remote, which is capable of detecting three-dimensional movement and can be used as a pointing device, thanks to its embedded accelerometer and infrared camera. The controller connects to the console using Bluetooth and features rumble, as well as an internal speaker. This design allows users to control the game using physical gestures and traditional button presses. Originally, Wii was oriented commercially towards sports, fitness and fighting games. Nevertheless, research has demonstrated additional remarkable uses of it. WiiArts [39] is a software utilizing Wii Remote as a tool for creating collaborative expressions of art, whereas projects like [3, 8] report its successful use as an input device for 3D interaction in virtual reality environments. Furthermore, the latter study takes advantage of its potential for rehabilitation of people with physical disabilities.

Figure 12. Kinesthetic interaction using Wii peripherals. Left: Wiiarts [39], Right: Rehabilitation system [3]

In addition, Wii is supported by the Balance Board™, an orthogonal board used to measure the users' center of balance by its integrated pressure sensors. The Balance Board's development was tightly coupled with the *Wii Fit* game, designed for fitness activities led by graphical character on screen. Moreover, Haan et al. [22] suggest its use as a 3-degrees-of-freedom input device to implement 3D rotation, navigation and general control techniques. Conclusively, Wii peripherals prove to be governed by good potential for kinesthetic interaction. Although their functionality exercises mainly the upper-body, full-body motion can be achieved by their integration in a VR environment [8].

Eye™ [75] is a hardware extension designed for Playstation 3™. Its look may be similar to a webcam, though its use differs in that it allows players to interact with games using motion, color detection and also sound, through a built-in microphone. Eye's functionality bases on computer vision and gesture recognition to offer natural representation of user motion in the digital world generated by the Playstation device.

The PlayStation Eye is capable of capturing standard video with frame rates of 60 hertz at a 640×480 pixel resolution, and 120 hertz at 320×240 pixels. The camera features a two-setting adjustable fixed focus zoom lens. Selected manually by rotating the lens barrel, the PlayStation Eye can be set to a 56° field of view for close-up framing in chat applications, or a 75° field of view for long shot framing in interactive physical gaming applications.

Playstation Eye is the descendant of EyeToy™, which has been widely studied in relation to rehabilitation of disabled people through motion, despite its basic orientation towards gaming. Rand et al. [55] investigate the use of EyeToy for the rehabilitation of older adults with disabilities. Their results proved its potential as a means to promote exercise for high-functioning individuals with stroke. EyeToy has also been tested for play-therapy with success by Brooks et al. [7].

Kinect™ [77] is a newborn extension device of Xbox 360, which abolishes the need for a controller during play. Particularly, it enables users to control and interact with the Xbox 360 through a natural user interface using gestures and spoken commands, so that full-body motion replaces clicks and strokes. It mainly differs from the Playstation Eye in that it is capable of sensing depth without the need of any extra controller, like the Playstation Move™ (a wireless handheld controller) required in the latter.

The device features an RGB camera, depth sensor and multi-array microphone running proprietary software to provide full-body 3D motion capture, facial, gesture and voice recognition capabilities. Apart from Xbox gaming, Kinect demonstrates a clear potential for kinesthetic interaction of maximum two users with computer technologies. However, its efficiency compared to the Playstation Eye (formerly described) is yet to be evaluated.

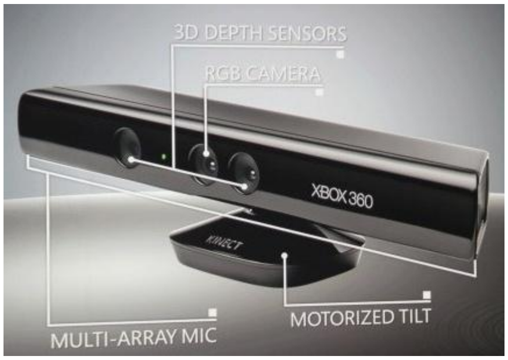


Figure 13. Technologies in Kinect

2.5. Discussion

Following the overview of related work in the previous sections, and in particular of the areas of Educational AmI environments, Playful Learning systems, Interactive Floors and Kinesthetic Gaming, this Section discusses the limitations and drawbacks of current approaches, and highlights how these are addressed in the context of AmI Game Floor. Table 1 summarizes the related projects' features and categorizes them according to their purpose. For simplicity reasons, all projects described in Section 2.3 are labelled as "console applications". Moreover, the "Qui Qui" project under the "Playful Learning" category (Section 2.2) is omitted from this analysis, due to its very specific focus, whereas the remainder project of the category (the *Karime et al. AmI educational system [33]*) is classified under the "Educational AmI environments" category.

The remaining projects are hence divided into educational AmI environments, interactive floors and kinesthetic console games. Educative AmI environments aim at providing education through real-world context activities and are characterized by a learner-centred design, while the majority of them implements Tangible User Interfaces (TUI) as means of transmitting knowledge. In these efforts, therefore, kinesthetic interaction is considerably static and corresponds mainly to hand-manipulation of objects. The "Snark" project [54] represents an exception in this category, by implementing a plethora of user interfaces activating a rich set of the learner senses. Nevertheless, the physical interaction achieved is limited to specified moves in a relatively constrained space, whereas interactivity is focused on the use of handheld devices and augmented objects. Moreover, none of the educational AmI environments defines a system of performance metrics supporting learning-quality evaluation in an empirical way. .

Interactive floors are basically characterized by a high potential of full-body kinesthetic interaction and present diverging objectives closely connected to *fun*. In particular, interactive floors can be categorized into game surfaces [21, 41, 40], socializing media [36] and, literally, dance floors [52, 34, 20]. The framework presented in this thesis would rather fit in the first category, although it does not build on top of any specialized electrical surface, in contrast to most related projects, which impose an immediate contact between users and the sensing technology. Further analysis unveils categorization by the type of tracking technology adapted, which is mostly represented by floor sensors. Vision-based [21] or radar-based [52] systems appear to allow for greater freedom of movement and control, but provide less tracking accuracy.

Regardless of their purpose, the majority of the interactive floors is not utilized for learning purposes, despite its relative potential. iGameFloor [21] –consists the only exception, providing a dynamic learning framework, which also combines full-body kinesthetic interaction based on limb-tracking. For this purpose, iGameFloor implements an expensive projection system with limited control facilities, as it is solely bases on multi-touch input from the users' limbs.

Kinesthetic console applications are characterized by higher interactivity capabilities than interactive floors, due to the plethora of input techniques they implement. Their utilization has widely focused on fun, in addition to various prominent applications intended for rehabilitation purposes. Yet, none of their uses so far builds on real-world contexts or presents educative content.

AmI Game Floor, presented in this thesis, covers the aforementioned limitations and drawbacks through a different approach towards promoting entertainment and learning. AmI Game Floor provides a technology-augmented natural environment, which encourages full-body kinesthetic and collaborative learning. While the system combines intense physical and multimodal interactivity, its installation is considerably low-cost, based mainly on computer vision, with the only additional hardware devices required being a visual display and mobile phones. Providing a playful infrastructure for learning applications, it is suitable for a wide variety of educational subjects and concepts, determined by the content design of each application. Furthermore, AmI Game Floor embodies an innovative measurement system for learning quality, using a special set of performance metrics. Performance reports are stored in the file system, and can be used as points of reference on the long run to assess the quality of acquired learning of a stable players' sample.

		Features					
	Project	**Collaborative environment**	**Learning orientation**	**Kinesthetic interaction**	**Multimodal interface**	**Technology**	
						Input	**Output**
Educational AmI environments	*Snark*	x	x	x (TUI ++)	x	Vision, Handheld devices, RFID	Graphics, Sound
	Karime et al. AmI educational system		x			Vision, speech recognition	Sound Display
	COHIBIT		x	x (TUI)	x	Vision, Tangible pieces (with RFID tags)	Sound, Graphics (with embodied agents)
	I-BLOCKS	x	x	x (TUI)		Sensors, Switches	IR, light, sound
Interactive floors	*iGameFloor*[3]	x	x	x		Vision	Sound, Graphics
	BodyGames			x		Floor sensors	Lights, Sound
	Lumetila	x		x		Floor sensors	3D Graphics, sound
	iFloor		x		x	Vision, Handheld devices	Graphics
	LiteFoot			x		Floor sensors	Sound, Graphics
	Magic Carpet			x	x	Floor sensors, Doppler radars	Sound
	Lambert Reactive	x		x		Floor sensors	Light and sound cues
Kinesthetic Console Games	*Console applications*	x		x	x	Vision, sensors, handheld devices	Sound, Graphics
	AmI Game Floor	**X**	**x**	**x**	**x**	**Vision, handheld devices**	**Graphics, Sound, Excel reports**

Table 1. Summary of related studies in contrast to the current study's features.

[3] iGameFloor is categorized under Game surfaces, due to its specialized game surface. Alternatively, it could also be considered as part of the first category (educative AmI projects).

3. AmI GAME FLOOR DESIGN

AmI Game Floor is targeted to the development of educational technology that encourages kinesthetic and collaborative activity and creates challenging learning conditions through play and entertainment. Towards synthesizing the image of an effective contemporary educative environment, the learning theories analyzed in Section 1.4 constituted a useful basis. According to the work of Roschelle et al. [60], *learning*, particularly in childhood, is enhanced through the learners' (i) active engagement, (ii) participation in groups, (iii) frequent interaction and feedback, and (iv) connections to real-world contexts. In addition, kinesthetic play represents an attractive context for embedding educational concepts, which aids the absorption of knowledge by stimulating the learner's motor memory [17].

AmI Game Floor addresses the main objective of providing technological support in an Ambient Intelligent environment appropriate for accommodating and encouraging learning processes based on contemporary teaching methods (playful learning, learning by participation). Towards creating a challenging learning environment, the system provides a technological infrastructure for learning applications which:

- Promotes kinesthetic activity;
- Promotes collaboration among learners.

In order to provide the required infrastructure, the system builds on Ambient Intelligence technologies to compose a playful environment, in which participants engage in learning by physical play. In this context, technology overlooks activity unobtrusively to provide personalized feedback and performance analysis, whereas AmI Game Floor applications provide the content of learning and set the logic of interaction. Such user experiences, in which the learner's whole sensory system participates, are believed to foster learning efficiently [18], as opposed to the sedentary play style of typical computer games.

Kinesthetic activity arises through natural interaction with the system's playfield, which is supported by vision-based tracking software [66]. In particular, the tracking software is capable of multiple users' localization, allowing for interaction through movement around the game positions. Inheriting this feature, the system a priori supports collaborative activity. During play in the *collaborative game* mode, individual scores are summed per team to output the winning partners. Also, depending on the application design, input from different players' moves can be combined in various ways to yield collaborative results. Finally, a different type of collaboration is examined in the evaluation of the system, described in Chapter 6, which involves team work of the players acting in the playfield with their partner(s) performing supportive tasks in the surroundings.

This Chapter deals with the design features of AmI Game Floor. The next section describes the requirements set and the followed design process. Further, the Chapter analyses the framework's functionality as embedded in its various components.

3.1. Design methodology

In the design of AmI Game Floor user-centred design approach [70] was followed, involving several field-experts at various stages of the design process.

The starting point for requirements elicitation was the study of A. Druin et al. [12], which correlates children requirements from technology with multiple control, social interactions and expressivity. The process was based on brainstorming sessions Appendix G contains a detailed description of the system's initial requirements, whereas a summary of the final requirements recorded can be found in Table 2 and Table 3, presenting pupil (player) and teacher (educator/facilitator) requirements respectively. At the end of this stage, pupil requirements were considered for the overall framework design, whereas teacher requirements mostly served the design of reporting and statistics presentation.

Pupils' Requirements		Design implications
Needs	a. Entertainment b. Stimulus c. Concentration d. Good learning conditions e. Easy and multiple control[4] f. Social experiences[1] g. Expressivity[1] h. "Cool" accessories[5] i. Good-looking features[2] j. Richness of multimedia[2]	1. Play music & sound effects; 2. Consider rich colors and lights; 3. Enforce motives by score and prizes; 4. Provide multi-sensory experience; 5. Promote communication 6. Design graphics for children.
Age range	4-7, 7-11, 11-15, 15-18 * * *The four age stages of children, according to Piaget* [53]. Content design has to consider the respective cognitive levels of children.	7. The equipment must be suitable for children's age; 8. Place components in the right height, distance etc.; 9. Design for s/w versioning depending on the age of the target group.
Gender	Any	10. Allow customization of controls and graphics to accommodate various tastes
Experience (with technology)	Various	11. Design for intuitive and attractive use
Frequency of use	High	12. Supply robust components

Table 2. Pupil Requirements

[4] Classified as "what kids want in technology" by A. Druin [xx].
[5] Classified as "what kids notice about technology" by A. Druin [xx].

Teachers' Requirements		Design implications
Needs	a. Participation stimulus b. Good teaching conditions c. Clear overview of the pupils activities in class d. Cheating detection e. Flexible environment to support method switching f. Unobtrusive system operations	1. Report on pupils' personal performance 2. Take care of room conditions 3. Design for infrastructure flexibility
Age range	25-70	4. Make the system intuitive in order to appeal even in old-style teaching
Experience (with technology)	Various	5. Provide hints and help

Table 3. Teacher Requirements

Following the requirements phase, Wizard of Oz prototyping proved insightful in designing interaction patterns, as well as in composing a vocabulary (Appendix A) intended to be used in the future for extending interaction through speech recognition. Gameplay simulations worked as a guide for narrowing the target-groups scope to elementary school pupils and foreseeing necessary features prior to development, such as the need for a remote game manager or multiple views of the game activity. Moreover, application scenarios were discussed during brainstorming sessions, involving personnel experienced with children psychology. Finally, two evaluation sessions were conducted involving 9 children aged from 7 to 10 years old (second to sixth class of elementary school, according to the Greek educational system). Chapter 6 describes the evaluation procedure and discusses its results.

Experiments were conducted to choose vision sensors for player-tracking, as various technologies were put to suitability tests. Foot-tracking was tested - using embedded RFID tags in shoes -, but this option was rejected due to low-fidelity results. On the other hand, the Nintendo Wii Remote (controller) [38] was also tested for its ability to allow position tracking relative to its infrared light sensor. Though, its recognition scope proved too short for the needs of AmI Game Floor (about 1 m^2 from a 3 m height). Therefore, vision sensors were selected as a mid-cost solution, capable of covering a sufficiently large area unobtrusively and with high accuracy.

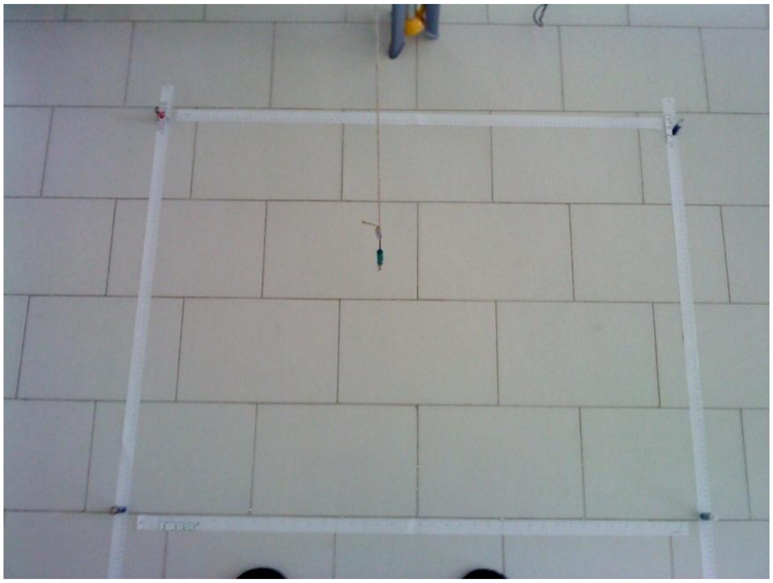


Figure 14. Wii Remote recognition scope at 3 m height.

In conclusion, AmI Game Floor has been influenced both by technological advancements in the field of gaming and contemporary studies on computer-supported children education [21, 27, 29, 35, 37, 56, 69, 77]. It constitutes a game framework that aims to provide an AmI infrastructure for learning applications, fostering kinesthetic interactivity and collaboration among learners.

3.2. Hardware Setup

The system is installed in a 6 x 6 x 2.5 m3 room, in which a 4.88 x 1.83 m^2 dual back-projection display is located at the wall opposite to its entrance. The display is implemented by two bright (3000lm) 1024 x 768 short-throw projectors and a projection screen, providing shape to the main part of the system's virtual world.

In the room, a computer vision system [65, 66] is used for tracking purposes. The vision system includes two computers and eight cameras (Dragonfly, Point Gray Research), which overlook the scene from the ceiling, obtaining synchronized images. The cameras are synchronized by timestamp-based software that utilizes a dedicated FireWire bus across computers and guarantees a maximum of 125 msec temporal discrepancy in images with the same timestamp. The cameras are connected and evenly distributed to the available computers. The software on the computers acquires the incoming synchronous groups of images and an estimate of the locations of the visitors is computed for each such group. The computation is distributed across computers and parallelized on their GPUs. Taking advantage of the multiple viewing angles, the vision system offers non-invasive multi-user tracking, as no additional garment and devices need to be carried or worn.

Additionally, there is an information kiosk, used to host the Remote Game Manager (Section 3.3.3), and several Wi-Fi enabled mobile phones. Mobile phones, connected to the controller webpage, are used as

command controllers (Section 3.3.4). Further, sound is produced by a surround speaker system of 12 speakers installed on the ceiling. The speaker system is capable of localizing sound in relevance to player location.
The hardware setup described above is illustrated in the following Figure.

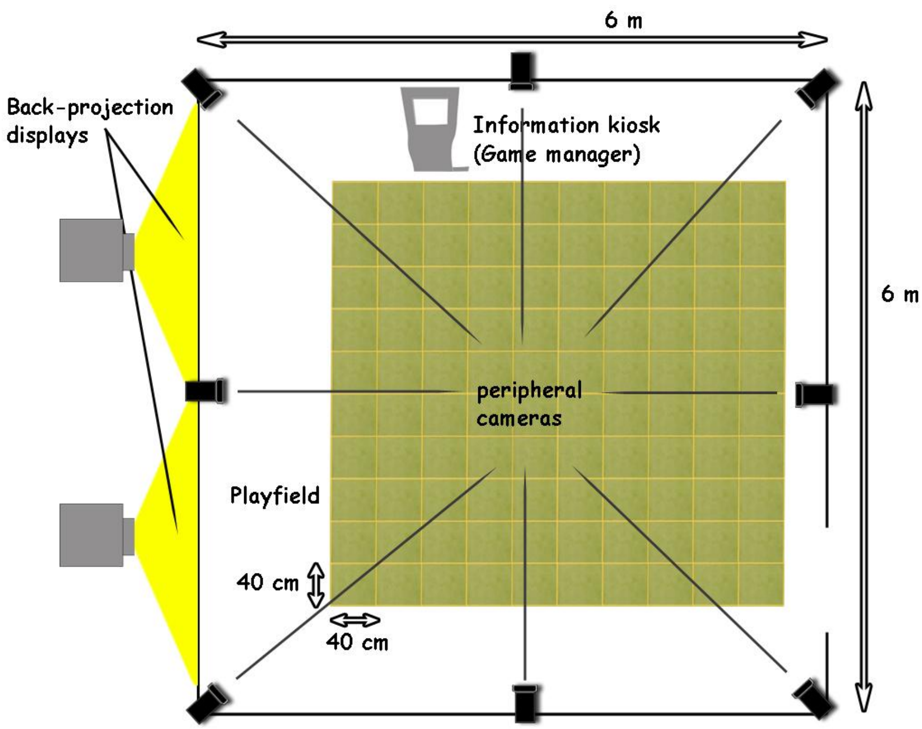


Figure 15. AmI Game Floor's Hardware setup

3.3. Ambient Game Floor overview

AmI Game Floor is composed of the following scattering components:

1. A vision-augmented playfield, capable of tracking the players' moves;
2. Various graphical user interfaces that illustrate game action dynamically from different views on a dual back-projection display;
3. A game manager used as a general remote controller, handled by a touch-screen;
4. A couple of controller interfaces, accessed by mobile phones, for profile creation and command submission during the game;
5. A sound direction facility.

Achieving seamless communication, the components offer physical infrastructure, networking, 2D graphics, sound and control support for applications. In details, the following figure lists the facilities offered by the AmI Game Floor framework.

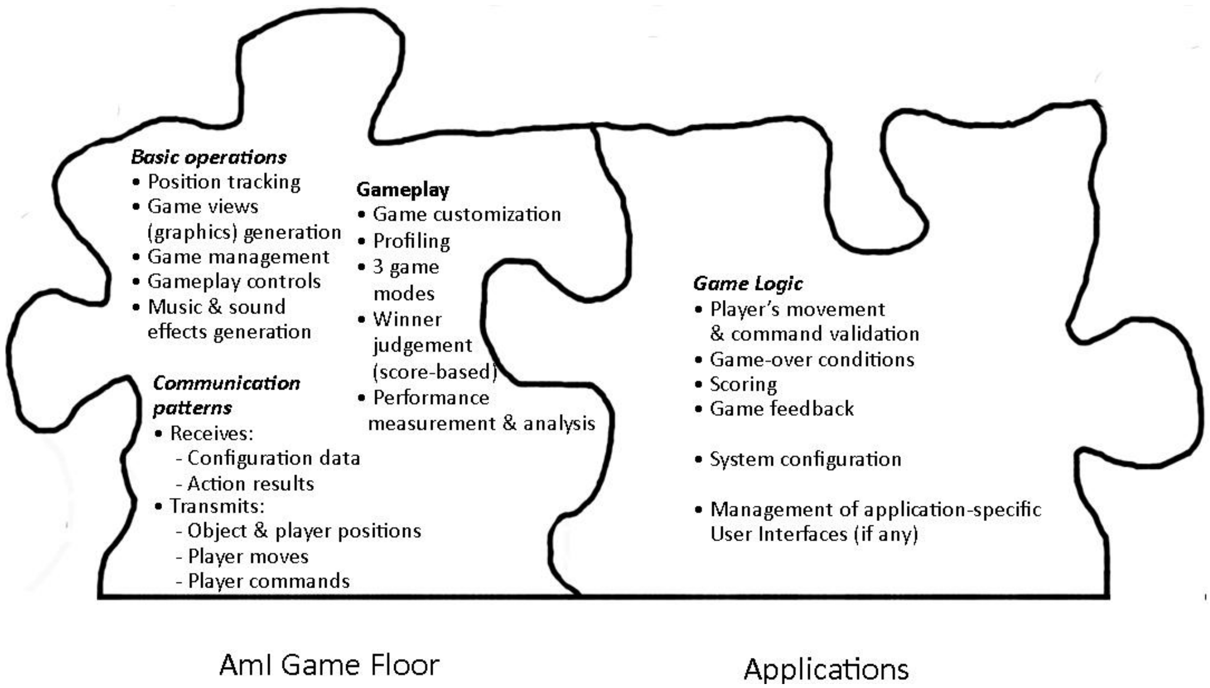


Figure 16. Facilities distribution

3.3.1. Playfield

The playfield consists of a 4 x 4 m^2 thick carpet, shaped to form a grid of game positions (Figure 17). Several equally-distanced plastic stripes, firmed with clips, portray the square boundaries of the positions, which may entail (stickers bearing) marks. Above the playfield, eight cameras scan the game floor for player motion, dispatching tracking information every 167 milliseconds (see Section 3.2 for more details). During the game, a move is sensed as a positioning outside the bounds of the current position.

Figure 17.The positions grid shaped on a 4 x 4 m^2 carpet.

Position marks are determined by the game type, which may vary from arithmetic or alphabetic to custom. In particular, arithmetic games will use numbers, whereas alphabetic games will use letters as identifiers. Extra flexibility is offered by the custom game type that allows any combination among symbols, numbers and characters. Adopting a unified marking practice, the framework defines two potential locations for hosting marks, the centre and the upper-left corner of a position. As shown in Table 4, the central location is considered as default, whereas both locations are marked in case a combination of characters or/ and symbols is required.

The framework is capable of generating any kind of grid styles, enabling applications to choose from empty to sequenced or randomly-marked positions within the playfield, relevant to their game type. Examples of grid styles can be found in the following Table.

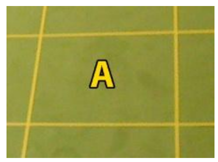
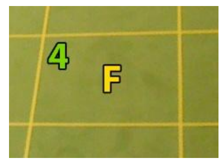
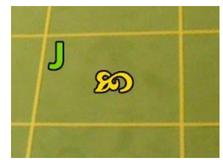
Grid style				
Game type	Arithmetic	Alphabetic	Custom	

Table 4. Grid styles related to game types

Apart from identifying game positions, position marks play a significant role in that they represent some sort of player choice information. For example, if a player decides to move to position "D", the player might be trying to construct a word containing the specified mark. More information about the use of position marks for the synthesis of learning content can be found in Chapter 5.

The playfield can also contain targets or/and obstacles, aimed at enriching the game strategy, as well as creating levels of complexity. Any small objects (dummies) can be used for their physical representation, as their presence does not interfere with the operation of the cameras. An example of the playfield barring physical objects representing targets and obstacles can be seen in Figure 35 (Section 5.1.1). In addition, depending on the game application scenario, rival-players may also be considered as targets or obstacles.

3.3.2. Game Views (Graphics)

A couple of graphical user interfaces, projected in parallel on a dual display, convey the image of the game's virtual world. The GUIs are designed to dynamically depict the game action, in order to aid in the players' decisions and activity.

Illustrating different views of the game at the same time, the GUIs are aimed mainly at providing visual feedback during all phases. The first view is focused on personalized information, whereas the second view depicts mainly the playfield activity. The following Table contains a brief description of each view per game phase.

Phase	*View 1 (personal view)*	*View 2 (playfield view)*
1. Login	Player log-in activity	Selected game description
2. Floor preparation	Starting positions	Starting positions
3. Game action	Player performance	Virtual playfield (activity)
4. End of game	Results (& report)	Selected game description

Table 5. Views per game phase

The GUIs are described by game phase below:

a. Login Phase

During the login phase, the participating players are illustrated in the first display, represented by their avatars (chosen during registration – see Section 3.3.4).

Figure 18. Log-in activity - View 1

At the same time, the second display shows information related to the selected game application, such as rules and hints. The following Figure corresponds to a welcome screen for the "Apple Hunt" game[6].

[6] "Apple Hunt" is an AmI Game Floor game application described in Chapter 5.

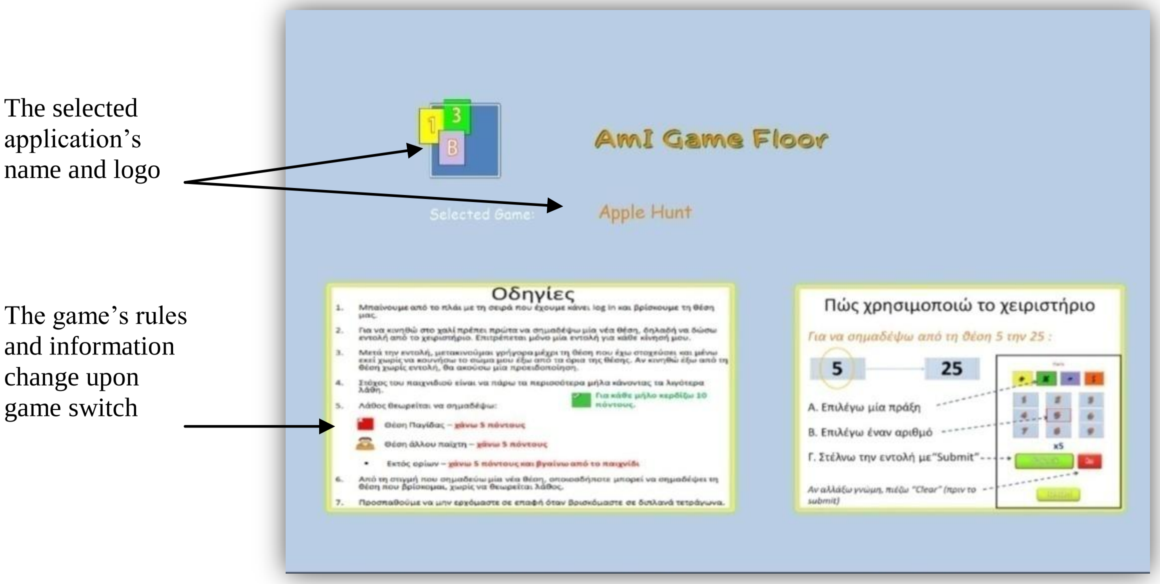


Figure 19. Welcome screen (with game descriptions) - View 2

b. Playfield Preparation Phase

Upon login, both displays show a playfield overview, used to inform the players about their starting positions, as well as the positions of targets and obstacles (if any). This view is identical to the arrangement of the physical playfield and flexible enough to vary among the different game types. Depending on the game type, each position may be marked with a central and/or a upper-left side identifier, a target, an obstacle and/or an avatar.

At this point in time, the players undertake to prepare the playfield by placing the objects required on the playfield and then occupy their positions. As soon as this procedure is over, the game is ready to begin. The following Figure illustrates an example of the displays' look during this phase. Apart from color, shapes indicating the element on the playfield are used to accommodate color-blind players and intuitively demonstrate their purpose.

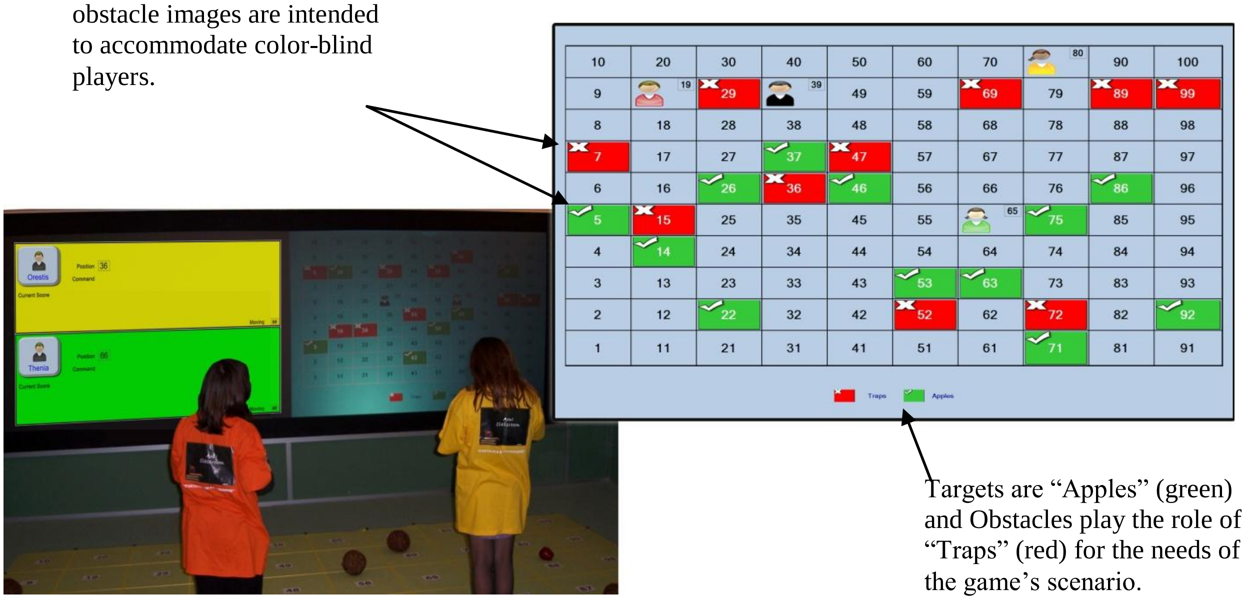


Figure 20. The Virtual Playfield - View 2

c. Action Phase

During the game, the first display splits into an amount of slots equal to the number of participants (Figure 20). Each slot contains information regarding a player's activity, such as score, positioning, target or obstacle hits (if any). Some space is also reserved for game messages (Figure 21, upper-left slot) and command data, which are customizable according to the needs of each game. Furthermore, avatar and slot background color correspondence is established to enforce quick detection during play. At the same time, the playfield overview on the second display depicts dynamically all actions occurring in the game.

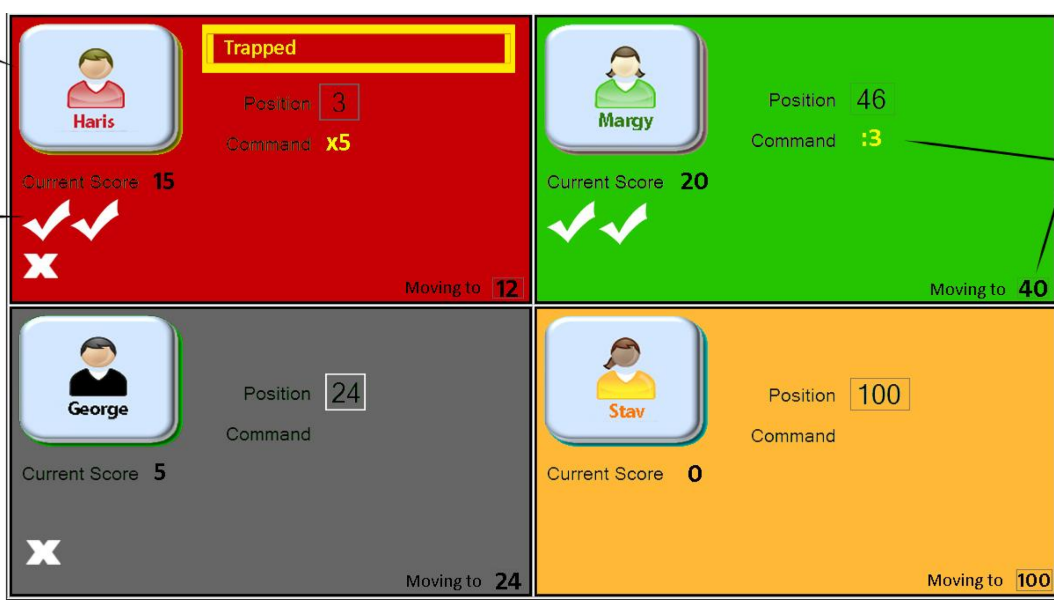


Figure 21. The Player-performance view, shown in the first display

d. Game-over Phase

The end of the game triggers the Results-view (Figure 22), shown in the first display, which contains a summary of the game activity. Apart from informing about the winner(s), the Results-view also entails performance messages (summary statistics) presented in playful mood, by making use of funny expressions and images. For example, animal images appear to describe the "player speed" measure, instead of numbers or percentages.

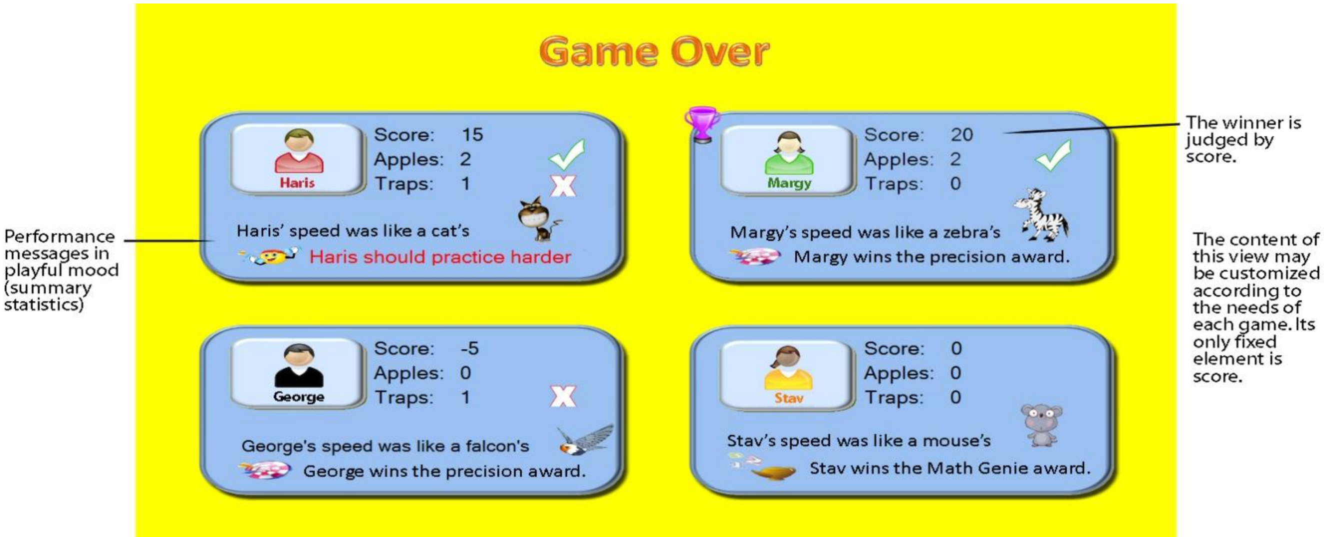


Figure 22. The Results-view displayed at the first display

Finally, an analysis of the game activity can be displayed by activating the Performance Report, generated by the system upon the end of the game. Performance Reports aim at helping the teacher advice low-performers during a game or during a series of games. They can also help the players follow up their performance from game to game. A detailed description and an instance of Performance Reports are illustrated in Section 4.2.4 and Appendix C respectively.

3.3.3. Remote Game Manager

The Remote Game Manager controller is a menu designed exclusively for managing game options in a uniform and convenient way. The menu options are divided into:

- Game selection
- Mode selection
- Level selection
- Sound preferences
- Game management

It should be noted that the framework only determines the Mode and Management options; whereas all the others are dynamically customized according to the selected application's features (see Section 4.3

for more information). The game modes supported vary among Practice, Individual and Collaborative game. In particular, Practice allows one-player's free game, whereas individual game supports maximum capacity single-player game. Collaborative game creates two teams by default and uses the sums of each team's individual scores as winning criteria.

The Management options, which are only available upon the login phase, are summarized in the following Table, whereas the component's GUI is presented in Figure 23.

Phase	Management Options
1. Login	Play (Generate random positions)
2. Floor preparation	Ready (Start the game)
3. Game action	Force game over
4. End of game	View report Start new game Restart (keeping the same configurations) Exit

Table 6. Game management options per phase

Figure 23. The Remote Game Manager interface

3.3.4. Mobile Controller

Mobile controllers are supplied to participants for creating a profile prior to the game (Figure 24) and submitting commands during play. Profile creation is as simple as typing a username and a password and choosing an avatar. Upon registration, the player profiles are stored in order to be recoverable in subsequent games. Profile creation can be even faster, if the "Guest" option is selected, where profiles are temporarily stored for a game session only.

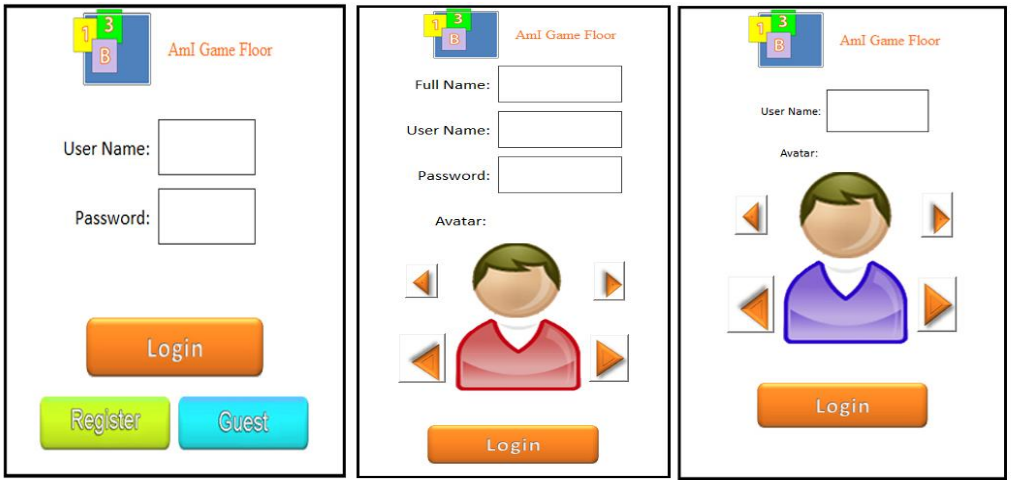


Figure 24. Profile creation - Left to right: Log-in view, Register view, Guest view

The login redirects to a controller interface relevant to the game type of the selected application. In particular, the controller interfaces are divided into the Calculator view, designed to accommodate command submission for arithmetic games, and the Dictionary view, appropriate for both the alphabetic and the custom game types. Using the Calculator controller, the players are enabled to submit commands in the form of numbers or arithmetic operations, whereas the Dictionary controller offers an interface for submitting letters, words or custom symbols, making use of the mobile phone's keypad. Commands are meant to be handled by each application's logic and therefore have no specific result. The following Figure depicts both the views described above. Detailed descriptions of the use of controllers can be found in Section 5.

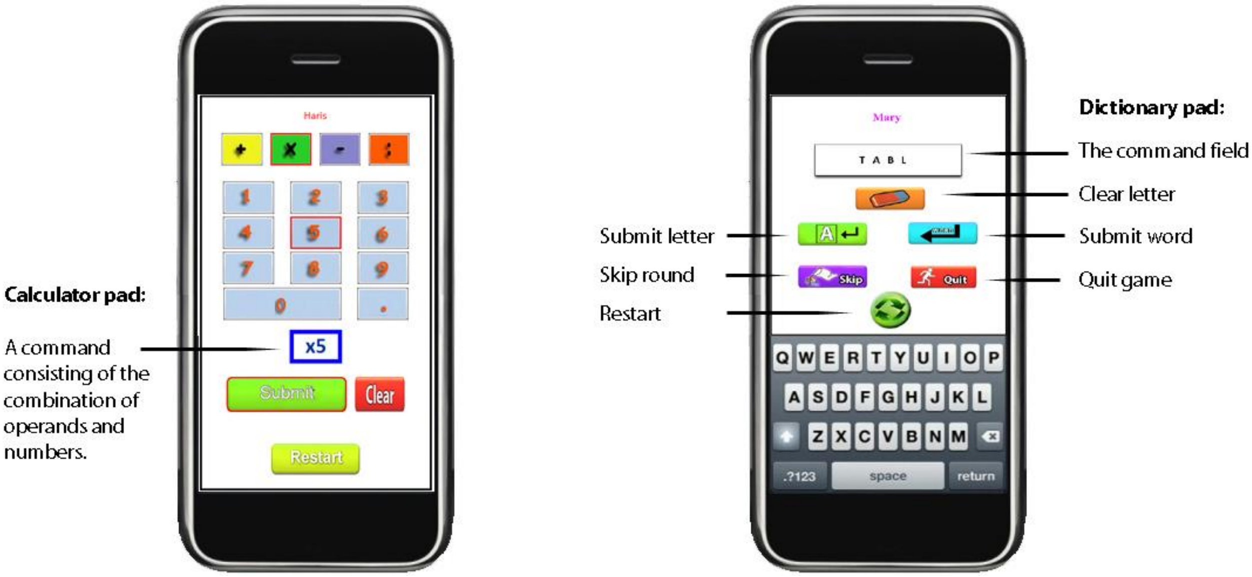


Figure 25. Command Submission - Left to right: The Calculator and Dictionary views

Mobile controller interfaces were developed in the form of locally hosted web pages. TCP/IP communication was preferred in order to provide platform independence, thus taking advantage of the expansion of Wi-Fi enabled mobile devices. Finally, the controller web pages were designed to record the user's clicks on the Calculator and Dictionary pads for usability evaluation purposes (see Section 6 for more information). Upon the end of the game, each controller transmits a *Clicks Log* to the system, which stores it in the file system in .txt format, classified by username. Instances of *Clicks Logs* can be found in Appendix D.

3.3.5. Sound

Speech synthesis is used to enrich interaction in a natural way. Using the Microsoft Speech API, the system is capable of transmitting voice messages to the players during game events, thus rendering visual interaction incidental. Moreover, with the aid of a sound direction modality [70], it can also transmit multiple directed messages, oriented towards the players' locations. Hence, sound source proximity helps the players sense increased message personalization.

Sound effects can additionally be activated to cover routine (repetitive) messages, whereas background music is also available, in order to promote a feeling of joy and vigilance.

4. SOFTWARE ARCHITECTURE

This Chapter describes the implementation of AmI Game Floor. A brief analysis of its underlying infrastructure prefaces the Chapter, followed by a detailed description of the AmI Game Floor architecture and software components. Figure 26 illustrates the software architecture of AmI Game Floor, including the components of a hypothetical AmI Game Floor application. Their functionality and interoperation is analyzed in the following Sections.

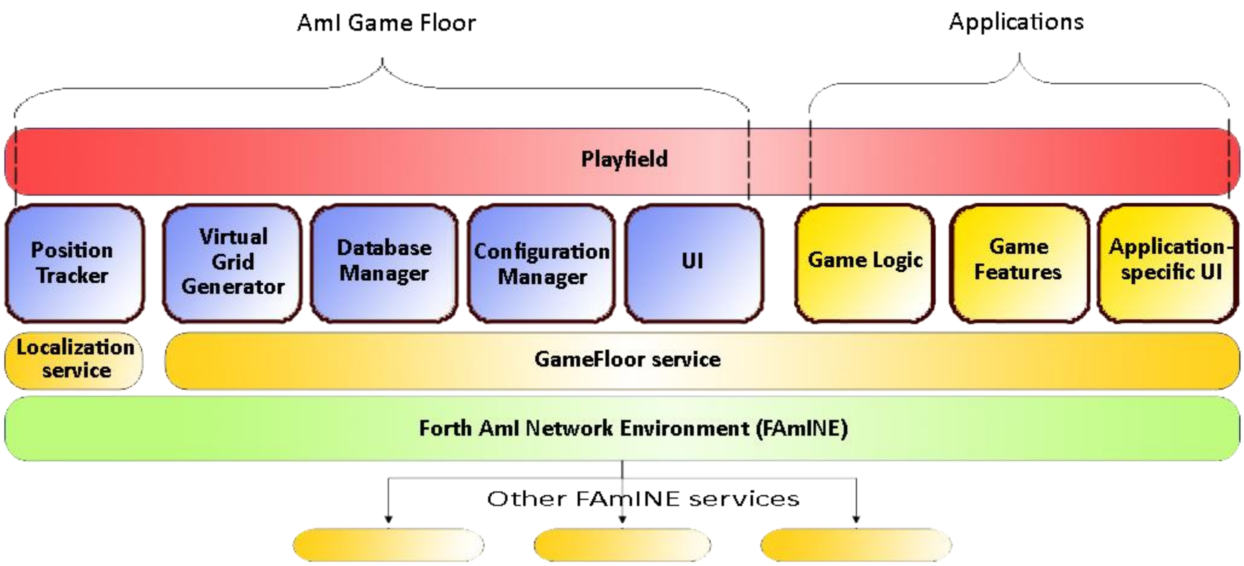


Figure 26. AmI Game Floor software architecture

4.1. Underlying infrastructure

AmI Game Floor has been developed and integrated in an area dedicated to Ambient Intelligence (AmI) research and development at the Institute of Computer Science (ICS) of the Foundation for Research and Technology – Hellas (FORTH). The environment comprises various devices, such as computers, display monitors and projectors, as well as various types of sensors, such as cameras and antennas. Sensors are used to monitor the environment and forward relevant information for other devices to process or display it. This infrastructure is supported by an internally developed middleware layer [19] based on CORBA, which offers intercommunication facilities among the low-level services that control and monitor the various hardware components and the interactive applications. The middleware layer provides libraries and tools to enable software developers to create services with a true Object-Oriented Application Programming Interface (API). These services can be developed and used from any program written in any of the supported programming languages, which include C++, .NET languages, Java, Python, and ActionScript. The middleware allows services to be distributed across the network, hiding the details of network connections and data serialization from the programmers. The software units that comprise this AmI environment can be seen in Figure 27. AmI Game Floor uses the libraries and tools provided by the middleware in order to obtain references to remote services and invoke the methods that are part of their API.

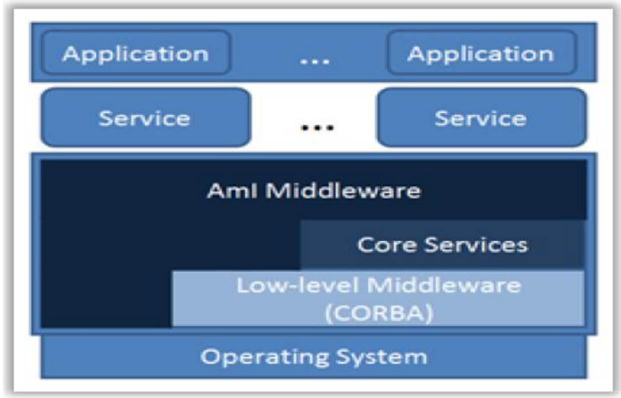


Figure 27. The AmI Middleware [19] architecture

4.2. AmI Game Floor Components

In the framework's environment, interactions are realized by the confluence of several interconnected computing systems based on the AmI Middleware (see previous Section). The technology utilized to bring these distributed units in communication, as well as their operations, are described below.

Orchestrating the system's components, the GameFloor service provides an API for data interchange. Any component, regardless if it is part of the framework or part of an application uses the GameFloor service in order to receive and/or transmit game information. Moreover, the framework utilizes two more services, the Localization and SoundPlayer services, responsible for user tracking and sound generation respectively. In particular, the Localization service is responsible for transmitting user coordinates within a specified context, generated by the vision system (Section 3.2). In the same context, the Sound Player service undertakes to direct sound to the closest speaker related to the user coordinates.

The .Net framework (mainly C#) was used for the development of the components, mainly because of interoperability and performance reasons, associated with the underlying infrastructure and the absence of complex graphics. In order to provide platform-independent means of controlling the game, mobile controller interfaces were developed in ASP.NET, supported by JavaScript and AJAX. Sound interactivity was based on the Windows Media Player API and the Microsoft Speech API (version 5.3)[7] in order to enhance the game environment with sound effects and natural dialogs. A MySQL database, cooperating with the components via the MySQL connector[8], was used to keep a record of game events and player actions. The database also contributes to the publishing of Performance Reports produced in Excel format, with the aid of the Microsoft Office SDK. Finally, to provide a universal means of describing

[7] Microsoft Speech API (SAPI) 5.3. http://msdn.microsoft.com/en-us/library/ms723627(VS.85).aspx
[8] MySQL Developer zone. http://dev.mysql.com/downloads/connector/net/

game configurations, the Features Schema (Appendix B) was composed, using XML Schema[9]. The Features Schema works as a guide of the relative system's configuration required by each AmI Game Floor application.

4.2.1. Configuration Manager

The Configuration Manager works as a "gate keeper" for the framework, as it is responsible for the correct binding with any application requesting to use AmI Game Floor. The Configuration Manager undertakes the following operations:

a. *Application Features Validation*
To use AmI Game Floor, an application will first try to introduce itself by committing its description (i.e., features composed in XML format). The GameFloor Service supports this communication between the framework and the application. Subsequently, a validation against the Features Schema is triggered (Appendix B).

b. *Keeping a Features record*
Since the application is successfully registered, its features are stored for as long as the system operates. This way, the user may switch among a variety of registered game applications at any time.

c. *Parsing Features to customize the game environment*.
At the beginning of every new game, the Configuration Manager will parse the document corresponding to the game selected, in order to activate only the corresponding framework's user interfaces selected by the application. A description of the customizable framework's components can be found also in Appendix B.

4.2.2. Virtual Grid Generator

Following the document parsing, the Virtual Grid Generator utilizes the extracted information in order to construct a Virtual Grid which is sized according to the game specifications. The Virtual Grid is implemented as a jagged array of Positions, i.e., static data structures with the same potential attributes as a physical position, namely a central mark, a side mark, an avatar (identifier) and a target or an obstacle.

Whenever applicable, the Virtual Grid is also depicted by a Graphical User Interface, as described in Section 3.3.2(b). Again, the marks can be comprised of numbers, letters or symbols, according to the selected application's game type. In its visualized form, a Virtual Grid is identical to the physical grid.

The Virtual Grid Generator is responsible for the following operations:

[9] W3C XML Schema. http://www.w3.org/XML/Schema

a. Virtually fragmenting the floor space

Such fragmentation forms a grid of marked positions according the requirements of the selected application. The following Figure visualizes the fragmentation for the sequenced numerical grid style (Section 3.3.1).

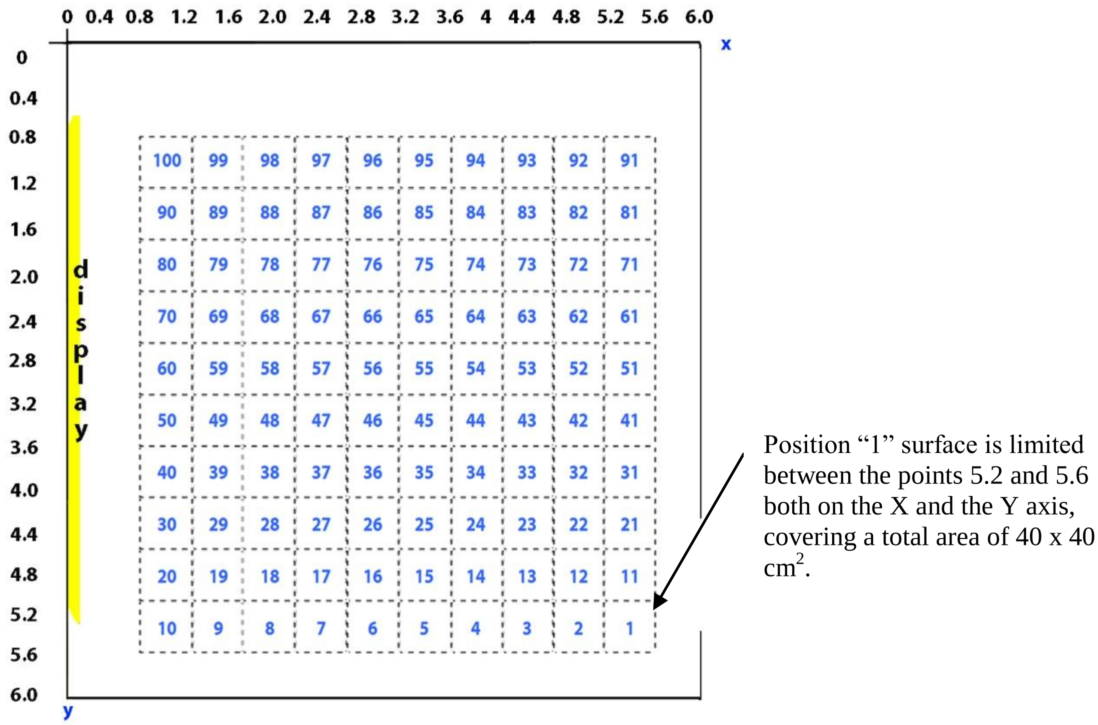


Figure 28. An example of playfield fragmentation according to the sequenced numerical grid style. (The submarginal numbers correspond to playfield coordinates, whereas the central numbers represent game positions)

b. *Orientating positions*

Game positions are oriented towards the system's display. Using a sequenced numerical grid for example, this would mean that the numbers' (marks') sequence increases towards the display. This component is capable of dynamically rearranging the virtual grid given a desired orientation.

c. *Generating Player (starting) positions*

The algorithm used to generate the player random starting positions is described by the following flow diagram. Note that positions might not be generated randomly if an application specifies the starting positions (e.g., Math Labyrinth – Section 5.2.3).

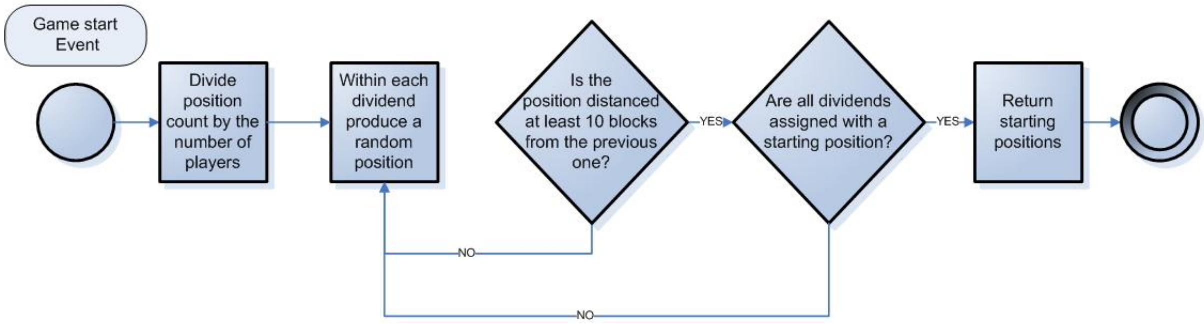


Figure 29. Setting the player starting positions

d. *Generating Target and/or Obstacle positions*

The algorithm used to generate the object positions is described by the following flow diagram. Note that positions might not be generated randomly if an application dictates so, according to the Features Schema (Appendix B).

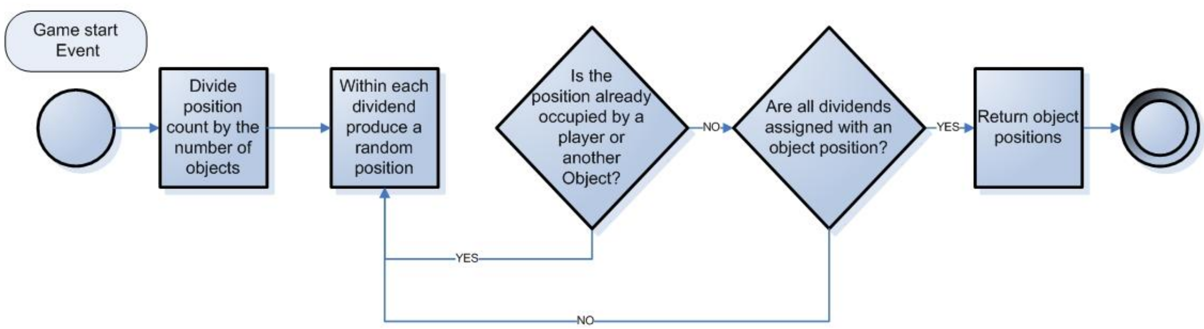


Figure 30. Setting the object positions

4.2.3. Position Tracker

Receiving input from the vision system, the Localization service informs the Position Tracker of the exact location (in terms of floor coordinates) of the persons present in the AmI environment, triggering a tracking event every 125 milliseconds. The Position Tracker, combining input from the Virtual Grid (Section 4.2.2), decides which players have moved since the last event and updates their current positions with the new ones. The Position Tracker also discriminates between players and observers, by keeping track of the playfield's boundaries. According to its function, a new position is registered only upon a player's movement and permanence inside the limits of a different position for at least one second. The algorithm used by the Position Tracker is illustrated in the following flow diagram.

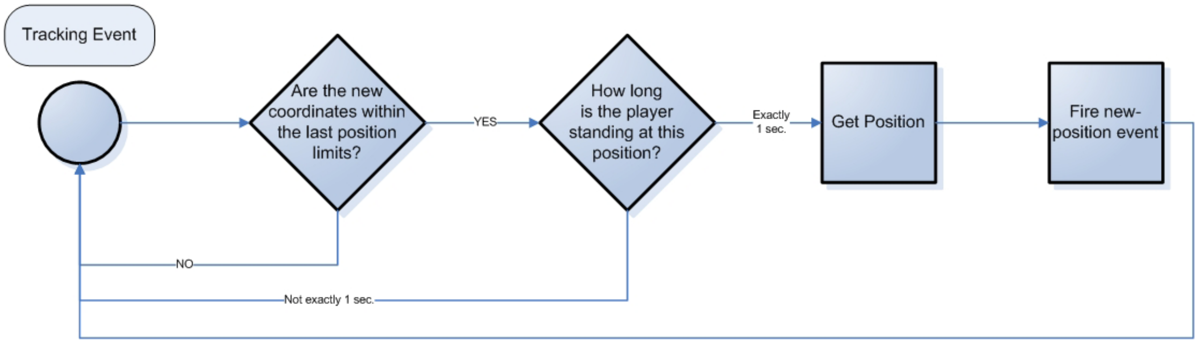


Figure 31. The Position tracking algorithm

4.2.4. Database Manager

In connection with the database, the Database Manager component is responsible for the following operations:

a. *Data storage during the game*

   Every action occurring during the game is stored into the database, stamped by a game and a player identifier. Actions describe the player's activity in terms of a command (if applicable), its result, a timestamp and the time elapsed from the last action (related to the same player). Actions are considered as either *player moves* or *player commands*. For simplicity reasons, an empty *command* field denotes a "movement" action. The full schema of the database used is depicted in Figure 32.

   Action information is used for the composition of the Performance Reports produced by the Database Manager (see subsection "Reporting" below). For example, timing contributes to the estimation of a player's average idle and movement time. If a command-action takes place, "*time elapsed from last action"* counts the time from the player's last move until the current command (**Idle time**). On the contrary, if a movement-action takes place, "*time elapsed from last action"* counts the time from the player's last input until the completion of the move (**Movement time**).

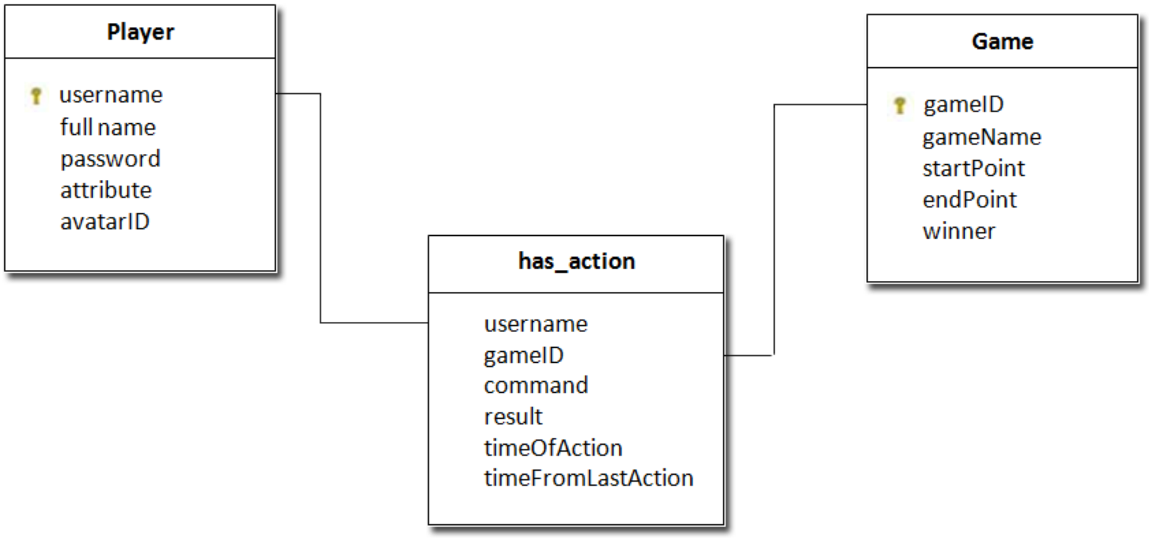


Figure 32. The database Schema

b. Profiling

The profiling operation is designed to keep a complete record of reusable user profiles across the different game applications bound to AmI Game Floor. Profiles are stored into the Player Table, which contains information about a player's full name, username, password, attribute and avatar. To create profiles, the players need to use the corresponding mobile controller's interface, which forwards the data to the current component for storage.

c. Winner judgement

Winner judgement is performed by the Database Manager at the end of a game according to the defined winning criteria. In fact, scoring is the only winning criterion defined, but its evaluation shall vary among the different game modes. The Practice mode, allowing only single player games, results in no winner, whereas in the individual-game mode the winner is judged by the highest score. In the team-game mode (collaborative game), the winning team is announced according to the highest team score. The team score is calculated by the sums of the team players' individual scores.

Every application has to perform a score keeping operation in order to provide the framework with score fluctuations upon every player action. The Database Manager uses the database to store and retrieve score information.

d. Reporting

At the end of every game, the Database Manager retrieves and combines the information stored, in order to publish a Performance Report, also accessible through the file system in Excel format. Performance Report statistics are divided into two basic categories that correspond to the numeric and alphabetic game types, as described in Table 7 and Table 8. Although custom-type statistics are not defined, custom-type games can use many of the alphabetic-type statistics, such as the "Total moves" or the "characters usage" statistics. The latter corresponds to "symbol usage" in case symbols are used instead of letters. In their features description, applications are allowed to determine which statistic category is to be included in the final report or may also contribute to the performance layout by publishing their own additional statistics.

Finally, the Database Manager also generates summary statistics, which are graphically displayed in the Results-view (Section 3.3.2). Likewise, a list of summary statistics is available for applications to choose from or contribute to, according to their needs. Examples of statistics-screening and contribution by applications can be found in Chapter 5. The complete list of the framework's available statistics is presented below.

Actions Summary		**Input analysis**	**Response times**
Successful actions	**Wrong actions**		
• Targets found • Right moves	• Obstacle hits • Out of bounds • Wrong moves • Other player hits	• Total commands • Total distinct commands • Mathematical Operation usage (*e.g. Multiplication: 30%, Addition: 40%*) • Number usage (*e.g. 5: 10%, 10: 12%*) • Command analysis (*e.g. x4: 5%, -6: 10%*)	• Average idle time (*time from last move till new input*) • Average movement time (*time from input till right move*)

Table 7. Numeric-type report statistics

Actions Summary	**Input analysis**	**Response times**
• Total moves • Avg. moves per word	• Total submitted characters • Total distinct submitted characters • Total submitted words • Characters usage	• Average input time per character • Average input time per word • Average idle time (*time from last move till new input*)

Table 8. Alphabetic-type report statistics

Summary statistics

- **Player speed** (moves / movement time)
  Represents the average movement time calculated by the fraction of total moves by the total movement time. Movement time estimation varies across the different game types. In numeric games, movement time calculates the time elapsed from the player's command to the player's right positioning. In alphabetic games, movement time corresponds to the time taken to submit a word, regardless what the player attempts in between. In both occasions the respective controller command signatures the timer to start.
- **Precision award**
  Awarded to the player with the minimum wrong actions. Applies to numeric-type games.
- **Should-practice-harder warning** (wrong actions > 40% moves)
  Appears to each player whose wrong actions correspond to a percent higher than 40% of the player's moves. Applies to numeric-type games.
- **Math-Genie award**
  Awarded to any player that manages to use flawlessly all four mathematical operations. Applies to numeric-type games.

An example of Performance Reports is presented in Appendix C.

4.2.5. User Interfaces

Operating in an Ambient Intelligence environment, the system takes advantage of the computing power distribution in order to provide the following customizable user interfaces:

Input modalities	Output modalities
○ Playfield (positioning data) ○ Mobile Controller • Profiling • Calculator pad • Dictionary pad ○ Remote Game Manager	○ Sound Director with speech synthesis ○ Graphics Generator • Personal view • Playfield view ○ Reports Composer

Table 9. A summary of the framework's UIs

All user interfaces implement the GameFloor service (introduction of the current Section) both for in-between and external communication with the applications' components. In particular, the service facilitates communication for the following events:

a. Application registration
b. Game start
c. Player registration

d. Player move
e. Command submission
f. Result feedback (e.g., target hit)
g. Statistics submission
h. Game end.

For instance, a player move on a position marked by a letter has a simultaneous effect on the Mobile Controller, the Graphics Generator and the Sound Director. The Mobile Controller outputs the letter on screen, whereas the Graphics Generator illustrates the player move and any change in score. At the same time, the Sound Director announces the result of the move, if any, as processed by the application. Depending on the application's configuration, some UIs may be deactivated. For example, the *personal view* (controlled by the Graphics Generator) or the sound output could have been disengaged. The Configuration Manager is charged with activating only the required UIs in the beginning of the game, as described in Section 4.2.1. The functionality of the user interfaces summarized above is described in Section 3, whereas their communication patterns are depicted in Figure 34.

4.3. Game Application Components

AmI Game Floor applications must contain a description of their features and logic operations. If extra interaction is required, applications may contribute with supplementary user interfaces. A general outline of the functionality required by each application is provided below.

a. Game Features

To communicate with the framework, applications have to first introduce themselves using the GameFloor service. In practice, this is performed by submitting a features description in XML format. The Configuration Manager undertakes to validate the application features, which have to conform to the Features Schema, in order to establish a correct binding.

Through their features description, applications are allowed to:

- Register their identity, including the arrangement of positions, their game type and the mark coverage required (what extend of positions is to be marked);
- Define difficulty levels and (optionally) the amount of targets and/ or obstacles per level;
- Specify the positions of the players, targets or obstacles;
- Screen the provided user interfaces, as well as disable parts of their operations judged unnecessary or inapplicable;
- Screen the report and summary statistics and declare any additional ones they need to contribute with;
- Specify a default start-up configuration.

A description of the Features Schema is available in Appendix B.

b. *Game Logic*

The Game Logic component defines the core functionality of the application. The Game Logic manipulates the information deriving from the players' moves and commands, instructing the system to take proper actions and interact with the players through all activated User Interfaces. Various examples of Game Logic operations can be found in Chapter 5.

c. *Supplementary User Interfaces*

Any scattering component, using the GameFloor service, can (selectively) manipulate the information in transit, thus taking advantage of the loosely-coupled communication supported by the AmI Middleware, which contributes to overall multimodality. Therefore, applications are enabled to extend the established interactivity with supplementary modalities, simply using the GameFloor service as their communication channel with the framework. In addition, applications may also determine to handle data privately, by disengaging the respective framework's UI, through their features description. A future extension will allow applications to choose which modality is associated to specific types of information. Section 4.4 further discusses the communication between components.

Further analysis related to implementing AmI Game Floor applications is provided in Chapter 5.

4.4. Components communication

The AmI Middleware, the communication infrastructure below AmI Game Floor and its applications, dictates an event-based model for data exchange among cooperating components. Accordingly, information flowing within, and transmitted from, AmI Game Floor is carried by (GameFloor service) events, to be handled by components with a matching listener. On the other hand, applications convey their information through remote function invocations, supported by the GameFloor service.

Typical information exchange among the software components involved is depicted in the five data flows illustrated in the following Figure.

Figure 33. Typical Data exchange between AmI Game Floor and applications

Considering the natural course of the game, data exchange between the framework and an application starts with the application's registration request (orange flow in Figure 33). If the application Features, escorting the request, pass the Configuration Manager's check, the application is considered part of the system for as long as it runs. At the beginning of the game, the Virtual Grid Generator looks for the selected game's Features copy, preserved by Configuration Manager, so as to customize the corresponding interfaces and set starting positions. In more detail, it constructs a Virtual Grid of a type and size dictated by the application and generates the player positions, pointing at various locations within the grid. Depending on the application's requirements, targets' or/and obstacles' positions are also generated. Finally, an event is raised to notify all stakeholder-components of the outcome (red flow in Figure 33). In parallel, players may use their controllers to create a profile or log in and operate the Remote Game Manager to edit preferences and start the game. Profile information can be reused from past in games as it is stored by the Database Manager.

During the game, Position Tracker raises a "new position" event every time a player moves to a different position (blue flow in Figure 33). In the meanwhile, commands may be submitted from the players' controllers, directed internally to the Graphical User Interfaces, as well as externally (via the GameFloor service) to the application's Game Logic for processing (green flow in Figure 33). The application analyzes the positioning and command data of each player to decide and transmit back their actions' results, containing the effects in score (light blue flow in Figure 33). Further, action results are directed towards the User Interfaces and stored into the database for performance analysis. At the end of the game, the Database Manager analyzes profile and action data to produce a Performance Report and summary statistics displayed on screen. In case the application includes in its specifications a private report and / or summary statistics, the Database Manager requests that information in order to include it in the final report and / or summary.

All information exchanges described above are illustrated in the following Figure.

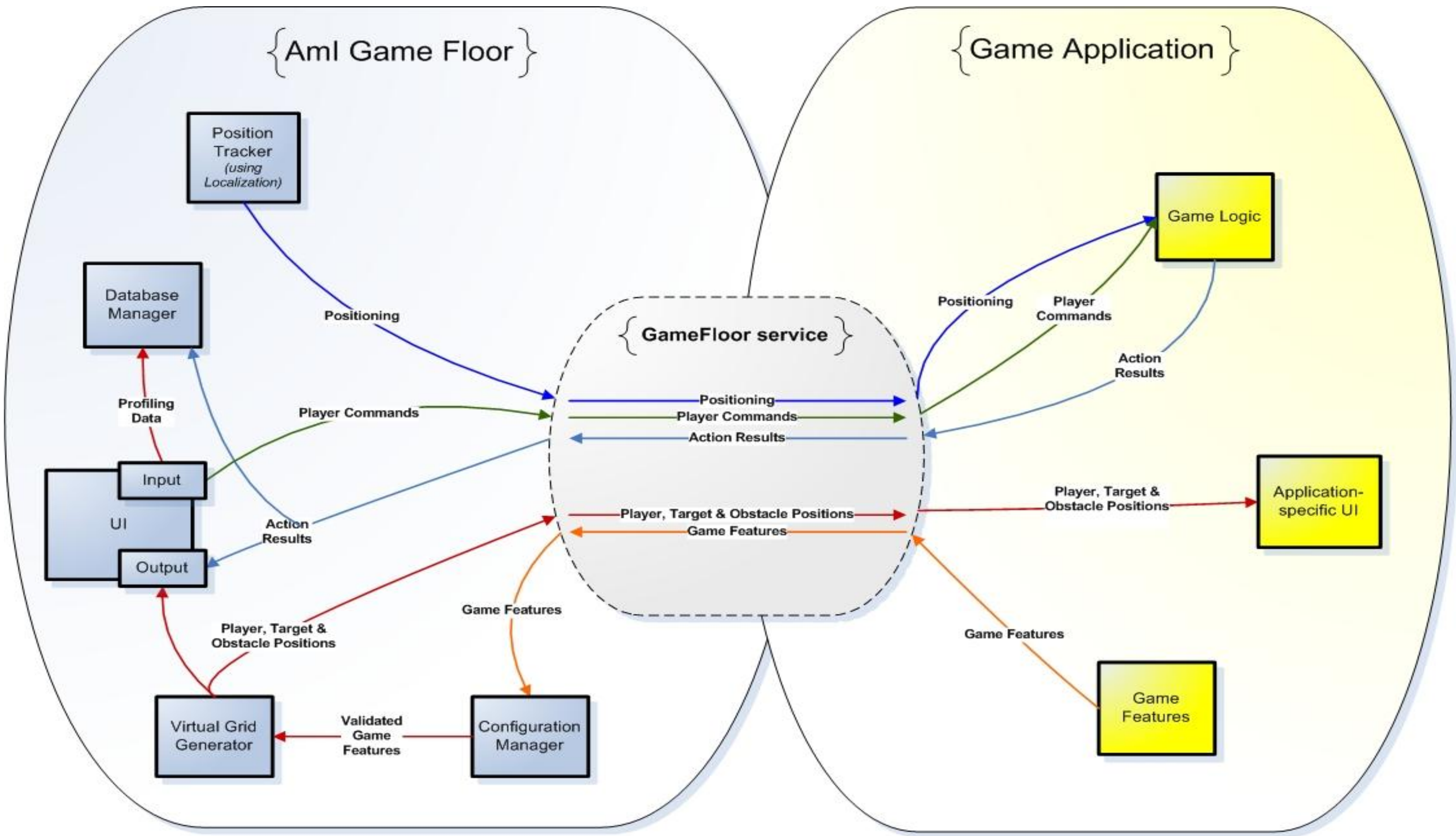


Figure 34. Information exchange among the software components

5. AmI GAME FLOOR APPLICATIONS

AmI Game Floor applications provide the content of learning and set the logic of interaction during play. Influenced by the research on playful learning applications, described in [54, 67], the objectives set for AmI Game Floor applications are targeted to provide:

- Fun
- Challenge
- Engagement
- Learnability

The current Chapter deals with the design and development of AmI Game Floor applications. “Apple Hunt” is an arithmetic game developed as a fully-fledged case study to demonstrate and evaluate the framework. Additionally, three more games, “Word-search game”, a variation of Scrabble™ and “Math Labyrinth” have been designed, so as to provide more insight in how games of various types can exploit the flexibility supported by the framework.

5.1. Apple Hunt

Apple Hunt is an educational application that was developed in order to: (i) assess the degree to which AmI Game Floor facilitates the development of kinesthetic educational games, and (ii) investigate whether educational games integrated within the AmI Game Floor’s environment can have positive influence in the education of children. The application aims at exercising players with fundamental arithmetic operations, taking advantage of the framework’s provisions for playful and collaborative learning, and is oriented towards elementary school children. A prototype of Apple Hunt has been implemented and is up and running in a laboratory space of ICS-FORTH.

5.1.1. Overview and Features

The game’s overall idea is based on the capability of the playfield to bear number-marks (position identifiers). The game is situated on a sequenced arithmetic grid of 10x10 positions, covering a range of numbers from 1 to 100. Figure 35 depicts the playfield, consisting of a carpet shaped accordingly to form 100 equally sized positions.

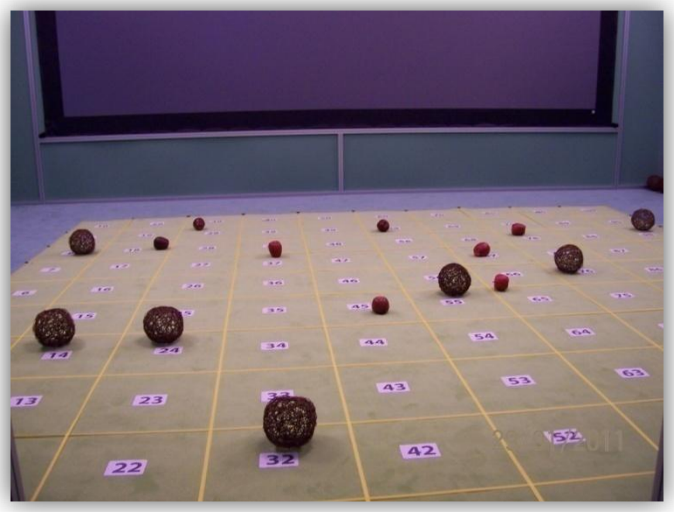


Figure 35. The Apple Hunt playfield, consisted of 100 sequenced arithmetic marks, apples and traps

Apple Hunt defines various Targets and Obstacles, naturally represented by apple and trap dummies, which are randomly placed on the carpet, according to the Virtual Grid Generator output (Chapter 4). The objective of the game is to pick as many apples as possible, avoiding the traps. To move on the grid, the players have to input arithmetic operations which involve the number on the current square and yield as a result the number on the target square.

Hardcoded in its features, the identity of the game also prefigures a four-player capacity and a variety of difficulty levels related to the ratio of traps (obstacles) to apples (targets). For example, "Medium" Level sets twelve targets and ten obstacles in a four-player game, whereas "Hard" Level sets eight targets and fifteen obstacles. In addition, the features allow the use of all the user interfaces provided by AmI Game Floor, relating to the arithmetic game type. The only customization performed by the application is to restrict the Calculator interface from allowing decimal numbers or numbers higher than 9, for educational reasons described in the following paragraphs. Finally, the Performance Report includes all the provided math-type report statistics. Appendix F presents the complete features document of "Apple Hunt".

5.1.2. Action

To log in the game, each player is provided with a mobile controller, which prompts for profile creation or the use of an existing one. As soon as log-in is complete, the system first responds by the generation of random player, target and obstacle positions, and the controller interface switches to the Calculator (Section 3.3.4), enabling the user to submit commands in the form of arithmetic operations. The players then have to place the apples and the traps on the respective positions and occupy their own starting positions. Action begins.

To move around the playfield, each player must first target a position using the mobile controller. No turn-restrictions are applied. A command is composed of a simple arithmetic operation combined with the number of the player's current position. For example, standing on position "5" and targeting "25" would require a "multiply by 5" command. More specifically, to submit a command, players first have to choose an operand, then a number varying from 1 to 9 and finally press the "submit" button, as shown in Figure 36. After submitting a command, the player is allowed to move towards the targeted position.

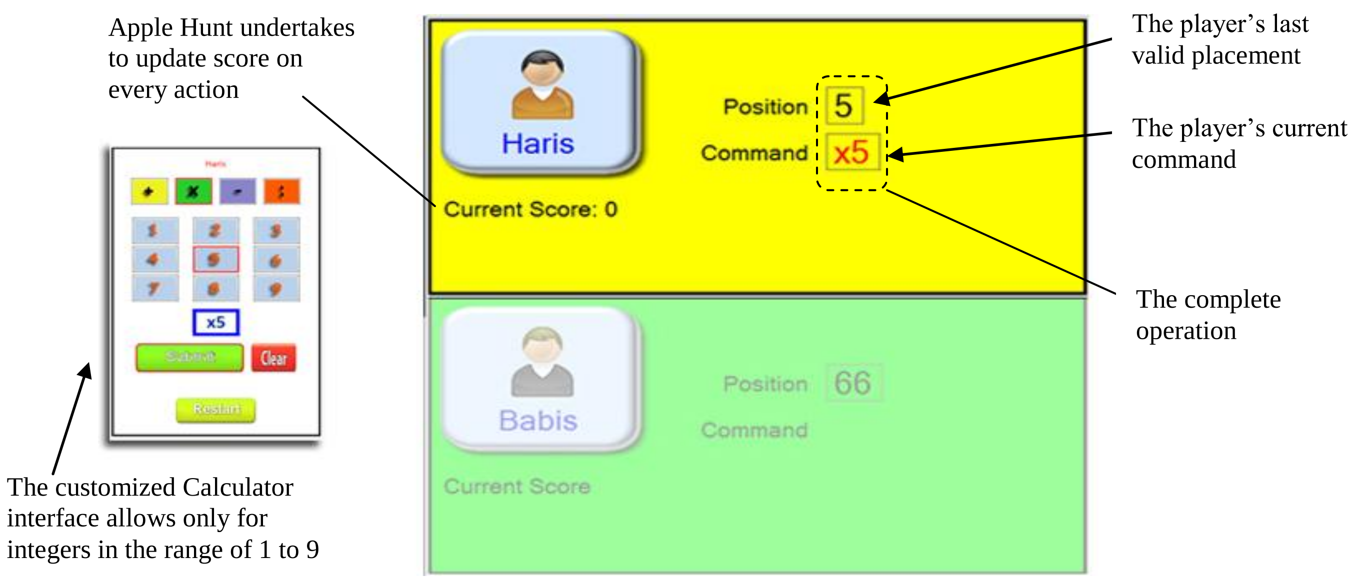


Figure 36. Using the Calculator interface (left) to target position 25. The command "x5" is shown in the player-performance view (right). Adapting to "Apple Hunt" features, the controller interface disables two-digit and decimal numbers

It should be noted that the use of numbers higher than 9 is restricted according to the educational objectives of the game. In fact, it was noticed that using numbers higher than 10 would allow the players to avoid the multiplication or division operations[10]..

Strategic play not only induces a careful choice of target positions, but also leads to analytic thinking of the optimal arithmetic operation that will minimize the actions (commands and moves) required to reach the apples. Apart from the presence of apple and trap dummies on the game floor, strategic play is also accommodated by the framework's playfield view GUI (virtual grid representation), which provides a dynamic sky-view of the game. At the end of the game, the winner (or winning team) is judged by the highest score (or score sum), as a result of the best balance between apple and trap hits. An analysis of the score rules is shown in Figure 37.

5.1.3. Implementation

As described in Chapter 4, successful communication among applications and AmI Game Floor lies in the correct design of the game's features and logic. Complementing the analysis of "Apple Hunt" features in

[10] Also, operations containing "10" were considered too obvious and of negative effects on children's learning. Using the first prototypes, during the initial development, it was noticed that "10" was the most frequently used number. This fact was attributed to the playfield's arrangement, as addition or subtraction by 10 required only one step rightward or leftward respectively.

Section 5.1.1, this Section deals with how the application handles the information received via the AmI Middleware by its Game Logic component.

The Game Logic undertakes to process information relating to the players' actions, arriving through game events, in order to inform the system of the actions' outcome per player. Considering the fact that any player might move or submit a command in any order and frequency, this component was designed to monitor the sequence of actions in accordance to the rules of the game. In particular, the players are instructed to move only upon command submission and then find the right position before targeting a new one. Covering the time gap between targeting and moving to a position, an action-log was designed for storing the potential results of commands until the right moves take place. Take, for example, a player who stands on "4" and targets position "32" by the wrong command "x7". The Game Logic receives the command event ("x7") and logs the action as if the player is targeting position "28". If position "28" is not empty (i.e., a player or an object is present), the score modification responding to this action will be logged. Unaware of the results of this action, the player will wander within the playfield to find the right position. The application will not reveal the results until the player finds the right position, which might never occur, in case the game ends in the meanwhile. The Action log is implemented as a string array that entails coded information for each player's actions. In conclusion, the logging procedure (Figure 37, described here, stores potential outcomes in pairs of result and score modification, which comprise the information transmitted back to AmI Game Floor upon the players' right placements.

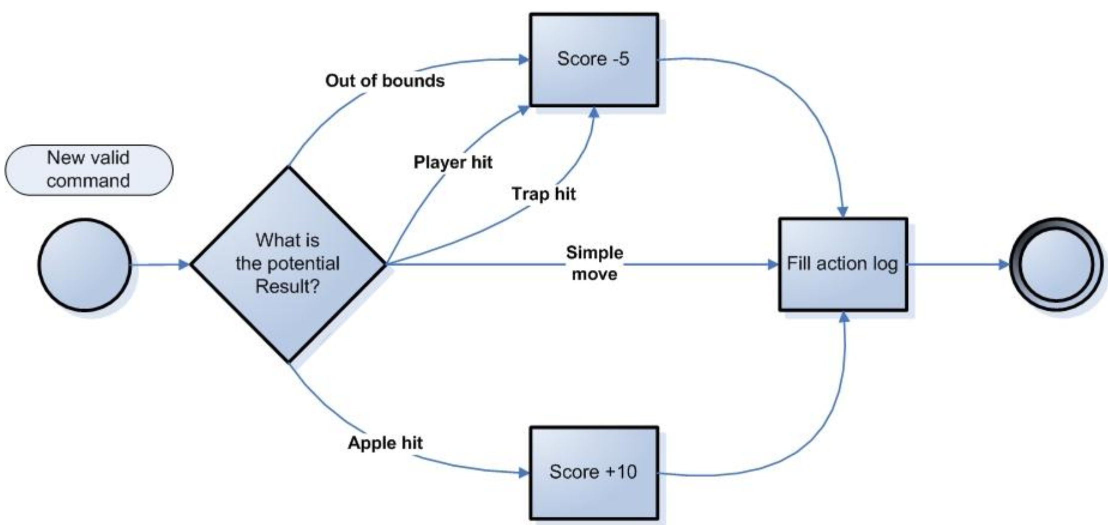


Figure 37. The action-logging procedure

In short, the Game Logic receives movement or command information corresponding to arithmetic operations submitted by each player. Upon each command, it foresees the outcome of the player acting accordingly (moving to the targeted position), by calculating the result of the submitted operation. The outcome and score change will stay logged until the player makes the right move. Upon each move, the Game Logic will validate and compare the positioning information received to decide for its correctness, so as to transmit the information logged back to the system via the AmI Middleware. In addition, upon each move, the Game Logic checks for game-over conditions. In practice, two facts are only considered as game-over conditions: all apples being collected or all players being eliminated by targeting out-of-bounds.

Game Logic's operations break down in two algorithms, executed whenever a player submits a command or moves to a new position, as described below:

a. *Action Management algorithm*
Action management is triggered on every "player-move" event, occurring when a player moves to a new position. The algorithm first checks for the validity of the move, depending on whether the player had previously submitted a command. If a command was placed, it will then use the event's positioning data, deriving from Position Tracker, to decide if the move was right. If the number identifying the new position matches with the submitted arithmetic operation result, the Game Logic will read the action log, filled upon the player command, and inform the system of the successful outcome and score change. At this point, the Database Manager (Section 4.2.2) stores the action and starts the idle-time counter for the current player.

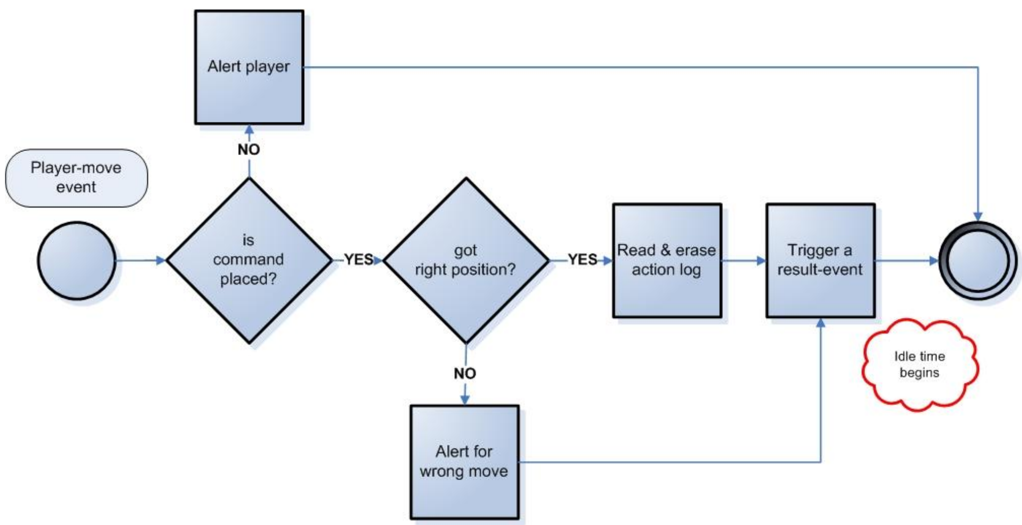


Figure 38. The Action Management algorithm

b. *Command Evaluation algorithm*
Command evaluation is triggered upon every "new command" event. It is designed to use the positioning data (a position number coupled with a player identifier) received from the framework in order to form the complete arithmetic operation and estimate its (potential action) result. Action results may vary from catching an apple to hitting a trap or another player (not allowed by the rules) and from a simple move to targeting outside the game boundaries. In any case, this task will trigger action logging (Figure 37), in order to keep track of the outcome separately for each player. At this point, the Database Manager (see Section 4.2.2) is enabled to start the move-time counter. The following Figure illustrates this algorithm.

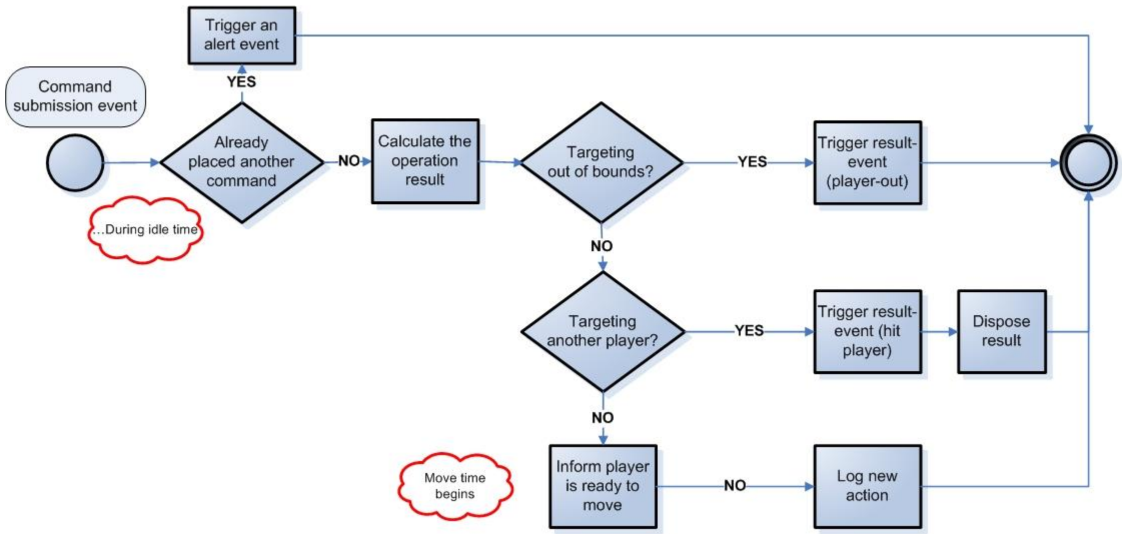


Figure 39. The Command Evaluation algorithm

5.2. Design Examples of Supported Applications

This Section illustrates several AmI Game Floor application designs in order to uncover the flexibility offered by the framework in the development of diverse applications. In particular, the following sections describe three different application designs, focusing on their diverse characteristics as supported by the framework.

5.2.1. The Word-search Game

The word-search scenario is borrowed from the homonymous game, well-known as a quiz-type game in magazines and newspapers. The word-search game typically appears as a grid of randomly sequenced letters, some of which shape words if read in order (horizontally, vertically or diagonally), as seen in Figure 40(b). The objective is to highlight as many word-forming letter combinations contained in the grid as possible. A Word-search game has been designed for the AmI Game Floor framework. The design focuses on:

- The custom game type (with alphabetic & sequenced numeric marks), involving full-grid mark-coverage;
- The contribution of additional statistics calculated by the application;
- Alternatives of the same scenario with and without targets and obstacles.

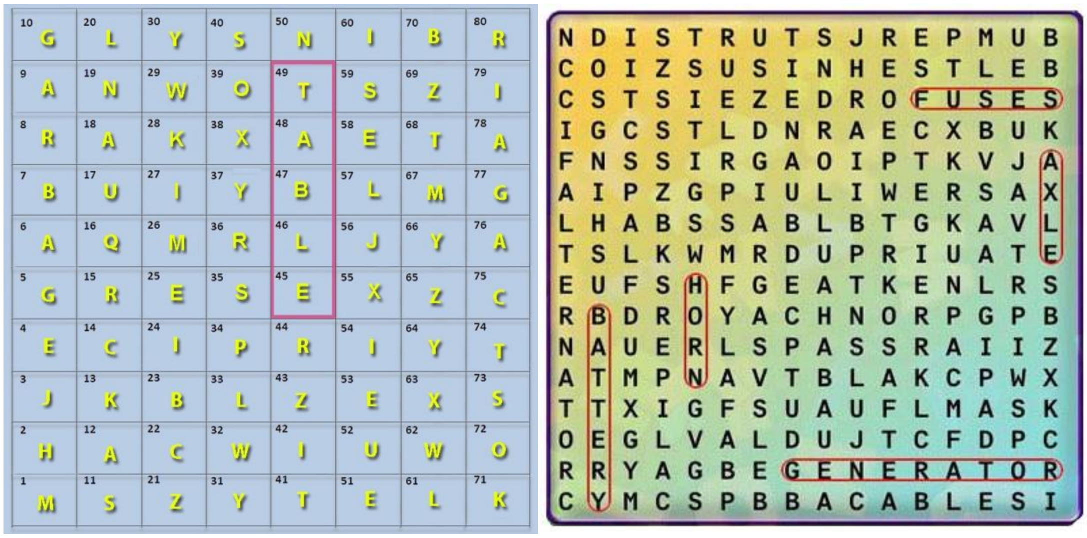


Figure 40. (a) Left: The playfield view illustrated by the frameworks GUI. (b) Right: Word-search typical look

Overview and Features

Shaping the playfield to match the game's typical design would require creating a grid of randomly sequenced letter marks, one for each position, thus designing for an alphabetic type of game (Section 3.2.1). Yet, provided that the same letters may appear more than once in different positions, a number or a symbol could be additionally used as their unique identifiers, necessary for supporting the players' navigation. In the proposed design, a custom grid style is used with alphabetic central marks supported by sequenced numeric side identifiers. In addition, full range mark-coverage is required. Figure xx(a) represents the playfield's virtual look, as illustrated by the framework' respective GUI, in contrast to the typical appearance of the Word-search game, depicted by Figure xx(b). The natural playfield has to be identically shaped to the virtual playfield view. Although the design suggests an 8 x 10 grid, alternative versions could use any other sizes provided that they avail from adequate physical space.

The customization of the graphical user interfaces omits the player-performance view (Section 3.3.2), as in this instance the virtual playfield view provides all information required. Moreover, the Remote Game Manager is fully utilized, whereas the alphabetic controller interface is used, as dictated by the game's custom type.

Customizing the Performance Report is more complex, because the scenario does not allow the use of the most framework's given measures and statistics. This is due to the fact that the majority of the alphabetic-type statistics are strongly connected to the letter-submission operation of the mobile controller, which is not applicable in this design. Subsequently, the application overrides a part of the provided report structure, passing its private measures and statistics in the final report. This procedure is analyzed in the next subsection. The complete application's features document can be found in Appendix F.

Game Logic

Having defined the shape of the framework's natural and virtual infrastructure, the application has to implement the Word-search game rules and logic. The application logic encapsulates operations like *word validation*, *spelling check*, *score keeping* and *performance measurement*. For the purpose of the current analysis, the game rules are not described.

Word validation and spelling check can make use of an open-source dictionary or of a word-editor, supported by a supplementary GUI, to set the quested words. Upon every new game, the application has to pass a new set of words to the framework, which will trigger an update request for the Virtual Grid Generator (Section 4.2.2). Each player's score may rise on correct word submission and lower on wrong words. Spelling mistakes also affect the final score, whereas a bonus is offered for short-word tracking as well (less than four letters).

To conduct performance measurement, a database is needed to record the players' actions during game. Analysis of the data stored will contribute to the Performance Report, generated by the framework. Action recording, also performed by the system, is required, only because this design extends the provided statistics-set. Analyzing the players' actions at the end of the game, the game logic will generate the following statistics:

Report statistics	Summary statistics
○ Total wrong words (*inexistent words*) ○ Wrong words analysis (*List of wrong words submitted*) ○ Total spelling mistakes	○ Grammar-expert award (*no spelling mistakes & no wrong words*) ○ Practice warning (*spelling mistakes extend 20% of the letters submitted - moves*)

Table 10. Additional statistics generated by the game logic

Finally, the application will have to dispatch the statistic results to the framework, in order to fill in the Performance Report, in addition to the following native measures, requested by the game's features description:

- Total moves
- Average moves per word
- Total submitted words
- Average idle time
- Average input time per word

Action

Implied by the analysis above, the objective of the game is to find the word-forming combinations of positions within the playfield. For this purpose, the players are required to use the (alphabetic) controller interface. In particular, to construct a word, the players simply need to subsequently stand on their chosen series of letters (positions) for at least one second per position. Every move is reported to the application, which records it as an action involving a player's move to a letter in a specified point of time. At the same time, the virtual playfield marks the positions covered with a distinctly-coloured frame, as seen in Figure 40(a).

As soon as the player has covered all the appropriate positions required for the envisioned word, the player uses the controller's word-submit command (Figure 41), so as to trigger the word validation performed by the Game Logic.

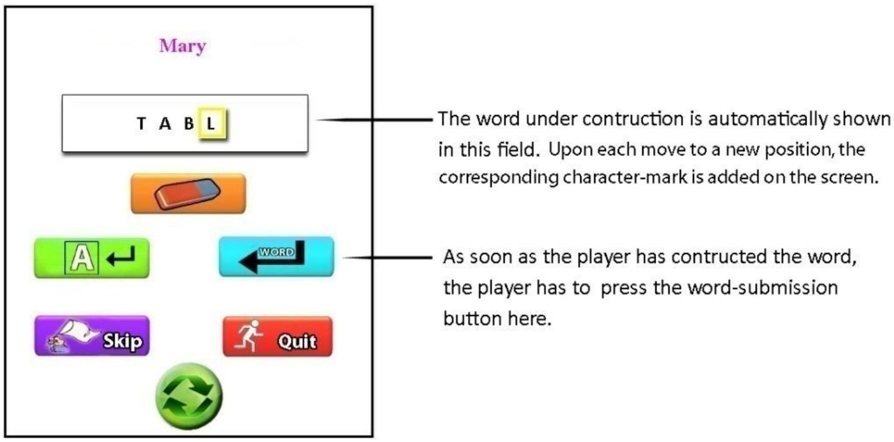


Figure 41. Using the alphabetic controller interface to submit a word

Depending on the validation result, the application logic updates the score and forwards both the result and the score back to the framework, in order to inform all components involved. For example, the virtual playfield (Figure 40a) will hide the highlight from the respective word, if informed that the validation result was negative, whereas a sound effect will be heard, advising the player to look more carefully. At the same time, the wrong action is recorded in the application's database, excluding the player from the candidacy to the "Grammar-Expert award".

At the end of the game the application analyzes the database information to generate the statistics described in Table 10, in order to contribute to the final Performance Report, by dispatching the statistics via the middleware.

It should be noted, finally, that this design does not involve any targets or obstacles. However, to enrich the game experience, targets could be used as wild-cards. Standing on a wild-card, the player would have to use the controller's letter-submission function to cast it to a proper letter, if judged applicable to an extra word.

5.2.2. A Scrabble Variation

Scrabble™ [72] is a word game in which two to four players score points by forming words from individual lettered tiles on a game board marked with a 15-by-15 grid (Figure 42b). The words are formed across and down in crossword fashion and must appear in a standard dictionary. The game is round–based, and during a round each player is allowed to use only seven letters picked in random. Official reference works (e.g., The Official Scrabble Players Dictionary) provide a list of permissible words. The following paragraphs describe the fundamental design of a Scrabble-like game which makes use of the AmI Game Floor framework. The design emphasizes in:

- The custom game type (with alphabetic & sequenced numeric marks), involving a short mark-coverage;
- The customization of the framework's UI (how the application handles information through a supplementary GUI);
- The use of obstacles for neutralizing game positions from the start.

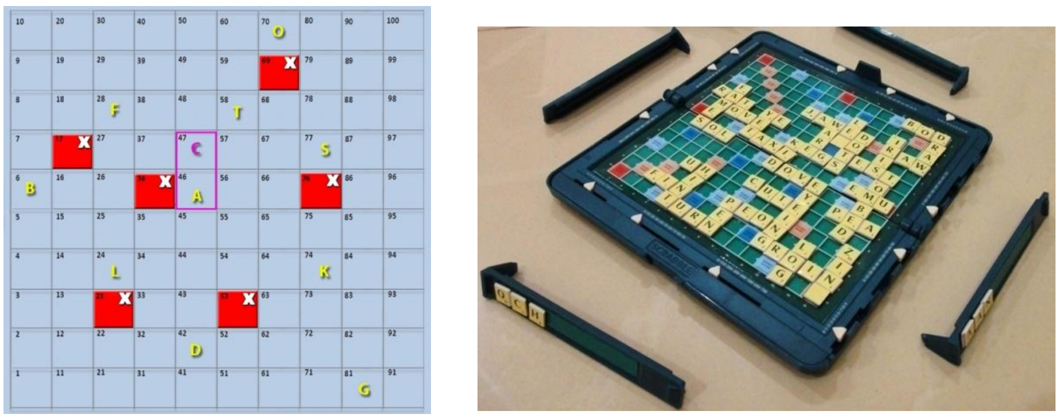


Figure 42. (a) Left: The playfield view illustrated by the frameworks GUI. (b) Right: Scrabble's typical look

Overview and Features

This design makes use of a 10x10 custom grid style, containing random alphabetic and sequenced numeric marks, as depicted in Figure 42(a). In contrast to Scrabble, the playfield will contain obstacles (letter-input blockers), whereas dispersed letters will cover an extent of the available positions, rendering alphabetic mark coverage equal to 10%. It should be noted that the numeric (side) marks are only suggested for cross-reference reasons between the GUI and the playfield and should not be confused with the letter-points appearing on Scrabble's tiles. Also, the application is not round-based and defines three difficulty levels, discriminated by the number of obstacles in use.

Furthermore, the application defines three different game views in total, supplementing the existing infrastructure (Section 3.3.2) with a GUI containing information for each player's available letters and score (Figure 43). More specifically, the application chooses to show score in its private GUI, allowed by the framework to deactivate the respective function from the standard personal view. Besides, the design of the GUI outweighs the player-performance view and therefore the latter is deactivated.

The customization of the Performance Report and Summary differs from the Word-search game only in that it additionally includes the input-category statistics. A summary of the statistics required is presented in Table 11, whereas the complete application's Features document can be found in Appendix F.

Actions Summary	Input analysis	Response times	Application specific
○ Total moves ○ Avg. moves per word	○ Total submitted characters ○ Total distinct submitted characters ○ Total submitted words ○ Characters usage	○ Average input time per character ○ Average input time per word ○ Average idle time (time from last move till new input)	○ Total wrong words ○ Wrong words analysis ○ Spelling mistakes

Table 11. A summary of the game's report statistics

The summary statistics required (same as the Word-search game's) are:

- Player speed
- Grammar-expert award
- Practice-harder warning

Application-specific GUI

As referred above, the application chooses to present score in its own way, hiding the framework's respective view and embedding it into its private GUI. However, the private GUI's main purpose is to inform the players about their available letters for each word-formation (seven per word). The GUI also hides the letters used during word formation to reveal only the ones in disposal. Besides, it presents a list of each letter's value according to the English letters usage frequency, as adopted in Scrabble. The following Figure presents a mock-up of the GUI.

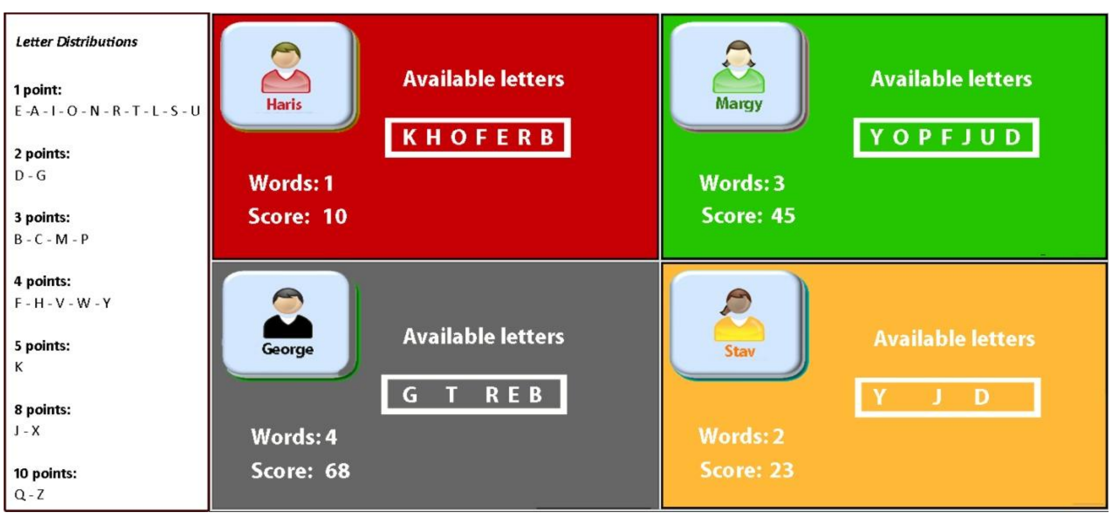


Figure 43. Supplementary GUI with letter availability and scoring details

Game Logic

The application logic encapsulates operations like word validation, spelling check, score keeping and performance measurement, similar to the Word-search application's logic described in the previous Section. Nevertheless, based on the adapted Scrabble's rules, the application has to impose restrictions on word formation, according to some chosen dictionary. Therefore, word validation has to be based on some Scrabble dictionary, such as those widely spread on the web. Score calculation performs a sum of the corresponding letter values and is enabled upon each correct word submission by the players. Irrespectively of whether a submission is correct or false, performance measurement procedures record all actions and input, so as to provide a complete overview of the game in the end. Despite the fact that the application undertakes score keeping, it is the framework's responsibility to account for the winning criteria, which vary according to the game mode. More related information can be found in Section 4.2.4(c).

Action

Once again, the game starts as soon as the players have prepared the playfield by laying down letter dummies, as appearing in the virtual playfield view of the system. The objective is to combine the placed letters with available (non-placed) letters, so as to form as many words as possible across and down in crossword fashion, in such a mixture that maximizes the points earned.

In order to place letters on the playfield, the players first have to stand on a chosen position and use the alphabetic controller's "submit letter" button (light green button in Figure 44). Afterwards, the players are allowed to lay down the corresponding letter dummy. To decide which of their available letters should be used, the players can be advised by the extra GUI provided by the application. On the other hand, in case an existing letter is to be combined, the players simply need to stand on the related position for at least 1 second, in order to be considered as part of the word under formation. The following picture shows the details of the mobile controller.

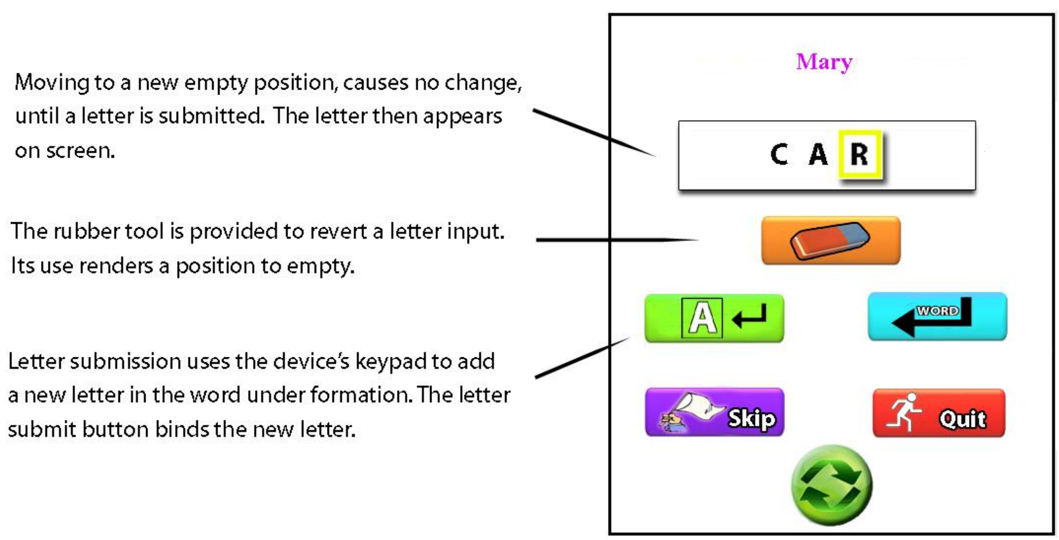


Figure 44. Using the alphabetic controller interface for word formation and submission

Each letter or word submission is further recorded into the application's database, which also accounts for each action's occurrence time. Upon game over, the application analyses the data stored, in order to complement the Performance Report with the statistics summarized above.

5.2.3. Math Labyrinth

Math Labyrinth is an original mathematics game, played by two to four players, seeking their way to the exit through the corridors of a labyrinth. The game is played on a (randomly) numerated playfield that contains a considerable amount of obstacles ("walls") forming a "labyrinth". Moving around the labyrinth requires step-by-step moves on the numerically-marked positions. For each move, the players have to execute the right mathematical operation, whose result must target an adjacent position, avoiding the walls (Figure 45).

This design shows how:

- A numeric game type (with random numeric marks) is implemented, involving full-range mark-coverage;
- The application defines starting positions, instead of letting the framework perform this operation in random fashion;
- The Calculator interface can be used for more complex commands (in the form of mathematical operations).

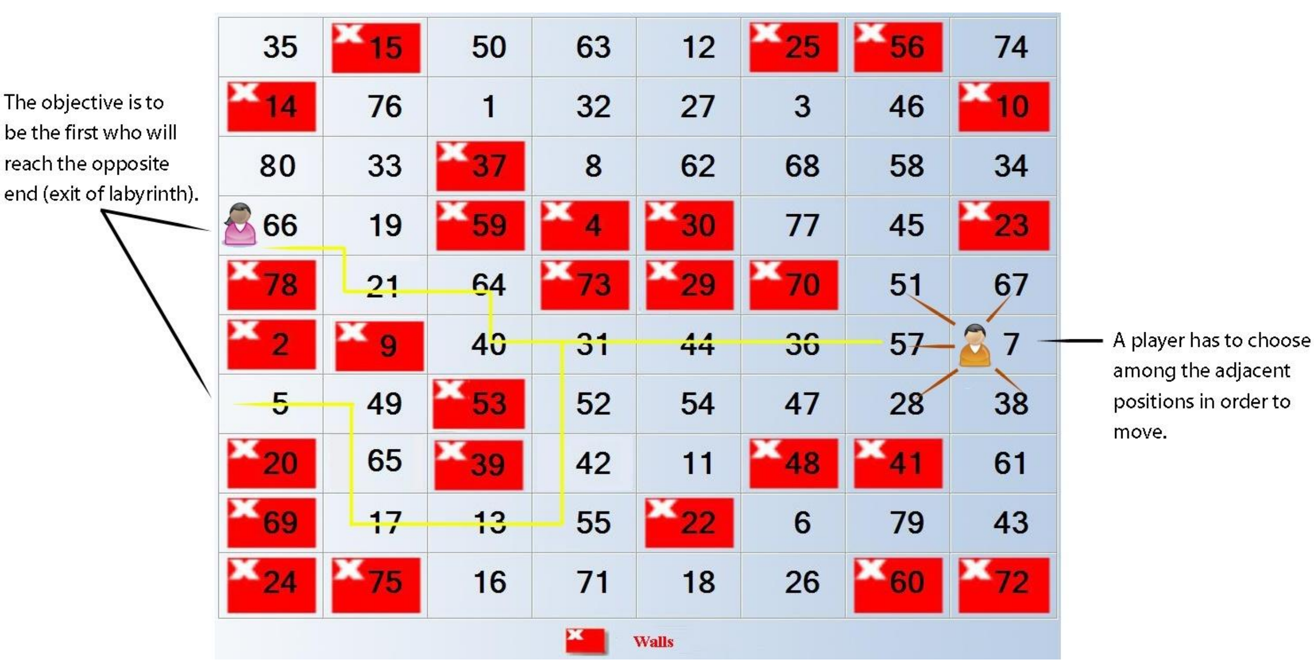


Figure 45. Math labyrinth's virtual playfield

Overview and Features

The application will make use of a fully-marked numeric grid style, which corresponds to the (fixed) randomly-numerated virtual playfield generated by the framework. Although the order of the numeric marks remains the same, the labyrinth's pathways will differ from game to game, as the application dictates the system to randomize the wall (obstacle) placements.

In order to set opposite starting positions on the submarginal columns of the grid, the application defines fixed starting positions (e.g. 66 - 7 - 5 -34), on which no obstacles are allowed to be assigned by the system. Moreover, Math Labyrinth defines three levels of difficulty, characterized by varying numbers of walls.

Finally, this design is fully supported by the framework's graphical user interfaces and performance measurement. It only excludes a couple of statistics relating to the presence of targets and defines the "total targeting mistakes" statistic, as well as the "Most Valuable Player" award for the performance summary in addition. The latter is awarded to the player who performs the maximum actions with the minimum mistakes. The application's Features document can be found in Appendix F.

Game Logic

The Math Labyrinth's Game Logic should not differ considerably from Apple Hunt's, described in Section 5.1.3. The Game Logic here has to cater also for command and move validation, game-over conditions check and performance measurement in communication with the framework. During move validation, the application must prohibit moves on obstacles, which can also result in consequences in score. Moreover, to enforce strategic play, the Game Logic prohibits the use of the same position by two or more players. This way, each player has to smartly decide for their route in the labyrinth. Also, upon every action, the game-over conditions are to be checked, which are limited to a player reaching the exit (a submarginal empty position) or all the opponents' elimination by targeting out-of-bounds. The same performance measurement operations (as of the formerly described applications) are followed.

Action

The game may start as soon as the players have laid down the walls on the playfield as indicated by the virtual playfield interface. Each player has to choose the shortest path within the labyrinth and use the numeric controller to target the pathway positions step-by-step. In contrast to Apple Hunt, Math Labyrinth does not impose any restrictions on the use of the Calculator interface, so that the players can be flexible with composing mathematic operations. The following Figure shows several examples of valid operations for targeting position "57" from position "7".

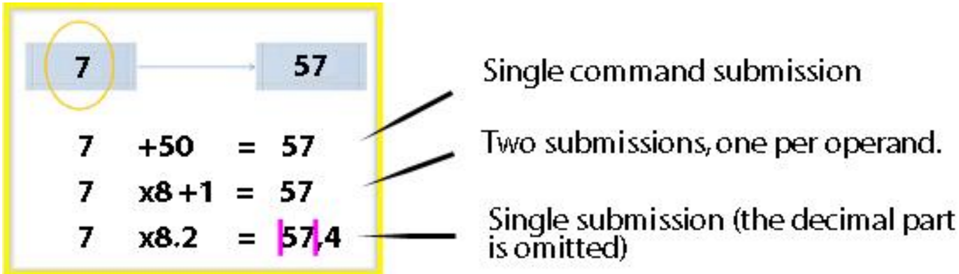


Figure 46. Three different valid ways to target position “57” from position “7”.

The game continues until a player reaches the exit and signatures the end of the game. A particularity found in this game is that random obstacle locations may favour a player or a team against the others with shortest pathways. However, points here are gained with every right move and lost with every wrong one. Therefore, the framework’s winning criterion compensates for this injustice, as the winner is judged by the total points gained, even if the respective player is not the first to reach the exit.

6. EVALUATION

This Chapter describes the evaluation of the Apple Hunt Game by young testers. The conducted evaluation, besides providing useful feedback on the game itself and its current implementation, also allows drawing some conclusions regarding the added value of the overall AmI Game Floor framework. In particular, Apple Hunt's evaluation is designed to uncover the effectiveness of the framework, as the application builds on its features, which are eventually part of the whole user experience. While learning applications intend to invoke learning through *fun*, *challenge*, *engagement* and provision of appropriate content, the framework aims at providing the technological means to support this achievement. The AmI Game Floor platform plays this supportive role by encouraging kinesthetic activity and collaboration among learners. A full list of the evaluation parameters is provided in Table 13.

The Chapter begins with an introduction about game evaluation compared to the evaluation of other types of applications, as well as the challenges faced when involving children in the evaluation process. Then, the adopted evaluation process and the conducted experiment are described, followed by an overview of the obtained results. Finally, the chapter closes with a discussion of the findings.

6.1. Background knowledge on game evaluation with children

Games, in contrast to productivity applications, are designed to entertain. Pagulayan et al. [51] stress the importance of considering the differences between games and productivity applications in evaluation design. The following table summarizes their relative key differences.

Productivity applications	Games
- Are designed to make tasks easier, quicker and flawless; - Focus on producing an improved product or result; - May have externally defined goals; - Should demonstrate consistency of user experience.	- Are designed to entertain; - Intend to stimulate thinking and feeling; - Define their own goals; - Should provide a variety of experiences.

Table 12. The main facts influencing the evaluation of games compared to productivity applications' [51]

The analysis above indicates that typical usability evaluation does not apply to games, as the emphasis in playful activities does not necessarily lie in efficiency, but rather in pleasure and fun. According to [51], "the ease of use of a game's controls and interface is closely related to fun ratings for the game", whereas Barendregt et al. [5] show that task-oriented evaluation of *fun* in computer games is ineffective. Jordan [32] argues that the relationship among *functionality*, *usability* and *fun* can be described as a hierarchy of needs, implying that as soon as a lower level need has been met, the subsequent level should be considered. In this sense, AmI Game Floor would be characterized by the following "needs hierarchy":

1. Fun and learning → 2. Usability of controls → 3. Functionality

Evaluation with children is gaining considerable attention in research in Human Computer Interaction (HCI) in recent years [4, 5, 12, 13, 23, 29, 42, 44, 67, 68]. These works have proved that designing for children without their involvement in the process results in serious drawbacks of the final product. This is mostly attributed to the fact that adults cannot accurately see the world through the eyes of children, as a result of the evolution of their intelligence. In order to involve children in the design process, several guidelines [23] and methodologies have been suggested. Several evaluation methodologies for children have been suggested in the literature:

- The *Think Aloud protocol* [15] is an evaluation methodology according to which the testers are asked to express what their thoughts during a test.
- *Constructive interaction* ("Co-discovery") [46] involves teams of two participants, which perform tasks collaboratively while being observed. It is based on the constructivist theory of learning, as the presence of a partner helps the testers rebuild their understanding of the test through conversation.
- *Cooperative evaluation* [47] is performed by a group of people that engage in a collaborative task. A facilitator may question them with respect to their intentions or expectations when they are quiet.
- *Peer-Tutoring* [27] is performed by an experienced child teaching another child in a task in which the latter is a novice. This approach provides information about the teachability and learnability of a task or software and promotes communication during the evaluation process.
- *The Problem Identification Picture Cards* methodology [4] is oriented towards young testers, which are equipped with a set of picture-cards, intended to be used as evaluation material during some task.

Despite the researchers' attempts of elaborating effective evaluation methodologies for children, the criticism of the suggested models is noticeable. Xu et al. [64] argue that the referenced studies "have tended to report how adult-centered methods need to be adapted for child users and how easy the different methods are for children to use". Zaman [67] opposes to the Think Aloud approach as it requires a high cognitive effort, while the children need to focus on the game's exploration. Moreover, Hanna et al. [23] complements that observation plays a crucial role in understanding the children's feelings, as their behavioural signs are more reliable than their responses to questions.

6.2. Evaluation design

On the basis of the above, the evaluation process followed for Apple Hunt took into account the following basic guidelines:

- Apple Hunt should be evaluated as a game for children and not simply as an educational computer application;
- The evaluation should emphasize measuring *fun* and *learning*[11], and secondarily assess the controls' usability;
- No specified tasks should be indicated to children. They should be free to explore the game at their own pace and be given the chance to familiarize with the game before they are asked any questions.
- A combination of methods should be used with an emphasis on the observation of children's expressions.

In [4] seven types of problems in games are identified. In the evaluation of Apple Hunt, these seven types of problems were taken into account along with the objectives set for AmI Game Floor applications, as per the table below.

Apple Hunt objectives	Types of problems in games
Fun Challenge Engagement Learnability Collaboration Kinesthetic activity	Usability problems on the cognitive level Usability problems on the physical level Inefficiencies Challenge problems Fantasy problems Curiosity problems Control problems

Table 13. The evaluation parameters

The evaluation process was split in three phases:

Phase	Methodology	Duration
1. Familiarization (Practice)	Think aloud & active intervention	30 min
2. Free game	Observation	60 min
3. Evaluators' committee	Picture-cards & laddering	30 min

Table 14. An overview of the evaluation phases

The first phase involved familiarizing the children with the game environment, including an introduction to the rules and controls of the game. During this practice each child would try the game individually and be asked to think aloud, whereas a facilitator would motivate them to express their thoughts and expectations whenever they are quiet (*active intervention*). This methodology is similar to Zaman's [67]

[11] The evaluation of learning quality would require longitudinal tests [24] and is therefore beyond the scope of this thesis. Instead, the evaluation reported intends to assess the effectiveness of the system's technological factors that encourage and contribute to learning.

and comprises a combination of the *Think Aloud & Active Intervention* techniques. This phase was estimated to last for approximately 30 minutes.

The second phase foresaw free play, since by this point the children are already trained into the AmI Game Floor environment and the rules of the Apple Hunt game. *Observation* was chosen as the most appropriate method to record the participants' reactions and mood, whereas the performance reports (Section 4.2.4) would be used to extract information about regular problems (mistakes) during subsequent games. Similarly, the controller logs ("clicks logs" - Section 3.3.4) would be analyzed to assess the controller's interface usability. The second phase was estimated to have a duration of 60 minutes.

The third phase, influenced by the Picture Cards method [4], involves the composition of a young evaluators' committee. Each child would be provided with five expressive picture-cards (Figure 47), instead of the numeric cards typically used in evaluation committees. The children would be placed in hemicycle fashion facing the facilitator, who would address a series of questions to them (Section 6.4.3). The children would be asked to answer the questions by raising the appropriate card, while the facilitator would perform laddering [57] whenever judged necessary, in order to extract as much of the committee's comments. This phase was estimated to last 30 minutes.

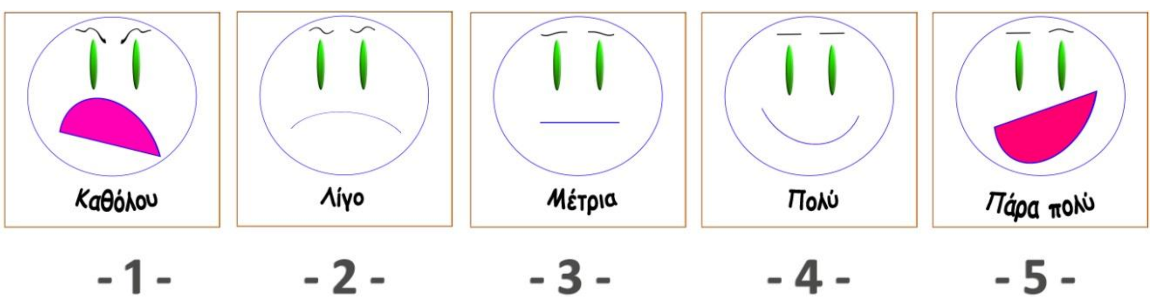


Figure 47. The picture cards designed for the young evaluators' committee, ordered from negative to positive. The numbers below the cards indicate the corresponding Likert-like score scale used to measure each question's score.

The evaluators' committee approach differs considerably from the *Problem Identification Picture Cards* methodology [4] in two main points. First, the cards used by Barendregt et. al correspond to eight specific emotional conditions, whereas those used in this case reflect a Likert-like score scale. Secondly, Barendregt et. al provide a box with eight compartments (one for each evaluation subject), in which testers have to place the appropriate card according to their taste *during* the testing process. Hence, it is unclear if this arrangement can be efficient in fast-paced games, where testers' reflexes are important. On the other hand, children committees, settled *in the end* of the testing process represent a fresh approach to evaluation with children which is believed to bring an interesting potential in the context of Ambient Intelligent Environments and educational games.

6.3. Evaluation experiment

The evaluation of the Apple Hunt game was conducted in two sessions involving 9 children in total, which were videotaped for later analysis. All children had a considerable experience with computers and technological gadgets, whereas most of them had been playing computer games since long. Their age varied from 7 to 11 years old, while 6 out of the 9 participants were 6th-grade elementary school students (according to the Greek educational system). The demographic characteristics of the participants are shown below.

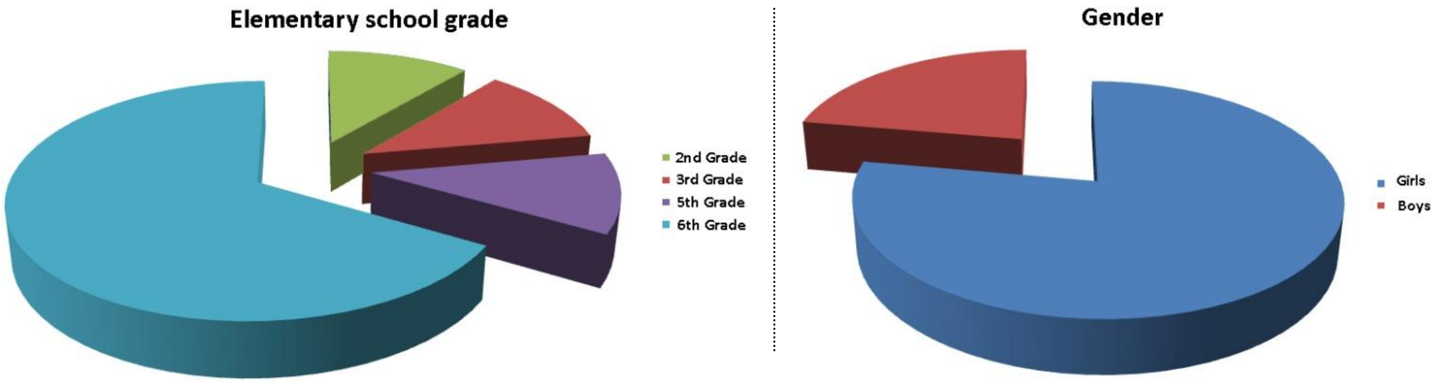


Figure 48. Demographic characteristics of the participants

The first evaluators' group consisted of 3 girls and 1 boy, the youngest being a 5th-grade student, which formed a cohesive group as for the children's cognitive levels. The second group consisted of 4 girls and 1 boy and had a wider age range (the youngest participant being 7 and the oldest being 11 years old).

During the *first phase* (practice), the children practiced in teams of two persons playing in turns, with only one team participating at a time. This arrangement helped the participants engage in discussions, which seemed to overweigh the drawbacks of the Think Aloud method. The main interest in this phase was to realize possible misconceptions or difficulties caused by the system's features. In the *second phase* (free play), the game was played by two teams, but this time only one member was supposed to act within the playfield, while the other(s) stayed aside to provide support, holding a pencil and a notepad, useful for calculating complex arithmetic operations. In this sense, two rival players were acting in the playfield without any turn restrictions. In this phase, four 15-minute games were played by the first group and other five by the second, during which all participants acted both inside and outside the playfield. In the *third phase*, each group sat on the playfield's floor in hemicycle fashion in front of a set of short tables. The tables were used to conceal each participant's picture-cards, so as not to affect the choices of the others. The children were asked to evaluate the system's objectives listed in Table 13 through a series of questions (Section 6.4.3) addressed to them by a facilitator. In cases of low-score answers, the facilitator used the laddering method [57] to elicit further details on the problems perceived by the young evaluators.

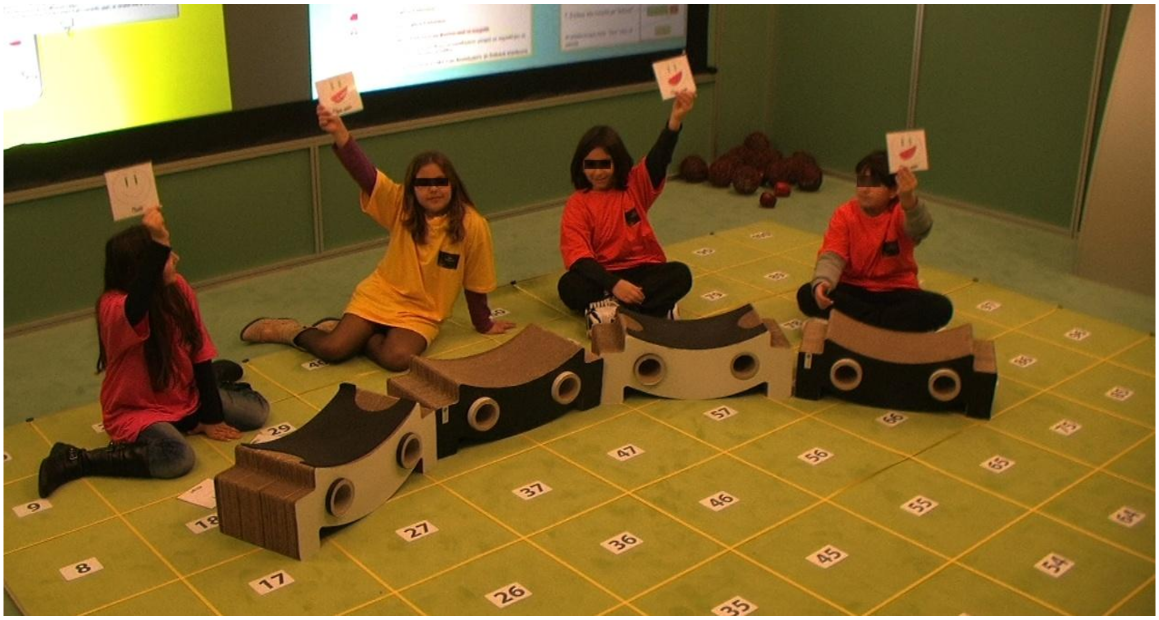


Figure 49. The evaluators' committee using the picture-cards

6.4. Results

This Section presents the results of the Apple Hunt evaluation. The results are organized in three parts, corresponding to the three evaluation phases described in Section 6.2. Within each part, the results are classified according to the evaluation parameters they respond to.

6.4.1. 1st phase (practice) Results

The first evaluation phase was based on the Think Aloud and Active Intervention methodology [67]. The results are reported below.

Usability problems at the cognitive level

- The younger participants did not seem to be comfortable with the arrangement of numbers on the playfield. Their decisions about the target positions were based mostly on counting the number of the interpolated positions, regardless of the numeric mark of their target.
- Self-orientation towards the right position was harder when a participant turned backwards (watching the playfield upside down).

Fantasy problems

- Especially the two youngest participants did not seem to relate subtraction and division to moving backwards. In contrast, they were inclined to use addition regardless of where they wished to move. This is also evident from the standard deviation recorded for Question A3 (see Section 6.4.3).

6.4.2. 2nd phase (free game) Results

The 2^{nd} phase's evaluation results were mainly collected by observation. In addition, the system's performance reports and controller logs were examined to extract information about the game's learnability and the controller's interface usability respectively.

Usability problems at the cognitive level

- Many participants did not cope well with standing still for at least 1 second in the target position, until it is registered by the system. Instead, they usually rushed to perform the next action.
- Invalid correlation of the apple (target) colour to the colour of the trap images in the virtual playfield display was often noticed to confuse participants who based their strategy on graphics.
- Some of the participants, especially the younger ones, experienced problems in calculating arithmetic operations including high numbers (e.g. 89 : 5). In general, the game seemed to favor players starting from lower numbers against those starting from higher numbers.

Usability problems at the physical level

- In general, two out of the nine participants experienced stress in their attempt to hit the most targets and get the winner's award.
- No particular problems related to this category were evident. Nevertheless, it should be taken into account that all children were fit and healthy, whereas a rather large percentage of Greek student of their age tend to be overweight.

Control problems

- The following problems were experienced during the use of the mobile controllers[12] (only the Calculator interface was evaluated, due to the arithmetic nature of the game):
    - Button interactivity was not sufficient. The participants did not seem to be always sure if their commands had been submitted.
    - Many participants tried to clear the command field before typing a new command, which was not necessary.
- "Body control" (positioning) problems included:
    - Difficulties in moving towards the right position when the participants' orientation was reverse (their back facing the visual display).
    - Standing still for the purpose of new position registration was often neglected.

[12] The usability problems of the mobile controller were also assessed with the aid of the controller logs (Section 3.3.4).

Fun

- Laughter and yelling were evident in all games. Even when the game was paused for technical reasons, the participants were eager to restart playing. On the other hand, no yawns or complaints were noticed.
- During the second phase (free game), the players outside the playfield did not stop supporting their partners, who were acting in it.
- Even when the evaluation process was finished in both sessions, all participants continued playing spontaneously until their parents had to leave.

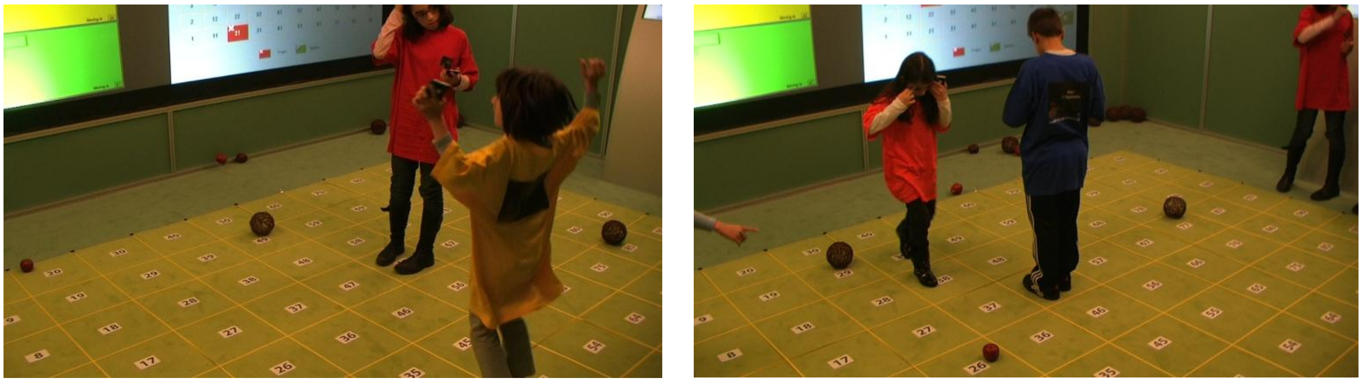


Figure 50. Evidence examples of *fun* and *collaboration* during free game in the children's reactions

Besides the above observations, another interesting related result is that all participants expressed the desire for the game to have a longer duration.

Collaboration

As described in Section 6.3, despite the competition taking place in the playfield, collaboration was realized among each player acting within the field and his/her partners located in the surrounding area. The following observations related to this category were made:

- A benefit of this collaboration style appeared in the participants' focus on their team's play rather than their rivals' performance. Despite no timer was set, they demonstrated a legitimate feeling of rush, attributed to the fact that they were not playing alone.
- Collaboration unexpectedly worked as a perfect "ice-breaker" among participants formerly unknown to each other, which pursued synergy under their common goal.

Kinesthetic activity

- All participants declared enthusiastic for using body motion during the game.
- Their kinesthetic activity involved body-turns, walking or running towards new positions and even jumping in order to register a new position (in contrast to the instructions for standing still for this purpose).
- In general, most children demonstrated eagerness for motion and very often moved (recklessly) right before submitting the required commands.

Learnability

- In general, the participants learned the game's behaviour fast. However, the introductory / practice phase was judged necessary.
- The usability problems of the controller's interface often delayed the game flow, but not so importantly as to affect the final results.
- The numbers' arrangement seemed to confuse a part of the participants, who faced difficulties in locating their target positions.
- The analysis of the system's performance reports showed that the rules were easily understood, as invalid moves and mistakes were minimal during both sessions.

6.4.3. 3rd phase (evaluators' committee) Results

The results of the third phase correspond to feedback offered by the participants through the committee processes. Laddering was used by the facilitator to elicit details on the problems they reported. The end of this Section presents the questions asked to the evaluators' committee, followed by an average score and standard deviation per question. It should be noted that the statistics are calculated according to the picture-cards score scale (Figure 47).

Usability problems at the cognitive level

- The arrangement of numbers was criticized by three participants, who presented their personal view of the ideal arrangement. In short, the arrangements mentioned are summarized to the: (i) horizontal zig-zag, (ii) vertical "snake" and (iii) horizontal "snake" arrangements[13] (Figure 51).

Challenge problems

- The older participant found the game too simple, whereas the younger perceived the gameplay as hard. For most of the participants, however, the game flow was reported to be "thought-provoking" and "interesting".

Fantasy problems

- The game was found childish or too hard by a few participants, as mentioned in the previous paragraph.

Curiosity problems

- The graphics did not appeal to all participants. Their language, arrangement and content were criticized by two of them. Most participants agreed that following up the visual display in parallel to the playfield arrangement was "tiring" and "time-consuming". Instead, it was suggested that the Apple Hunt game should be more independent of the visual displays.

[13] According to the "snake" arrangement, the position marks (numbers) follow a snake-tail order, whereas the one established by Apple Hunt forms a zig-zag order.

- A problem reported about the sound effects was that it was not clear whom they are designated for and thus their meaning was not understood in many cases.
- Also, the sound effect triggered upon each player's invalid move was reported to be annoying. It responded to the most usual sound heard during play and was attributed to the participants' carelessness with the positions' boundaries.

Collaboration

- The first group of participants agreed unanimously that they "liked the game's team-spirit".

Challenge

- The first group of participants characterized the game as challenging for "combining movement, thinking and team work during play".
- Judging by the opinions expressed regarding Question G1 (below), challenge varied among children of different age and cognition levels. Although challenge in general was judged as positive, some of the older children found the game childish, whereas the younger ones faced difficulties with their calculations and orientation within the playfield.
- A (6^{th}-grade) participant declared: "Even if I never liked Mathematics, I enjoyed this game".

Learnability

- Following up the visual display in parallel to the playfield was reported to be "tiring" and "time-consuming", whereas the majority agreed that watching the graphics was confusing during play.

Related Committee-Questions and Average Responses

The questions listed below are classified according to the evaluation parameter they measure. Evaluation parameters that are not represented in this list should be assessed with an alternative adopted technique.

Usability problems at the cognitive level / Learnability

A1. - Was it easy to know when you should move to a new position?

Average score: 4 **Standard deviation: 0.87**

not at all	a little	fairly	much	very much

A2. - Was it easy to understand where you should or should not target?

Average score: 4.56 **Standard deviation: 0.73**

not at all	a little	fairly	much	very much

A3. - Was it easy to decide which arithmetic operation you should use?

Average score: 3 **Standard deviation: 1.58**

not at all	a little	fairly	much	very much

Usability problems at the physical level / Learnability

B1. - Was it easy for you to find the right position on the playfield after a command?

Average score: 3.67 **Standard deviation: 0.71**

not at all	a little	fairly	much	very much

B2. - Was it easy for you to stand within the positions' boundaries (for as long as it took you to think)?

Average score: 4.78 **Standard deviation: 0.44**

not at all	a little	fairly	much	very much

Challenge problems

C1. - Was it easy for you to calculate the arithmetic operations?

Average score: 4.11 **Standard deviation: 0.93**

not at all	a little	fairly	much	very much

C2. - Was it easy for you to locate the targets?

Average score: 4.11 **Standard deviation: 0.78**

not at all	a little	fairly	much	very much

C3. - Did you feel bored?

Average score: 1.33 (*inverse concept*) **Standard deviation: 0.71**

not at all	a little	fairly	much	very much

Curiosity problems

D1. - Did you enjoy the visual display?

Average score: 4.44 **Standard deviation: 0.73**

not at all	a little	fairly	much	very much

D2. - Did you enjoy the music?

Average score: 4 **Standard deviation: 1.41**

not at all	a little	fairly	much	very much

D3. - Did you enjoy the sound effects?

Average score: 4.11 **Standard deviation: 1.36**

not at all	a little	fairly	much	very much

Control problems

E1. - Did you like the mobile phone?

Average score: 4.78 **Standard deviation: 0.44**

not at all	a little	fairly	much	very much

E2. - Was it easy for you to use the mobile phone, in order to target your positions?

Average score: 4.33 **Standard deviation: 0.71**

not at all	a little	fairly	much	very much

E3. - Was it easy for you to move around the playfield?

Average score: 4.33 **Standard deviation: 0.71**

not at all	a little	fairly	much	very much

Kinesthetic activity / Fun

F1. - Did you enjoy moving around, instead of sitting, during play?

Average score: 5 **Standard deviation: 0**

not at all	a little	failry	much	very much

Challenge / Fun

G1. - Did you like the game?

Average score: 4.11 **Standard deviation: 0.6**

not at all	a little	fairly	much	very much

Learnability

H1. - Was it easy to follow up the visual display?

Average score: 3.89 **Standard deviation: 0.93**

not at all	a little	fairly	much	very much

6.5. Discussion

The major findings of the evaluation process are analyzed below, whereas related future improvements are also suggested.

Fun: Although literature offers various approaches to quantifying *fun* for evaluation purposes (e.g., [30]), in the conducted experiment, *fun* measurement based on [23], according to which observing children's' expressions (e.g. Figure 50) has more to offer than expecting reliable responses from oral communication. The observations made during evaluation justify a feeling of entertainment that was mutually shared by the participants of both sessions. The main findings relating to this parameter include expressions of joy and laughter, enhanced by all participants' spontaneous will to continue playing at the end of both sessions. In addition, the acquired average score of Questions F1 and G1 attest the good feeling created during the game.

Challenge: The participants felt highly challenged in general, whereas the first group admitted their content for experiencing jointly "movement, thinking and team work during play". Besides, challenge was demonstrated both from participants acting inside and those outside the playfield, who were actively supporting their inside-partners. Moreover, some participants of different age appeared unevenly challenged. Some of the older participants considered the game too easy, whereas the two younger ones found it hard to calculate and orient within the playfield. The analysis of the system's performance reports showed that younger participants inclined to use mostly addition, whereas the others used all operations for targeting their desired positions. This finding indicates that the younger probably had not yet mastered the required breadth of knowledge from school. In addition, the fact that only a part of the older participants regarded the game as "too easy" leads to the conclusion that the cognition level also played a significant role on their perception of game complexity. Consequently, Apple Hunt has to address such a complexity scaling that will extensively accommodate the age and cognition level range of their target group.

Engagement: Considering Whitton's theory relating to games evaluation [63], engagement can be assessed through the examination of the following five factors: (i) Perception of challenge, (ii) perception of control, (iii) immersion, (iv) interest and (v) purpose ("the perceived value of the learning activity"). Accordingly, the following analysis describes various *engagement* interpretations of the game, depending on each participant's profile.

i. *Challenge*: In relation to challenge, *engagement* has been higher within the average ages and cognition levels, as described above.
ii. *Perception of control*: Considering that "body controls" (kinesthetic interaction) successfully outweighed the subordinate usability problems of the handheld controllers and induced enthusiasm, *engagement* originating from *control perception* is judged generally positive.
iii. *Immersion*: The observations described under the "Fun" and "Collaboration" parameters attest that the majority of the participants was absorbed in the game's action. On the other hand, the "simplicity" of the game perceived by the older children worked suppressive against their *engagement*.
iv. *Interest*: All participants demonstrated positive interest in the game regardless of their attitude towards Mathematics as a school subject. Multimodality played a significant role in this respect according to the recorded opinions.
v. *Purpose*: As in the case of *challenge*, the perceived value of the learning activity varied among the ages and cognition levels. Overall, it would seem that the lower the age and cognition level, the higher the perceived value of the activity.

Learnability: Independently of the age and cognition variables, learning the game did not pose difficulties on the participants. Performance reports showed that wrong actions were minimal, whereas the game's rules were easily understood during the practice phase. The children's age and cognition level was positively correlated with their understanding of the playfield's arrangement and their orientation within it. In addition, some of the participants declared confusion deriving from the graphical interactivity of the display's and the controller's interfaces.

Kinesthetic activity: The children appeared considerably active during the free-game phase of the evaluation. Their activity involved body-turns, walking or running towards new positions and even jumping, in order to register new positions (as opposed to the instructions provided for standing still for this purpose). Apple Hunt was proved to encourage a natural flow of body motion, which may not be constant - as players need to stand while submitting commands -, but is definitely intense whenever it is manifested. From the participant's aspect this kind of kinetic play was unanimously preferred to the typical sedentary style of play imposed by most computer games (Question F1).

Collaboration: Intense team work was demonstrated during the majority of the games played, which has been a critical factor of the players' immersion. It is believed that collaboration under a common goal worked as "ice-breaker" among them and also aided in that none of the young evaluators was stressed due to the presence of foreigners, thus unleashing *fun*. Even if *collaboration* was planned to be evaluated solely by observing the children's activities, the first group spontaneously mentioned their pleasure for "the game's team-spirit". However, further research will be required to examine collaboration in 2-vs.-2 games, during which the partners will all be acting within the playfield. As already mentioned (in Section 6.2), the evaluation setup described in this thesis focused only on 1-vs.-1 games.

Thinking and learning: Observing the gameplay created the impression of a highly competitive context (evident from the players' eagerness for move, yelling and seeking for help from the partners) in which players sensed that the more they mastered their thoughts the faster they would move. Thinking was supported by the presence of partners around the playfield, which helped players focus on their team work rather than on the rival's performance. This combination of thinking with kinesthetic activity and team work was referenced as "challenging" by the first group of participants. On the other hand, the thinking load was not equally distributed among the participants due to their varied age and knowledge. This is reflected in the negative comments recorded about the game's complexity (see "challenge problems" in the previous Section). Even if thinking is undoubtedly a prerequisite for *learning*, the evaluation of *learning quality* requires further testing in the long run, involving a stable sample of participants. The performance metrics set provided by AmI Game Floor could be used as a reference point towards this achievement.

6.6 Potential improvements of Apple Hunt

The challenge variance, discussed above, indicates that Apple Hunt's levels could be redesigned, in order to enhance the established complexity, which solely depends on the ratio of traps (obstacles) to apples (targets). This is possible through the use of the *Level* infrastructure provided by the system (Section 3.3.3). A related example of the game levels' design follows:

- **Level 1:** Easy

Only a part of the playfield will be used, restricted to low numbers. Also, only addition and subtraction will be used for practice purpose. Aborting a command will be allowed in this level. No obstacles will be set.

- **Level 2:** Medium

The full range of the playfield will be available, as well as all arithmetic operations. Again no obstacles will be set. Aborting a command will cost an amount of points.

- **Level 3:** Hard

A considerable amount of obstacles will be present in this level. Aborting a command will not be allowed any more.

Furthermore, interactivity appeared to require some improvements according to the evaluation outcomes. First, the controller's (Calculator) interface use revealed the need for more sophisticated button interactivity, whereas its command field should be cleared automatically upon submission. At the same time, the full arithmetic operation formed by each command should be kept visible in a different field. Secondly, since several participants suggested a higher degree of independency from the visual display, the controller's interface could be enhanced to hold more information, such as the score and current position. Thirdly, the sound effects have to be improved in a more personalized manner. As a headset controller is planned to be provided in the future, this problem will be solved (with the sound being heard only by the addressed player). Finally, jumping, as an act intended for registering a new position, was considerably popular among the participants. This leads to the assumption that jumping might be more natural for children than standing still for a second in such cases. Further testing is required to conclude whether this interaction pattern should be modified.

Understanding the playfield's numbers arrangement played a key role in the participants' successful play. While no clear conclusion was drawn in this respect, the opinions expressed about the ideal arrangement are illustrated in Figure 51, compared to the current one ("vertical zig-zag"). It should be noted that "snake" number-arrangements were mainly preferred by the younger participants, who based their thinking much on counting the positions in-between their location and their target.

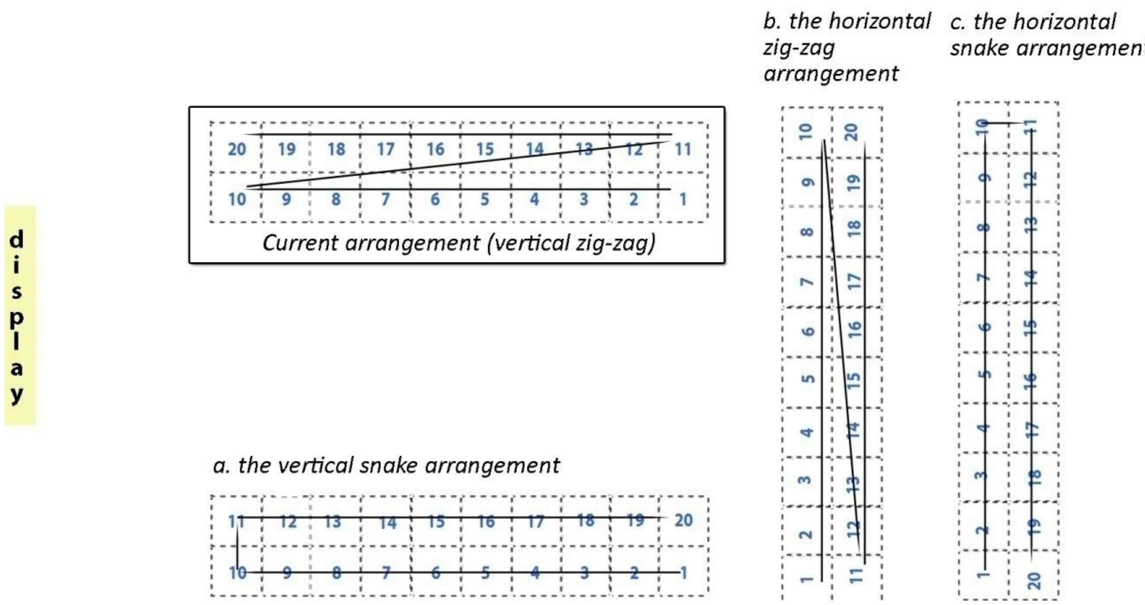


Figure 51. The three suggested number arrangements compared to the current one

6.7 Conclusion

Despite the fact that the evaluation revealed numerous strengths of Apple Hunt, as well as potential improvements, further assessment is required to attest and enrich the recorded results, due to the small sample of available participants. Moreover, considering that 7 out of the 9 participants were girls, there is a need for more gender-balanced groups in future evaluations. Other parameters that might have biased the recorded results are the participants' considerable experience with technology, as well as their good fitness and overall health.

Finally, the adopted evaluation technique proved joyful and productive. This is attributed to the high variety of evaluation methodologies used (think aloud and active intervention, observation, evaluators' committee with picture-cards and laddering) and the unbound collaborative play performed during the second phase. Further, the evaluators' committee idea (during the last phase), helped the children imprint experiences from the uninterrupted play and commit in fluent discussions both among them and with the facilitator, producing useful feedback. The picture-cards concept, basing on a similar idea in [4], appeared to be a widely approved tool by the children, as demonstrated by their desire to use the cards even when they were not required. On the other hand, a negative comment recorded with respect to the evaluation process was that half of the participants had to stay outside the playfield during free play. This arrangement was decided in order to promote collaborations and aid the players' decisions.

In summary, according to the above results, Apple Hunt, and consequently AmI Game Floor, proves to satisfy their purpose to a large extent. Their value lies in that they successfully promote kinesthetic activity, collaborative work and thinking in an entertaining multimodal environment. On the other hand, the drawbacks of their current implementation are related to the lack of complexity scaling and to interactivity issues.

7. CONCLUSIONS AND FUTURE WORK

7.1. Conclusions

This thesis has reported the learner-centred design, development and evaluation of a technological framework for learning applications, named AmI Game Floor, which constitutes an educative Ambient Intelligent (AmI) environment aimed at creating challenging learning conditions through play and entertainment.

AmI Game Floor provides an educational AmI environment within which users engage in kinesthetic and collaborative play, overlooked by unobtrusive technology. Its implementation includes a vision-augmented playfield, shaped as a grid of game positions, and a set of various User Interfaces offering multimodal interaction. Game control is performed both through movement around the playfield and the use of mobile phone GUIs, communicating with the system through the web. A remote game manager complements the system's controls, whereas a large visual display and a sound direction modality provide graphical and acoustic feedback respectively. Overall, the framework offers physical infrastructure, networking, 2D graphics, sound and control support for applications. Gameplay facilities of the system include profiling, mode switching, winner judgement, and performance analysis.

AmI Game Floor related to previous research work in several domains, including Educational AmI Environments, Interactive Floors, Kinesthetic Console Games and Playful Learning Applications. With respect to previous efforts, AmI Game Floor emphasizes low-cost kinesthetic and collaborative technology in *natural* playful learning, while embodying an innovative performance measurement system. Providing an appropriate framework for developing learning applications with extended customization facilities, the system supports a wide breadth of educational subjects and concepts.

The system depends on middleware connectivity [19] to orchestrate its various scattered modalities. In particular, it implements the *GameFloor* service, which determines the whole communication with its applications, and uses the *localization* [66] and *soundplayer* [70] services for multiple user-tracking and sound direction purposes. The system also provides applications with infrastructure customization facilities, described in XML, and a MySQL database for game information storage and performance analysis. The graphical user interfaces (GUIs) were developed in C# (visual display), ASP.NET and javascript (handheld controllers), whereas the game sound is enhanced with speech synthesis for natural interactivity.

Addressing the objective of uncovering the system's potential to encourage kinesthetic and collaborative play, a prototype of "Apple Hunt" was developed in a laboratory space of ICS-FORTH. "Apple Hunt" is an educational game designated for elementary school children, which aims at exercising players in fundamental arithmetic operations. The game's scenario requires players to move around the playfield, consisting of a grid of enumerated positions, in order to pick as many (dummy) apples as possible, avoiding the (dummy) traps within. In order to move around, each player has to input the right arithmetic operation (using the mobile controller), the results of which must match to the number of their target position. Apart from "Apple Hunt", three more application designs were discussed to uncover the flexibility offered by the framework for accommodating the development diverse applications.

The evaluation of Apple Hunt intended to assess the effectiveness of the system's technological features that encourage and contribute to learning. In particular, its evaluation was conducted in two sessions, involving 9 children in total, aged from 7 to 11 years old. A wide variety of evaluation methodologies were practiced, including *think aloud and active intervention*, *observation* and *laddering*, whereas the system's performance reports and controller logs were also examined for extracting behavioural patterns of the participants. Furthermore, *children committees,* a fresh approach to evaluation with children, influenced by [4], were introduced. Children committees are settled at the end of the testing process, during which participants are equipped with sets of picture-cards representing an expressive Likert-like score scale. In particular, participants are asked to evaluate a series of issues related to their experimentation, using the appropriate card each time. The application of this methodology is believed to bring an interesting potential in the context of Ambient Intelligent Environments and educational games, as demonstrated by the children's forward discussions and high engagement during the conducted process.

The realized evaluation, besides providing useful feedback on the game itself and its current implementation, also allowed drawing some conclusions regarding the added value of the overall AmI Game Floor framework. Their achievement appeared to lie in that they successfully promote kinesthetic activity, collaborative work and thinking in an entertaining multimodal environment. On the other hand, potential improvements of their current implementation are related to widening complexity scaling and to interactivity issues.

In conclusion, this thesis constitutes an encouraging step towards the adoption of entertaining educative practices for young learners that not simply foster their cognitive evolvement, but also benefit their emotional, social and physical development.

7.2. Future Work

The future priorities set for the system in the short run include:

- Reworking the drawbacks of the current implementation observed during the evaluation process (Apple Hunt's lack of complexity scaling, interactivity issues of the framework);
- Enhancing natural interactivity through speech and gesture recognition;
- Advancing the system's support for learning applications;
- Performing a subsequent full-scale evaluation.

The redesign of the Apple Hunt game's complexity will exploit the *Level* infrastructure provided by the framework (Section 3.3.3) to offer scalable variance of functionalities and difficulty per level. Interactivity modifications will emphasize higher independence from the visual display in the game action, rendering the controller GUIs' design more advanced. Moreover, the young evaluators' suggestions related to alternative playfield arrangements will be carefully examined under the current and different game scenarios. A detailed analysis of the aforementioned redesign is presented in Section 6.6.

Towards advancing interactivity to more intuitive levels, a headset controller, enhanced with speech recognition software, is planned to be used for including vocal commands. Further, embodying gesture recognition, AmI Game Floor will enhance its kinesthetic potential through more complex interaction patterns, while players will be offered with a wider variety of control modalities.

With respect to the system's features, a greater support for alphabetic type applications will be provided through the integration of a dictionary service and a word-editor (for input of quested words), which will also accommodate a wider statistics range in this category. The alternative scenarios discussed in Section 5.2 will be used as a basis for requirements elicitation for this purpose. Moreover, round-based games will be also supported by the framework, in order to enlarge its perspective for application scenarios.

Full-scale evaluation will be conducted to assess the system's effectiveness in detail. Provided that the results obtained so far depended on a small sample, not particularly balanced in age and cognition level, future sessions are expected to reveal more solid insights. Another related issue derives from the fact that the performed evaluation did not test the players' activity under full game capacity, according to the needs of the adopted process design. In addition, diverse applications need to be put to the test, in order to shed light on the system's applicability equally to non-arithmetic educational content.

Finally, the *learning quality* offered by the system is planned to be evaluated on the long run. Towards this objective, longitudinal tests involving a stable sample of young participants will be performed, utilizing the performance reports produced upon each test as their progress milestones during the whole process.

8. REFERENCES

[1] Aarts, E. and Encarnacao, J. L. (2006) Into Ambient Intelligence. True Visions – The Emergence of Ambient Intelligence, Springer.

[2] Alcañiz, M. and Rey, B. (2005) New Technologies For Ambient Intelligence. In The Evolution of Technology, Communication and Cognition Towards the Future of Human-Computer Interaction. Amsterdam: IOS Press; p. 3-15

[3] Anderson, F., Annett, M. and Bischof, W. (2010) Lean on Wii: physical rehabilitation with virtual reality Wii peripherals. Studies in Health Technology and informatics, 154:229-234.

[4] Barendregt, W., Bekker, M. M. (2005) Development and Evaluation of the Picture Cards Method. Workshop Interaction Design for Children, Interact 2005 Rome, Italy.

[5] Barendregt, W., Bekker, M. M., Speerstra, M. (2003) Empirical evaluation of usability and fun in computer games for children. Proceedings of Human-Computer Interaction INTERACT-03', 3 September 2003, IOS Press, Zürich, Switzerland, 705-708.

[6] Benford, S., Bederson, B., Akesson, K. P., Bayon, V., Druin, A., Hansson, P., Hourcade, J.P., Ingram, R., Neale, H., O'Malley, C., Simsarian, K.T., Stanton, D., Sundblad, Y., Taxen, G. (2000) Designing Storytelling Technologies to Encourage Collaboration Between Young Children. Proceedings of ACM Conference on Human Factors in Computer Human Interaction CHI'2000 (The Hague, 1-6 April 2000), ACM Press, New York, pp. 556–563.

[7] Brooks A.L., Petersson, E. (2005). Play Therapy Utilizing the Sony EyeToy. In: Slater, M. (ed.) Proc. of the Eighth International Workshop on Presence, pp. 303-314.

[8] Chow Y.W. (2008) "The Wii Remote as an input device for 3D interaction in immersive head-mounted display virtual reality," in Proc. IADIS International Conference Gaming 2008: Design for Engaging Experience and Social Interaction, Amsterdam, The Netherlands, pp. 85–92.

[9] Cordes, C., and Miller, E. (Eds.) (2000) Fool's Gold: A Critical Look at Computers in Childhood. Alliance for Childhood, College Park, MD; http://drupal6.allianceforchildhood.org/fools_gold

[10] Dede, C. (1995) "Testimony to the US Congress, House of Representatives Joint Hearing on Educational Technology in the 21st Century. Committee on Science and Committee on Economic and Educational Opportunities October 12, 1995. Retrieved on February 14, 2011, from http://newhorizons.org/strategies/technology/dede1.htm

[11] Dillenbourg, P. (1999). Collaborative Learning: Cognitive and Computational Approaches. Advances in Learning and Instruction Series. New York, NY: Elsevier Science, Inc.

[12] Druin, A., Bederson, B., Boltman, A., Miura, A., Knotts-Callahan, D., & Platt, M. (1999). Children as our technology design partners. A. Druin (Ed.), The design of children's technology (pp. 51-72). San Francisco, CA: Morgan Kaufmann.

[13] Druin, A., Hourcade J.B. (2003) Interaction design and children, COMMUNICATIONS OF THE ACM, January 2005/Vol. 48, No. 1

[14] Ducatel, K., Bogdanowicz, M., Scapolo F., Leijten J., and Burgelman, J.-C. (2001) ‘‘Scenarios for ambient intelligence in 2010,’’ Technical report, Information Society Technologies, European Commission.

[15] Erikson, K.A., Simon, H.A. (1984) Protocol Analysis: Verbal Reports on Data. MIT Press, Cambridge, MA

[16] Ertmer, P and Newby T.J. (1993) Behaviourism, Cognitivism, Constructivism: Comparing critical features from an instructional design perspective. Performance Improvement Quarterly 6(4), 50-71.

[17] Fogtmann, M.H., Fritsch, J. and Kortbek, K.J. (2008) Kinesthetic Interaction - Revealing the Bodily Potential in Interaction Design OZCHI: Conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-Human Interaction, ACM, Cairns, Australia.

[18] Gardner, H. (1993) Frames of Mind: The theory of multiple intelligences. In: Britain by Fontana Press, 2nd edn., p. 466. Basic Books, New York

[19] Georgalis, Y., Grammenos, D., Stephanidis, C. (2009) Middleware for ambient intelligence environments: Reviewing requirements and communication technologies. In: UAHCI 2009, San Diego, CA, United states, pp. 168–177. Springer, Heidelberg.

[20] N. Griffith and M. Fernstrom, (1998) “LiteFoot: A Floor Space for Recording Dance and Controlling Media,” Proceedings of the 1998 International Computer Music Conference, International Computer Music Association, San Francisco, CA (October 1998), pp. 475–481. See also http://www.ul.ie/;pal/litefoot/Index.htm.

[21] Grønbæk, K., Iversen, O.S., Kortbek, K.J., Nielsen, K.R., Aagaard, L. (2007) Interactive Floor Support for Kinesthetic Interaction in Children Learning Environments. In Proc. of INTERACT 2007, September 10-14, 2007, Rio de Janeiro, Brazil. Springer Verlag.

[22] Haan, G., Griffith, E. J., Post, F. H. (2008) Using the wii balance boardTMas a low-cost VR interaction device. In VRST ’08: Proceedings of the 2008 ACM symposium on virtual reality software and technology, pages 289–290, New York, NY, USA. ACM.

[23] Hanna, L., Risden, K., & Alexander, K.J. (1997). Guidelines for usability testing with children. Interactions, September–October, 9–14.

[24] Hanna, L., Risden, K., Czerwinski, M., & Alexander, K. J. (1998). The role of usability research in designing children’s computer products. http://www.reserach.microsoft.com/users/marycz/druin98.htm.

[25] Healy, J. (1998) Failure to Connect: How Computers Affect Our Children's Minds and What Do We Do About It, Simon & Schuster, New York.

[26] Hämäläinen, P., and Höysniemi, J. (2002). A Computer Vision and Hearing Based User Interface for a Computer Game for Children. In Proceedings of the 7th ERCIM Workshop "User Interfaces For All", 23-25 October 2002, Paris.

[27] Höysniemi, J., Hämäläinen, P., Turkki, L. (2003) Using peer tutoring in evaluating the usability of a physically interactive computer game with children. Interacting with Computers, Vol. 15, Issue 2, 205-225.

[28] Höysniemi, J. (2006). International survey on the Dace Dance Revolution game. ACM Computers in Entertainment, 4(2), 1-30.

[29] Höysniemi, J., Hämäläinen, P., Turkki, L. and Rouvi, T. (2005) Children's intuitive gestures in vision-based action games. Communications of the ACM, 48(1), 44-50.

[30] Hunicke, R., LeBlanc, M. & Zubek, R. (2004) MDA: A Formal Approach to Game Design and Game Research. Game Design and Tuning Workshop at the Game Developers Conference, San Jose.

[31] Johnson, R. T., & Johnson, D. W. (1986). Action research: Cooperative learning in the science classroom. Science and Children, 24, 31-32.

[32] Jordan, P.W. (2000) Designing Pleasurable Products: An Introduction to the New Human Factors, Taylor and Francis, UK.

[33] Karime, A., Hossain, M. A., El Saddik, A., and Gueaieb, W. (2008) "A multimedia-driven ambient edutainment system for the young children." In Proceedings of the 2nd ACM Workshop on Story Representation, Mechanism and Context. ACM, New York, NY, 57-64.

[34] Keating, N.H. (2007), The Lambent Reactive: an audiovisual environment for kinesthetic playforms. Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME07), New York, NY, USA

[35] Kligyte, G., Leinonen, T. (2001) D3.1 Study of functionality and interface of existing CSCL/CSCW systems. ITCOLE Project
http://www.euro-cscl.org/site/itcole/public_deliverables_html

[36] Krogh, P.G., Ludvigsen, M., Lykke-Olesen, A. (2004) Help me pull that cursor - A Collaborative Interactive Floor Enhancing Community Interaction. In proceedings of OZCHI 2004, 22-24 November, 2004 at the University of Wollongong, Australia. CD-ROM. ISBN:1 74128 079.

[37] Lakkala, M., Rahikainen, M. & Hakkarainen, K. (Eds.) (2001). D2.1 Perspectives of CSCL in Europe: A Review. Helsinki: University of Art & Design

[38] Lee, J. C. (2008) Hacking the Nintendo Wii Remote, IEEE Pervasive Computing, 7, 39-45.

[39] Lee , J. C., Kim, H., Gupta G., Mazalek, A (2008) "WiiArts: Creating collaborative art experience with WiiRemote Interaction," in Proc. Of the 2nd International Conference on Tangible and Embedded Interaction (TEI'08), Bonn, Germany, 2008, pp. 33–36. 2nd International Conference on Tangible and Embedded Interaction (TEI'08). Bonn, Germany, pp. 33-36.

[40] Leikas, J., Väätänen, A., and Räty, V. (2000) Virtual space computer games with a floor sensor control – human centred approach in the design process. In Haptic Human – Computer Interaction, in Proceedings of First International Workshop on Haptic Human-Computer Interaction (Glasgow, UK, August – September, 2000). Springer, 199-204.

[41] Lund, H.H., Jessen, C. (2005) Playware intelligent technology for children's play. Technical Report TR-2005-1, Maersk Institute For Production Technology, University of Southern Denmark.

[42] Markopoulos, P. & Bekker, M. (2003a) "Editorial: Interaction design and children" In Interacting with Computers 15 (2003), pp. 141-149.

[43] Marti, P., Lund, H. H. (2003) Ambient Intelligence Solutions for Edutainment Environments, In Proc of AI*IA (Eight National Congress of Associazione Italiana per l'Intelligenza Artificiale), Springer-Verlag.

[44] Mazzone, E., Xu, D., Read, J. (2007) "Design in Evaluation: Reflections on Designing for Children's Technology" in Proceedings of the British HCI conference 2007, Lancaster, UK. ACM Press.

[45] McFarlane, A., Sparrowhawk, A. And Heald, Y. (2002) Report on the educational use of games: An exploration by TEEM of the contribution which games can make to the education process [online]. Retrieved on February 14, 2011, from: http://www.teem.org.uk/publications/teem_gamesined_full.pdf .

[46] Miyake, N. (1986), Constructive Interaction and the Iterative Process of Understanding. Cognitive Science 10(2), 151–177.

[47] Monk, A. (1993) Improving your human computer interface: a practical approach. Prentice Hall International, Englewood Cliffs

[48] Moser, T.: Bevægelse i sproget – sproget i bevægelse (in Danish) http://kidlld.dk/temaer/krop/beveagelse/ PloneArticle_view (iGameFloor)

[49] A. Ndiaye, P. Gebhard, M. Kipp, M. Klesen, M. Schneider, W. Wahlster (2005) Ambient Intelligence in Edutainment: Tangible Interaction with Life-Like Exhibit Guides, in: Proceedings Conference on INtelligent TEchnologies for interactive entertainment (INTETAIN'05), Madonna di Campiglio, Italy.

[50] Object Management Group (1998) The Common Object Request Broker: Architecture and Specification. Object Management Group, Framingham, MA

[51] Pagulayan, R., Keeker, K., Wixon, D., Romero, R., Fuller, T. (2003) User-centered design in games. In The Human-Computer Interaction Handbook: Fundamentals, Evolving Techniques and Emerging Applications. J. A. Jacko and A. Sears (eds.). Lawrence Erlbaum Associates, Mahwah, NJ, 883-905.

[52] J. Paradiso, C. Abler, K. Hsiao, and M. Reynolds (1997) The Magic Carpet: Physical Sensing for Immersive Environments. Proceedings of 1997 Annual International Conference on Human Factors in Computing Systems, Atlanta, GA, ACM Press, New York, pp. 277–278. 67.

[53] Piaget, J. (1954) The Construction of Reality in the Child. New York: Basic.

[54] Price S., Rogers Y., Scaife M., Stanton D., Neale (2003) Using tangibles to promote novel forms of playful learning. Interacting with Computers 15, 169–185.

[55] Rand D, Kizony R, Weiss P. L. (2008) Sony PlayStation II EyeToy: low-cost virtual reality for use in rehabilitation. JNPT 2008; 32:155–63.

[56] Resnick, M. (2004) Edutainment? No thanks. I prefer playful learning. In Associatzione Cicita. 1, 1, 2-4.

[57] Reynolds, Thomas J. and Jonathan Gutman (1988), "Laddering Theory, Method, Analysis, and Interpretation," Journal of Advertising Research, 28, 1, 11-31.

[58] G. Riva, F. Vatalaro, F. Davide, M. Alcañiz (Eds.) (2005) Ambient Intelligence. IOS Press. http://www.ambientintelligence.org

[59] Roberts CK, Freed B, McCarthy WJ (2010) Low aerobic fitness and obesity are associated with lower standardized test scores in children. The Journal of Pediatrics, 156:711–718.

[60] Roschelle, J. M., Pea, R. D., Hoadley, C. M., Gordin, D. N.,&Means, B. M. (2000) Changing how and what children learn in school with computer-based technologies. The Future of Children: Children and Computer Technology, 10(2), 76–101.

[61] Subrahmanyam, K., Kraut, R. E., Greenfield, P. M., Gross, E. F. (2000) The Impact of Home Computer Use on Children's Activities and Development, The Future of Children: Children and Computer Technology. 10 (2).

[62] Vygotsky, L. (1978). Mind in society: The development of higher psychological processes. Cambridge: Harvard University Press.

[63] Whitton, N. (2010) Learning with Digital Games. A Practical Guide to Engaging Students in Higher Education. Routledge.

[64] Xu, D., Read, J. C., Mazzone, E. S., MacFarlane, Brown M. (2007) Evaluation of Tangible User Interfaces (TUIs) for and with Children – Methods and Challenges, in Human-Computer Interaction. Interaction Platforms and Techniques, vol. 4551/2007, Lecture Notes in Computer Science. Berlin / Heidelberg: Springer, pp. 1008-1017.

[65] Zabulis, X., Grammenos, D., Sarmis, T., Tzevanidis, K., Argyros, A. A. (2010) Exploration of large-scale museum artifacts through non-instrumented, location-based, multi-user interaction, In VAST International Symposium on Virtual Reality, Archaeology and Cultural Heritage. Palais du Louvre, Paris, France, 21-24 September 2010.

[66] Zabulis, X., Sarmis, T., Tzevanidis, K., Koutlemanis, P., Grammenos, D., Argyros, A. A. (2010) A platform for monitoring aspects of human presence in real-time, International Symposium on Visual Computing, Las Vegas, Nevada, USA, November 29 - December 1, 2010.

[67] Zaman, B. (2005). Evaluating games with children. Paper presented at the Proceedings of Interact 2005 Workshop on Child computer Interaction: Methodological Research, Rome, Italy.

[68] Zaman, B. (2008). Introducing contextual laddering to evaluate the likeability of games with children. Cognition, Technology & Work, 10(2), 107-117.

[69] Zurita G., Nussbaum, M. Mobile CSCL Applications Supported by Mobile Computing, Workshop Multi-Agent Architectures Supporting Distributed Learning in a Wired and Wireless Future, http://julita.usask.ca/mable/Accepted.htm #Submission6, Jan. 2003.

[70] Griffin, A., Tzagkarakis, Ch., Hirvonen, T., Mouchtaris, A., & Tsakalides, P. (2009). Exploiting the Sparsity of the Sinusoidal Model Using Compressed Sensing for Audio Coding. 2009 Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS "09). Saint-Malo, France

[71] What is User-Centered Design? Usability Professionals' Association, retrieved on February 14, 2011, from
www.usabilityprofessionals.org/usability_resources/about_usability/what_is_ucd.html

[72] “About scrabble” – The Official Scrabble Game From Hasbro, retrieved on February 14, 2011, from www.hasbro.com/scrabble/en_US/aboutScrabble.cfm

[73] Barbara Leigh Smith and Jean T. MacGregor – “What is Collaborative Learning?”, retrieved on February 14, 2011, from
http://learningcommons.evergreen.edu/pdf/collab.pdf

[74] Mission to learn – “What is learning”, retrieved on February 14, 2011, from http://www.missiontolearn.com/2009/05/definition-of-learning/

[75] Peripherals: - Playstation Eye, retrieved on February 14, 2011, from
http://uk.playstation.com/ps3/peripherals/detail/item78698/PlayStation%C2%AEEye/

[76] Wii at Nintendo, retrieved on February 14, 2011, from
http://www.nintendo.com/wii

[77] Kinect – Xbox.com, retrieved on February 14, 2011, from
http://www.xbox.com/en-US/kinect

[78] Dance Dance Revolution for Wii, PS2, PS3, Xbox 360, & PC, , retrieved on February 14, 2011, from http://www.ddrgame.com/

9. APPENDICES

APPENDIX A – Vocal Commands Vocabulary

EN	EL
Monitor Commands	
Game	Παιχνίδι
Mode	Είδος
Level	Επίπεδο
Volume	Ήχος
Effects	Εφέ
Play	Παίξε[14]
Start	Εκκίνηση[15]
Exit	Έξοδος
Ready	Έτοιμοι
Game Over	Τέλος παιχνιδιού (ή Τέλος)
New game[16]	Νέο παιχνίδι (ή Νέο)
Restart	Επανεκκίνηση[17]
Show Report (ή Report)	Αναφορά
Previous	Προηγούμενο
Next	Επόμενο
Ok	Οκ
Cancel	Άκυρο
Game Commands	
One, Two, … , Nine (1 – 9)	Ένα, Δύο, ... , Εννιά
Plus(+)	Συν
Minus (-)	Πλην
Times (x)	Επί
By (:)	Δια
Login	
[User name]	

[14] Σε φυσική γλώσσα θα ήταν «Πάρτε θέσεις»
[15] Σε φυσική γλώσσα θα ήταν «Παίζουμε»
[16] *Αν υπάρχει περίπτωση σύγχισης με το "Game", ας γίνει σκέτο "New"*
[17] *Αν υπάρχει περίπτωση σύγχισης με το "Εκκίνηση", ας γίνει "Ξανά"*

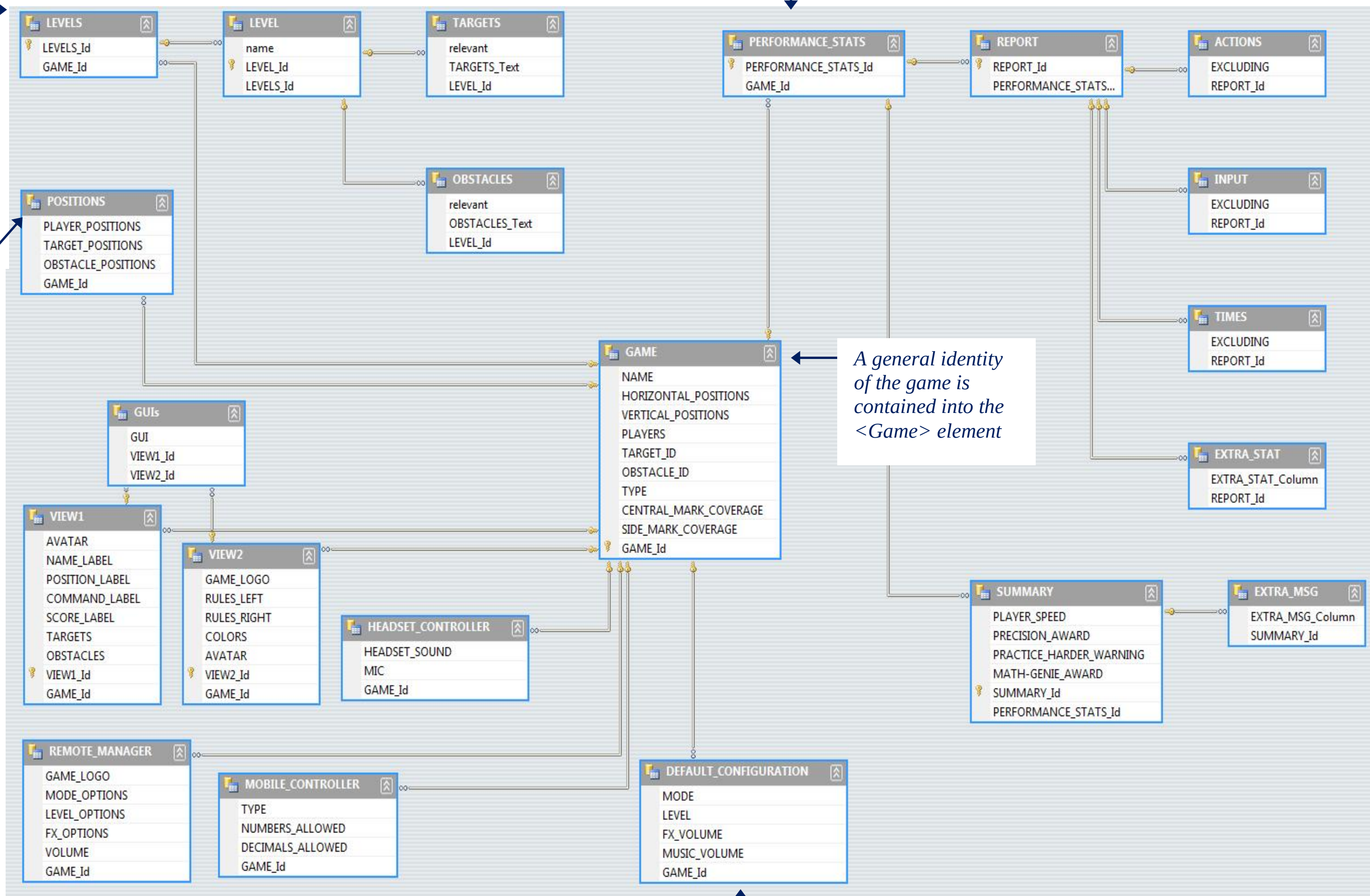
Performance statistics description, divided into the Report and the Summary statistics categories

APPENDIX B – The Features Schema

Levels description: Each level is described by a name and the number of Targets and / or Obstacles possibly involved.

Specified game positions can be optionally defined by the <Positions> element.

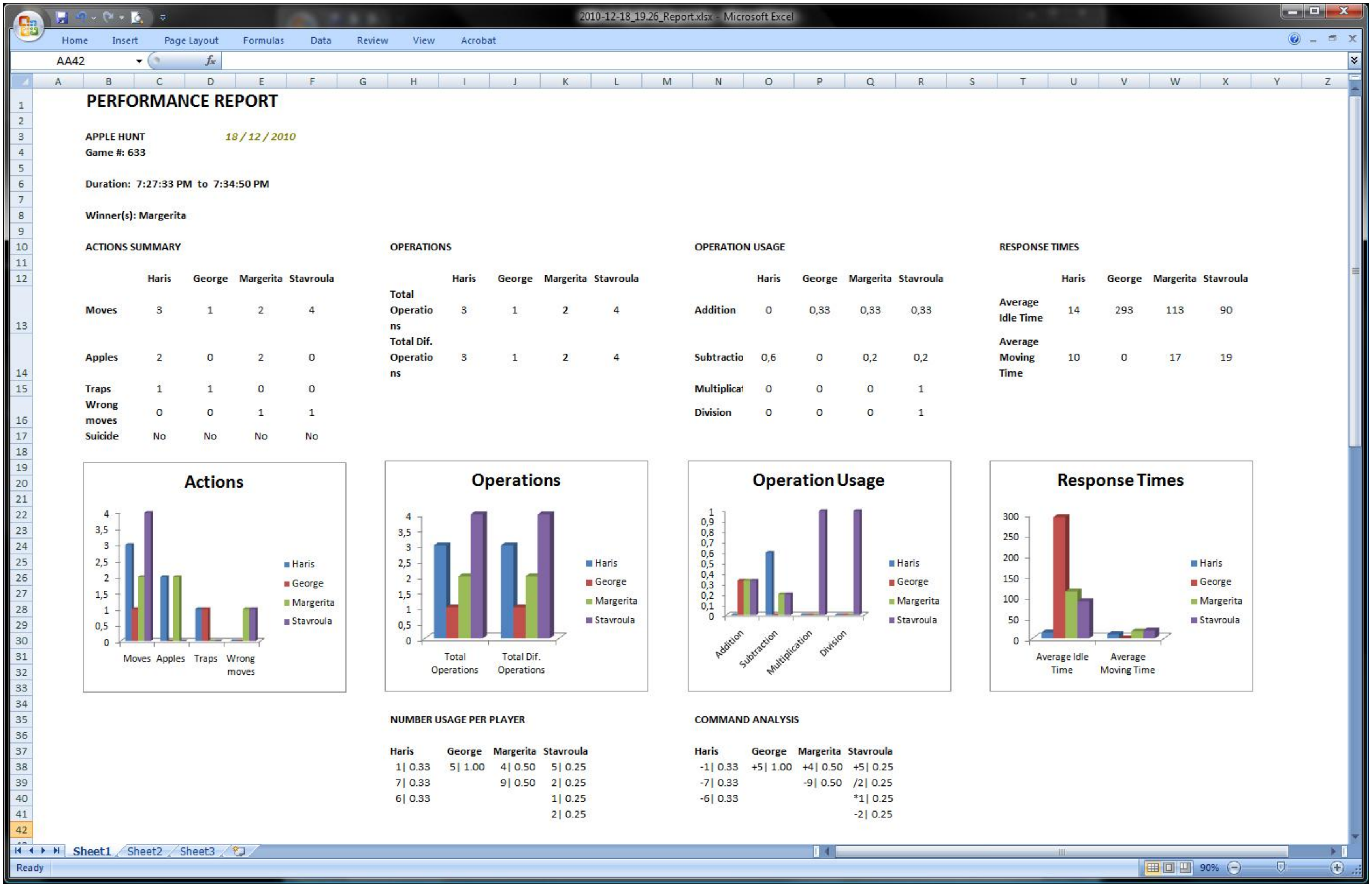
Customizable features per user interface



Statistic categories may be wholly adopted, shrunk by the <Excluding> element or complemented by the <Extra_Stat> and the <Extra_msg> elements.

A default-configuration option is provided by this element.

APPENDIX C – Performance Report Typical Layout (in Excel format)



2010-12-18_19.26_Report.xlsx - Microsoft Excel

PERFORMANCE REPORT

APPLE HUNT *18 / 12 / 2010*
Game #: 633

Duration: 7:27:33 PM to 7:34:50 PM

Winner(s): Margerita

ACTIONS SUMMARY

	Haris	George	Margerita	Stavroula
Moves	3	1	2	4
Apples	2	0	2	0
Traps	1	1	0	0
Wrong moves	0	0	1	1
Suicide	No	No	No	No

OPERATIONS

	Haris	George	Margerita	Stavroula
Total Operations	3	1	2	4
Total Dif. Operations	3	1	2	4

OPERATION USAGE

	Haris	George	Margerita	Stavroula
Addition	0	0,33	0,33	0,33
Subtractio	0,6	0	0,2	0,2
Multiplicat	0	0	0	1
Division	0	0	0	1

RESPONSE TIMES

	Haris	George	Margerita	Stavroula
Average Idle Time	14	293	113	90
Average Moving Time	10	0	17	19

NUMBER USAGE PER PLAYER

Haris	George	Margerita	Stavroula
1| 0.33	5| 1.00	4| 0.50	5| 0.25
7| 0.33		9| 0.50	2| 0.25
6| 0.33			1| 0.25
			2| 0.25

COMMAND ANALYSIS

Haris	George	Margerita	Stavroula
-1| 0.33	+5| 1.00	+4| 0.50	+5| 0.25
-7| 0.33		-9| 0.50	/2| 0.25
-6| 0.33			*1| 0.25
			-2| 0.25

Sheet1 Sheet2 Sheet3

Ready

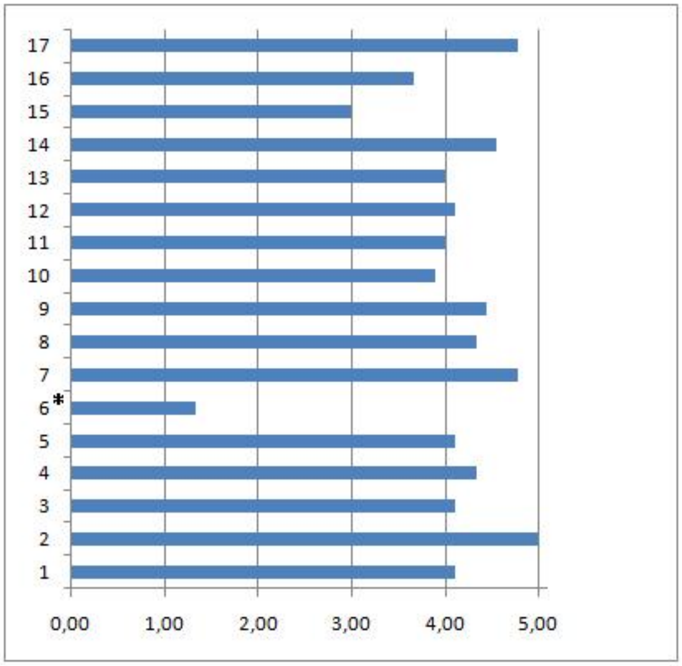
APPENDIX D – Controller log instances (from 2 different controllers)

APPENDIX E – Evaluators' Committee Results

	User	A	B	C	D	E	F	G	H	I
	School grade	5th	6th	6th	6th	6th	3rd	6th	2nd	6th
	Gender	female	female	male	female	female	female	male	female	female
No	Question	Answers								
1	Did you like the game?	4	4	5	5	4	4	3	4	4
2	Did you enjoy moving around, instead of sitting, during play?	5	5	5	5	5	5	5	5	5
3	Was it easy for you to calculate the arithmetic operations?	4	5	5	5	4	3	5	3	3
4	Was it easy for you to move around the playfield?	5	5	5	5	3	4	4	4	4
5	Was it easy for you to locate the targets?	5	5	5	4	3	3	4	4	4
6	Did you feel bored?	1	1	1	1	1	3	2	1	1
7	Did you like the mobile phone?	5	5	5	5	5	5	4	5	4
8	Was it easy for you to use the mobile phone, in order to target your positions?	5	4	5	5	4	4	5	3	4
9	Did you enjoy the visual display?	5	5	5	5	4	4	3	5	4
10	*Was it easy to follow up the visual display?*	5	4	5	5	3	3	4	3	3
11	Did you enjoy the music?	3	5	5	5	3	5	1	5	4
12	Did you enjoy the sound effects?	5	5	5	5	4	1	5	4	3

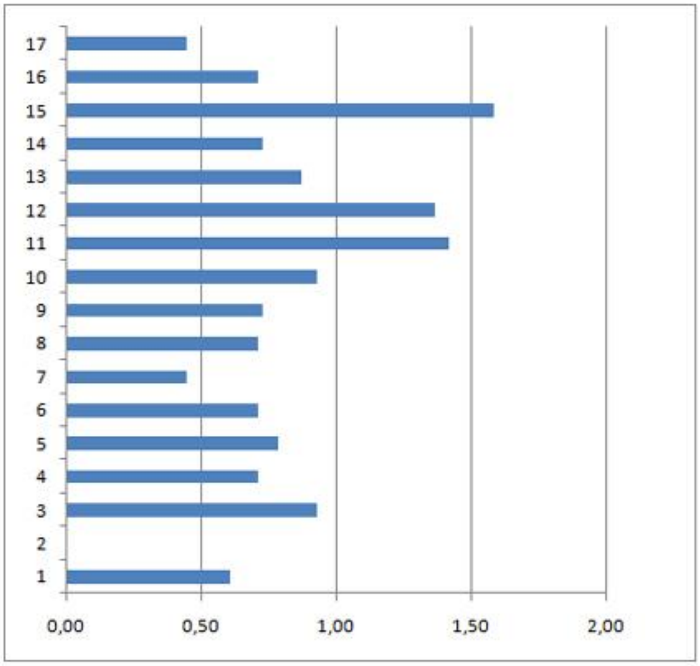
13	*Was it easy to know when you should move to a new position?*	5	5	5	4	3	3	4	4	3
14	*Was it easy to understand where you should or should not target?*	5	5	5	5	4	3	5	5	4
15	*Was it easy to decide which arithmetic operation you should use?*	1	4	5	1	4	1	4	4	3
16	Was it easy for you to find the right position on the playfield after a command?	3	4	5	3	4	4	3	4	3
17	Was it easy for you to stand within the positions' boundaries?	5	5	5	5	5	4	4	5	5

average score
(1: "not at all" | 5="very much")



*** Reverse meaning (1 - max | 5 - min)**

standard deviation



APPENDIX F – Application Features

Apple Hunt Features


```xml
<?xml version="1.0" encoding="utf-8"?>
<GAME xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Features.xsd">
  <NAME>Apple Hunt</NAME>
  <HORIZONTAL_POSITIONS>10</HORIZONTAL_POSITIONS>
  <VERTICAL_POSITIONS>10</VERTICAL_POSITIONS>
  <PLAYERS>4</PLAYERS>
  <TARGET_ID>Apples</TARGET_ID>
  <OBSTACLE_ID>Traps</OBSTACLE_ID>
  <TYPE>sequenced numeric</TYPE>
  <CENTRAL_MARK_COVERAGE>100</CENTRAL_MARK_COVERAGE>
  <LEVELS>
    <LEVEL name="Easy">
      <TARGETS relevant="true">3</TARGETS>
      <OBSTACLES>4</OBSTACLES>
    </LEVEL>
    <LEVEL name="Medium">
      <TARGETS relevant="true">3</TARGETS>
      <OBSTACLES>10</OBSTACLES>
    </LEVEL>
    <LEVEL name="Hard">
      <TARGETS relevant="true">2</TARGETS>
      <OBSTACLES>15</OBSTACLES>
    </LEVEL>
    <LEVEL name="Demo">
      <TARGETS>3</TARGETS>
      <OBSTACLES>5</OBSTACLES>
    </LEVEL>
  </LEVELS>
  <VIEW1>
    <GUIs>
      <GUI>Login activity</GUI>
      <GUI>Player performance</GUI>
      <GUI>Summary and report</GUI>
    </GUIs>
    <AVATAR></AVATAR>
    <NAME_LABEL></NAME_LABEL>
    <POSITION_LABEL>Position</POSITION_LABEL>
    <COMMAND_LABEL>Command</COMMAND_LABEL>
    <SCORE_LABEL>Current Score</SCORE_LABEL>
    <TARGETS></TARGETS>
    <OBSTACLES></OBSTACLES>
  </VIEW1>
  <VIEW2>
    <GUIs>
      <GUI>Selected game rules</GUI>
      <GUI>Starting positions</GUI>
      <GUI>Virtual playfield</GUI>
    </GUIs>
    <RULES_LEFT>Rules.png</RULES_LEFT>
    <RULES_RIGHT>controller_instr.png</RULES_RIGHT>
  </VIEW2>
  <REMOTE_MANAGER>
    <MODE_OPTIONS/>
    <LEVEL_OPTIONS/>
    <FX_OPTIONS/>
    <VOLUME/>
 </REMOTE_MANAGER>
 <MOBILE_CONTROLLER>
   <TYPE>numeric</TYPE>
   <NUMBERS_ALLOWED>1-9</NUMBERS_ALLOWED>
 </MOBILE_CONTROLLER>
 <PERFORMANCE_STATS>
   <REPORT>
     <!--The framework automatically exclude non-numeric type
statistics-->
     <ACTIONS/>
     <INPUT/>
     <TIMES/>
   </REPORT>
   <SUMMARY>
     <PLAYER_SPEED/>
     <PRECISION_AWARD/>
     <PRACTICE_HARDER_WARNING/>
     <MATH-GENIE_AWARD/>
   </SUMMARY>
 </PERFORMANCE_STATS>
</GAME>
```


The Word-search game Features


```xml
<?xml version="1.0" encoding="utf-8"?>
<GAME xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Features.xsd">
  <NAME>The word-search game</NAME>
  <HORIZONTAL_POSITIONS>8</HORIZONTAL_POSITIONS>
  <VERTICAL_POSITIONS>10</VERTICAL_POSITIONS>
  <PLAYERS>4</PLAYERS>
  <!--<TARGET_ID></TARGET_ID>
  <OBSTACLE_ID></OBSTACLE_ID>-->
  <TYPE>custom:alphabetic|sequenced numeric</TYPE>
  <CENTRAL_MARK_COVERAGE>80</CENTRAL_MARK_COVERAGE>
  <SIDE_MARK_COVERAGE>80</SIDE_MARK_COVERAGE>
  <LEVELS>
    <LEVEL name="Find the most words"/> <!--...within a time limit-->
    <LEVEL name="Find all words"/>
  </LEVELS>

  <VIEW1>
    <GUIs>
      <GUI>Login activity</GUI>
      <GUI>Summary and report</GUI>
    </GUIs>
    <AVATAR/>
    <NAME_LABEL/>
    <SCORE_LABEL>Current Score</SCORE_LABEL>
  </VIEW1>

  <VIEW2>
    <GUIs>
      <GUI>Selected game rules</GUI>
      <GUI>Starting positions</GUI>
      <GUI>Virtual playfield</GUI>
    </GUIs>
    <RULES_LEFT>Rules.png</RULES_LEFT>
    <RULES_RIGHT>controller_instr.png</RULES_RIGHT>

    <COLORS/>
  </VIEW2>

  <REMOTE_MANAGER>
    <MODE_OPTIONS/>
    <LEVEL_OPTIONS/>
    <FX_OPTIONS/>
    <VOLUME/>
  </REMOTE_MANAGER>

  <MOBILE_CONTROLLER>
    <TYPE>custom</TYPE>
  </MOBILE_CONTROLLER>

  <PERFORMANCE_STATS>
    <REPORT>
      <!--The report automatically excludes the arithmetic stats-->
      <ACTIONS/>
      <INPUT>
        <EXCLUDING>Total submitted characters</EXCLUDING>
        <EXCLUDING>Total distinct submitted characters</EXCLUDING>
        <EXCLUDING>Characters usage</EXCLUDING>
      </INPUT>
      <TIMES>
        <EXCLUDING>Avg. input time per character</EXCLUDING>
      </TIMES>
      <EXTRA_STAT>Wrong words</EXTRA_STAT>
      <EXTRA_STAT>Wrong words analysis</EXTRA_STAT>
      <EXTRA_STAT>Spelling mistakes</EXTRA_STAT>
    </REPORT>
    <SUMMARY>
      <EXTRA_MSG>wins the Grammar-Expert award</EXTRA_MSG>
      <EXTRA_MSG>should practice harder</EXTRA_MSG>
    </SUMMARY>
  </PERFORMANCE_STATS>
</GAME>
```

The Scrabble variation Features


```xml
<?xml version="1.0" encoding="utf-8"?>
<GAME xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Features.xsd">
  <NAME>A Scrabble-like application</NAME>
  <HORIZONTAL_POSITIONS>10</HORIZONTAL_POSITIONS>
  <VERTICAL_POSITIONS>10</VERTICAL_POSITIONS>
  <PLAYERS>4</PLAYERS>
  <!--<TARGET_ID></TARGET_ID>-->
  <OBSTACLE_ID></OBSTACLE_ID>
  <TYPE>custom:sequenced numeric|alphabetic</TYPE>
  <CENTRAL_MARK_COVERAGE>10</CENTRAL_MARK_COVERAGE>
  <SIDE_MARK_COVERAGE>100</SIDE_MARK_COVERAGE>

  <LEVELS>
    <LEVEL name="Easy"/>
    <LEVEL name="Medium">
      <OBSTACLES>6</OBSTACLES>
    </LEVEL>
    <LEVEL name="Hard">
      <OBSTACLES>18</OBSTACLES>
    </LEVEL>
  </LEVELS>

  <VIEW1>
    <GUIs>
      <GUI>Login activity</GUI>
      <!--<GUI>Player performance</GUI>-->
      <GUI>Summary and report</GUI>
    </GUIs>
    <AVATAR/>
    <NAME_LABEL/>
  </VIEW1>

  <VIEW2>
    <GUIs>
      <GUI>Selected game rules</GUI>
      <GUI>Starting positions</GUI>
      <GUI>Virtual playfield</GUI>
    </GUIs>
    <RULES_LEFT>ScrabbleRules.png</RULES_LEFT>
    <RULES_RIGHT>controller_instr.png</RULES_RIGHT>
    <COLORS/>
  </VIEW2>

  <REMOTE_MANAGER>
    <MODE_OPTIONS/>
    <LEVEL_OPTIONS/>
    <FX_OPTIONS/>
    <VOLUME/>
  </REMOTE_MANAGER>

  <MOBILE_CONTROLLER>
    <TYPE>custom</TYPE>
  </MOBILE_CONTROLLER>

  <PERFORMANCE_STATS>
    <REPORT>
    <ACTIONS/>
    <INPUT/>
    <TIMES/>
    <EXTRA_STAT>Wrong words</EXTRA_STAT>
    <EXTRA_STAT>Wrong words analysis</EXTRA_STAT>
    <EXTRA_STAT>Spelling mistakes</EXTRA_STAT>
  </REPORT>
  <SUMMARY>
    <EXTRA_MSG>wins the Grammar-Expert award</EXTRA_MSG>
    <EXTRA_MSG>should practice harder</EXTRA_MSG>
  </SUMMARY>
 </PERFORMANCE_STATS>
</GAME>
```


Math Labyrinth's Features


```xml
<?xml version="1.0" encoding="utf-8"?>
<GAME xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Features.xsd">
  <NAME>Math Labyrinth</NAME>
  <HORIZONTAL_POSITIONS>8</HORIZONTAL_POSITIONS>
  <VERTICAL_POSITIONS>10</VERTICAL_POSITIONS>
  <PLAYERS>4</PLAYERS>
  <OBSTACLE_ID>Walls</OBSTACLE_ID>
  <TYPE>numeric</TYPE>
  <CENTRAL_MARK_COVERAGE>80</CENTRAL_MARK_COVERAGE>
  <POSITIONS>
    <PLAYER_POSITIONS> 66|07|05|34 </PLAYER_POSITIONS>
  </POSITIONS>
  <LEVELS>
    <LEVEL name="Easy">
      <OBSTACLES>15</OBSTACLES>
    </LEVEL>
    <LEVEL name="Medium">
      <OBSTACLES>20</OBSTACLES>
    </LEVEL>
    <LEVEL name="Hard">
      <OBSTACLES>30</OBSTACLES>
    </LEVEL>
  </LEVELS>

  <VIEW1>
    <GUIs>
      <GUI>Login activity</GUI>
      <GUI>Player performance</GUI>
      <GUI>Summary and report</GUI>
    </GUIs>
    <AVATAR/>
    <NAME_LABEL/>
<POSITION_LABEL>Position</POSITION_LABEL>
    <COMMAND_LABEL>Command</COMMAND_LABEL>
    <SCORE_LABEL>Current Score</SCORE_LABEL>
    <OBSTACLES/>
  </VIEW1>

  <VIEW2>
    <GUIs>
      <GUI>Selected game rules</GUI>
      <GUI>Starting positions</GUI>
      <GUI>Virtual playfield</GUI>
    </GUIs>
    <RULES_LEFT>Rules.png</RULES_LEFT>
    <RULES_RIGHT>controller_instr.png</RULES_RIGHT>
    <AVATAR/>
  </VIEW2>

  <REMOTE_MANAGER>
    <MODE_OPTIONS/>
    <LEVEL_OPTIONS/>
    <FX_OPTIONS/>
    <VOLUME/>
  </REMOTE_MANAGER>

  <MOBILE_CONTROLLER>
    <TYPE>numeric</TYPE>
    <NUMBERS_ALLOWED>0-9</NUMBERS_ALLOWED>
    <DECIMALS_ALLOWED/>
  </MOBILE_CONTROLLER>

  <PERFORMANCE_STATS>
    <REPORT>
      <!--The report automatically excludes the alphabetic (custom)
stats-->
      <ACTIONS>
        <EXCLUDING>Targets found</EXCLUDING>
      </ACTIONS>
      <INPUT>
        <EXCLUDING>Total distinct commands</EXCLUDING>
      </INPUT>
      <TIMES/>
      <EXTRA_STAT>Total targeting mistakes</EXTRA_STAT> <!--accounts
for targeting on wrong positions-->
    </REPORT>
    <SUMMARY>
      <PLAYER_SPEED/>
      <PRECISION_AWARD/>
      <PRACTICE_HARDER_WARNING/>
      <MATH-GENIE_AWARD/>
      <EXTRA_MSG>MVP award</EXTRA_MSG> <!--awarded for most moves &
minimum mistakes-->
    </SUMMARY>
  </PERFORMANCE_STATS>
</GAME>
```

APPENDIX G – Initial System Functions and Functional Requirements

A. Playfield

User tracking

Functional Requirements

#	Requirement	Description
1	Players and teams recognition	The system must be able to discriminate among teams and players.
2	Recognition speed	Recognizing a player when moving from a square to another must occur instantly.

Technical Characteristics and Constraints

#	Characteristics	Description
Hardware		
1	Resistance to pressure	Any h/w used below the playfield for recognition purposes must be robust enough to endure high levels of pressure such as pressure caused by steps and jumps.
2	Embedded – unobtrusive readers	Any h/w used below the playfield for recognition purposes must be efficiently embodied and invisible. Also stepping on the carpet must feel as stepping on a normal one.

Technical Characteristics

#	Characteristics	Description
1	Size	Position minimum size: 40x40 cm^2 Playfield minimum size: 400x400 cm^2
2	Multi-face surface	Instead of using a different playfield for each application, the carpet should be modifiable to accommodate any activity. Both sides could be shaped.
3	Portability	Special attention must be paid to setting up a foldable, light-weight and easily stored carpet.

Safety Requirements

#	Requirement	Description
1	Stability	Despite its rich content, the carpet must cater for stability, in order to avoid injuries during play.
2	Softness	The carpet's material must cater for a soft activity floor.

B. Graphical world

Visual display

Functional Requirements

#	Requirement	Description
1	Floor illustration	The playfield must be accurately illustrated on the display and clearly conceived from any location.
2	Players and teams illustration	Teams and players must be mentioned on a side log on the board. The game illustration will depict players with distinct colours.
3	Player location illustration	Each player's position must be shown with a distinct colour. *Players may choose their colours by the first square they choose to start from.*
4	Moves validation	The display must envision only the right moves attempted by the players. *Right moves must be shown with colour and frames might need to flash to be obvious from a distance.* Wrong moves can be covered only by sound effects.
5	Player exclusion	The display must notify a player, when the application decides his exclusion according to the rules. This can be also supported by sound.
6	Ending the game	The display must display game-over information, such as the winning time, total time, score per team and/or player etc.

Appearance

Functional Requirements

#	Requirement	Description
1	Divided display	The board will be divided in two (80-20) to fit any game visualization on the big side and any team- or task-specific information on the small side.
2	Intuitive displays	Graphics used must be simple and colourful. They must also depict the game floor precisely.
3	Adjustable projection size	The screen content must be obvious from anywhere inside the room. Might need to cut the display in more pieces to zoom deeper where necessary.
4	Reports	After each round or game, the system's output must include logs / reports for all players, as well as keep high scores, statistics etc.

Interface Requirements

#	Requirement	Description
1	Varied content according to class level	There might be a need for content and graphics versioning depending on the target ages.

2	Look & feel appealing to children	Joyful graphics, colours, music and playful atmosphere.
3	Personalization capability	The players should choose their colours, the music or the theme of the game.

C. Sound

Functional Requirements

#	Requirement	Description
1	Sound effects	Sound effects must be used as feedback for each player's moves.
2	Music	Music can also be used optionally depending on the application

D. Learning applications

Controlling an application

Functional Requirements

#	Requirement	Description
1	Controls from a distance	Main menu and any properties could be controlled from the game floor. For example, the user's options, visualized on screen, could be controlled by the players positioning on the respective playfield positions.
2	Traditional controls	Additionally, the applications may be controlled with keyboard and mouse.
3	Automatic controls - Adaptation	The system must contain adaptation logic, similar to "understanding" what the players wish to play and launching the right application.

Input

Functional Requirements

#	Requirement	Description
1	Gesture input	Player moves will be perceived by the applications through gestures. Gestures like "multiply", "divide", "add", "subtract".
2	Mobile phone input	Mobile phones can be used as an alternative input modality.
3	Headset and voice recognition	A microphone enabled headset, capable of directing personal voice messages to voice recognition software, seems to be the ideal input solution.

Content editor

Functional Requirements

#	Requirement	Description
1	Add Q & A	Teachers must be able to add Questions and Answers, according to the needs of each game (learning application). Answers have to be declared as Right or Wrong
2	Modify or Delete an insertion	Changes must be possible.
3	Assign category to a project	Category responds to teaching field or discipline.
4	Assign game type	Game type responds to the type of application that the content is made for.
5	Assign level	Level denotes the content's difficulty or corresponding knowledge level.
6	Save a project	Save possibilities and extraction to a readable format will be required.
7	Simplicity	The editor should not provide more controls than those above and should be as simple as possible. It should require the minimum effort to be read (even by aged teachers).