



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

**Development and Evaluation of a Serious Game based on  
Interactive Surfaces for Children Instruction**

Desarrollo y Evaluación de un Juego Serio basado en Superficies Interactivas en el  
Ámbito de la Educación Infantil

**MASTER'S THESIS**  
MASTER'S DEGREE IN SOFTWARE SYSTEMS  
ENGINEERING AND TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS AND  
COMPUTATION

**SANDRA EL JURDI**  
**SEPTEMBER 2016**

**Supervisor:**

Dr. Francisco Javier Jaén Martínez

---

# Abstract

Collaborative Problem Solving is an essential soft skill that should be fostered from a young age. Many learning institutions now include the enhancement of this skill in their curricula in order to produce more capable engineers and scientists.

In this work, a multi-platform tool for CPS enhancement called Quizbot is introduced and its development on three different platforms is detailed. Quizbot itself was developed with the intention of enhancing several of the CPS sub-skills, such as communication, negotiation, organization and planning. The different platforms offer varying advantages and disadvantages to the process of CPS enhancement.

An experiment was performed with summer school students to compare the three platforms and observe the type of behavior and interaction that Quizbot encourages. The results of this experiment showed a generally positive reaction from the children and a promising usability for Quizbot, making it interesting to improve and test the tool further.

**Keywords:** Problem-based learning, Collaborative Problem Solving, Multi-touch display, Tangible User Interface, Physical Spaces

---

# Resumen

La resolución de problemas colaborativa es una de las habilidades que se deben fomentar desde una edad temprana. Muchas instituciones ahora incluye la mejora de esta habilidad en sus planes de estudio para producir ingenieros y científicos más capaces.

En este trabajo, se introduce una herramienta multi-plataforma llamada Quizbot para el fomento de la resolución de problemas colaborativa, y se detalla su desarrollo en tres plataformas diferentes. Quizbot se desarrolló con la intención de fomentar varias de las sub-habilidades que forman parte de la resolución de problemas colaborativa, como la comunicación, negociación, organización, y planificación. Las diferentes plataformas ofrecen varias ventajas y desventajas al proceso de fomento de la resolución de problemas colaborativa.

Se realizó un experimento con estudiantes de escuela de verano para comparar las tres plataformas y observar el tipo de comportamiento e interacción que Quizbot incentiva. Los resultados de este experimento muestran una reacción generalmente positiva de los niños y una usabilidad prometedora para Quizbot, dando interés a mejorar y probar la herramienta aún más.

**Palabras clave:** Aprendizaje basado en problemas, Resolución de problemas colaborativa, Pantalla multi-táctil, Interfaz de usuario tangible, Espacios Físicos

# Table of Contents

<b>Abstract</b>	<b>i</b>
<b>Resumen</b>	<b>i</b>
<b>Table of Contents</b>	<b>iii</b>
<b>List of Tables</b>	<b>iv</b>
<b>List of Figures</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Collaborative Problem Solving . . . . .	2
1.3 Research Goal . . . . .	3
<b>2 Related Works</b>	<b>7</b>
<b>3 Quizbot: A Robot Board-based Gamification Approach to Support CPS</b>	<b>12</b>
3.1 Quizbot Gamification Dynamics . . . . .	12
3.2 Quizbot for Tabletops and Handheld Tablets . . . . .	15
3.3 Quizbot for Physical Spaces . . . . .	29
3.4 Quizbot Server . . . . .	34
<b>4 Evaluation</b>	<b>41</b>
4.1 Qualitative Results . . . . .	42
4.2 Observational Results . . . . .	53
4.3 Discussion . . . . .	54
<b>5 Conclusions and Future Work</b>	<b>57</b>
5.1 Conclusions . . . . .	57
5.2 Future Work . . . . .	58
5.3 Publications . . . . .	58

---

5.4 Acknowledgements . . . . .	58
<b>Bibliography</b>	<b>58</b>

# List of Tables

3.1	Table of Quizbot Functional Requirements. . . . .	14
3.2	Description of how the server's Styles module works. . . . .	39
4.1	Post-game session questionnaire . . . . .	42
4.2	Results for questionnaire Fun Sorter. Mean score for each platform is shown between parenthesis. . . . .	43

# List of Figures

1.1	Matrix of Collaborative Problem Solving skills for PISA 2015 . . . . .	4
1.2	Overview of factors and processes for Collaborative Problem Solving in PISA 2015 . . . . .	5
1.3	Indirect Collaboration . . . . .	6
1.4	Direct Collaboration by Delegation . . . . .	6
1.5	Direct Collaboration by Team Formation . . . . .	6
2.1	Left: tangible interface. Right: multitouch interface. . . . .	7
2.2	Schematic representation of CPSS. . . . .	8
2.3	Conceptual framework for building and supporting shared understanding. . . . .	9
2.4	Process in the CPSS . . . . .	9
2.5	Collaborative sorting group (x-axis: the number of pairs, y-axis: time in seconds). . . . .	10
2.6	Individual sorting group (x-axis: participant, y-axis: time in seconds). . . . .	10
2.7	Means and standard deviations of the sorting task. . . . .	10
2.8	The design of the multi-touch interactive jigsaw game. . . . .	11
2.9	The three jigsaw game scaffold types. . . . .	11
2.10	Results of the Problem-Solving Activity. . . . .	11
3.1	Quizbot Conceptual Model. . . . .	14
3.2	Instance of Quizbot running on a Windows Tabletop. . . . .	15
3.3	Four connected instances of Quizbot running on Android Tablets. . . . .	16
3.4	Partial Tiled interface with game board and layers. . . . .	16
3.5	Quizbot for Tabletops and Handheld Tablets in Normal Mode. . . . .	17
3.6	Quizbot for Tabletops and Handheld Tablets in Quiz Mode. . . . .	17
3.7	Quizbot for Tabletops and Handheld Tablets Class Diagram. . . . .	19
3.8	Quizbot for Tabletops and Handheld Tablets Initial Screen. . . . .	20
3.9	Quizbot for Tabletops and Handheld Tablets Loading Screen. . . . .	21
3.10	Quizbot game screen showing Go Forward at south position, Turn Right at east, Turn Left at west and Stop at north. . . . .	22

---

3.11 Quizbot game screen showing a subset of movement commands. . . . .	22
3.12 Quizbot for Tabletops and Handheld Tablets Selection Screen. . . . .	23
3.13 Quizbot game screen displaying game menu. . . . .	24
3.14 Quizbot game screen displaying wrong answer animation. . . . .	25
3.15 Quizbot game screen displaying bomb explosion animation. . . . .	25
3.16 Quizbot game screen displaying a Dialog with a quiz question. . . . .	27
3.17 Quizbot game screen displaying a Dialog with a correct answer and its informational text. . . . .	27
3.18 RFID tags placed around a game item under the foam mattress. . . . .	30
3.19 Tangible Lego Mindstorms robot setup. . . . .	30
3.20 Quizbot in a Physical Space. . . . .	31
3.21 Quizbot for Physical Spaces Class Diagram: RFID Registration. . . . .	32
3.22 Quizbot for Physical Spaces Class Diagram: Commands and Items Reader. . . . .	32
3.23 Quizbot for Physical Spaces Class Diagram: Key Item Cells. . . . .	33
3.24 Quizbot server communication with digital platforms. . . . .	35
3.25 Quizbot server communication with TUI platform. . . . .	36
4.1 Results for questionnaire Again-Again table, stating desire to play again in class. . . . .	44
4.2 Results for questionnaire Again-Again table, stating desire to play again outside class. . . . .	45
4.3 Results for which school subjects are preferred for questions on the Tabletop platform. . . . .	46
4.4 Results for which school subjects are preferred for questions on the Tablets platform. . . . .	47
4.5 Results for which school subjects are preferred for questions on the TUI platform. . . . .	48
4.6 Company preference for Tabletop platform: 10% alone, 90% with friends. . . . .	49
4.7 Company preference for Tablets platform. 16.25% alone, 83.75% with friends. . . . .	50
4.8 Company preference for TUI platform: 7.5% alone, 92.5% with friends. . . . .	51
4.9 Changes suggested for Quizbot. . . . .	52



# Introduction

## 1.1 Motivation

Problem Solving is one of many soft skills that a person may possess. Nowadays it is not enough for a student to simply have high grades, especially in engineering domains or in applied sciences; they are now also encouraged to have several other skills besides their technical knowledge, such as communication skills, teamwork, adaptability and problem solving, among others. Collaborative Problem Solving (CPS) emerges as a combination of several of these skills and is also highly valued and sought after. Vygotsky's Social Development Theory (Vygotsky, 1978) implies that a person's potential can only be achieved through interaction with and support from other, ideally more capable, people and various tools. This is based on the idea that when trying to solve a problem, the exchange of ideas could lead to a shared understanding that an individual cannot achieve alone. (Gokhale, 1995) also highlights the fact that "the active exchange of ideas within small groups not only increases interest among the participants but also promotes critical thinking." This leads to the conclusion that focusing on developing a person's problem solving skills on an individual level is not enough and it is now essential for a person to have a certain level of proficiency with collaborative problem solving.

There are many methods that can be used to nurture and enhance any given skill, including adding an element of play to the learning process, which has been proven to be a natural and successful way of improving the effectiveness of learning seeing as human culture is generated at least partially through play (Huizinga, 2007). With the aid of technology, educational games (also known as serious games) can be created to help develop skills like CPS through play and offer instant feedback and interactivity in a game-based learning environment. Educational games are designed to teach people about certain subjects, expand concepts, reinforce development, or help them learn or improve a skill (such as CPS in this case) (Fallis, 2013) and they have been shown to have many cognitive, motivational, emotional and social benefits (Granic et al., 2014; Wouters et al., 2013).

The traditional desktop computer is a known and reliable medium that is very often used for educational games (Liao and Shen, 2012; Brayshaw and Gordon, 2016; Hatzi-

lygeroudis et al., 2012; Raman et al., 2014; Siang and Radha Krishna Rao, 2003). However, studies show that children of a younger age have a hard time using a mouse and keyboard (Donker and Reitsma, 2007) and prefer newer multi-touch technologies (Romeo et al., 2003) that can be used for the same purpose as well and offer additional benefits over the traditional PC besides the basic technological assistance. Many studies showcase the benefits of using digital tabletops in education. These benefits include fostering creativity (Catala et al., 2011a), learning, knowledge transfer (Schubert et al., 2012), and, most importantly in this case, collaboration (Reski et al., 2014; Hornecker et al., 2008). Unfortunately, tabletops are a rare commodity in real educational settings, mostly due to their elevated cost, as well as because of their form factor which prevents their usage in scenarios that require mobility. Other limiting factors associated with tabletops include the fact that the workspace is always a public one making it difficult to perform any kind of private task, as well as the fact that the actual workspace dimensions are very limited and can only accommodate a certain number of participants (Garcia-Sanjuan et al., 2016b).

One way of dealing with the disadvantages of tabletops while maintaining the positive aspects of those, such as awareness (Hornecker et al., 2008; Gutwin and Greenberg, 2002), parallelism (Rick et al., 2011), and fluidity of the interaction (Hornecker et al., 2008), is to use handheld tablets instead. Handheld tablets easily solve the public vs. private space issue by having a different tablet assigned to each person. Mobility is also increased with these devices due to their small size and light weight, and the workspace dimensions become virtually unlimited if the application is so designed. Furthermore, handhelds are now very common and can be found in any regular household due to their low cost, making it possible to follow a “Bring Your Own Device” (Ballagas et al., 2004) strategy if necessary.

When dealing with younger children (such as primary school students or even kids in kindergarten) Tangible User Interfaces (TUIs) might be an even more interesting platform than purely digital ones like tabletops and tablets. Works like (Strawhacker et al., 2015) suggest that TUIs have an added value in early childhood education “as they resonate with traditional learning manipulatives.” Robots are a clear example of TUI and their usage in education has been steadily increasing. This is due to the fact that robots capture the imagination of children and therefore using them “to support teaching and learning [...] has become a popular research topic” (Li et al., 2009). Studies such as (Schneider et al., 2011) have showcased the advantages of TUIs and (Antle and N., 2013) even made a direct comparison between the traditional mouse-based setup and tangible interfaces where it was observed that the latter enable more exploratory actions, which in turn provide faster and easier ways of interaction.

## 1.2 Collaborative Problem Solving

Problem-solving tasks can be categorized by one or several of the following properties: large, complex, spatially distributed, need extensive communication and a large degree of functional specialization between the agents (Obeid and Moubaidin, 2009). If a problem satisfies one or more of these properties it is considered to be unsolvable by a single agent and therefore the collaboration of several agents is required. The PISA 2015 (Oecd, 2015) definition of Collaborative problem solving competency is “the capacity of an individual to effectively engage in a process whereby two or more agents attempt to solve a problem

---

by sharing the understanding and effort required to come to a solution and pooling their knowledge, skills and efforts to reach that solution.” From this definition we can extract three core competencies, which are:

1. Establishing and maintaining shared understanding
2. Taking appropriate action to solve the problem
3. Establishing and maintaining team organization

Furthermore, the PISA 2012 problem solving framework identified four cognitive processes in individual problem solving: Exploring and understanding, representing and formulating, planning and executing, monitoring and reflecting (Oecd, 2012).

1. Exploring and understanding implies understanding the situation by deciphering the initial information provided about the problem and any further information that is exposed during the exploration of and interaction with the problem.
2. During the representing and formulating process, the information gathered previously is selected, organized, and integrated with previous knowledge. This is achieved by representing the information in the most convenient way, whether using graphs, tables, symbols, or words, and then formulating hypotheses by extracting the relevant factors and evaluating the information.
3. Planning and executing includes clarifying the goal of the problem, setting sub-goals, and developing a plan to reach the main goal. The plan that is created in the first half of this process is then executed in the second part.
4. Finally, monitoring and reflecting implies monitoring the steps in the plan to reach the main goal and reflecting on any possible solutions and assumptions.

Figure 1.1 shows a matrix of skills formed when the three major CPS competencies are crossed with the four individual PS processes. An overview of the factors and processes for Collaborative Problem Solving (Oecd, 2015) can be seen in figure 1.2.

(Fan, 2010) identifies four major roles that are usually formed during a collaborative problem solving activity. These roles are called Brainstormer, Critic, Supporter, and Team Wrangler. However, these roles are taken on by the actors in different capacities depending on the type of activity. (Conen and Neumann, 1996) presents three different types of possible collaboration (Indirect Collaboration, Direct Collaboration by Delegation and Direct Collaboration by Team Formation) which can be seen in figures 1.3, 1.4 and 1.5.

### **1.3 Research Goal**

The goal of this thesis is to develop and evaluate a tool for enhancing collaborative problem solving skills in young students. More specifically, the research focuses on the following sub-skills that are associated with CPS (Oecd, 2015):

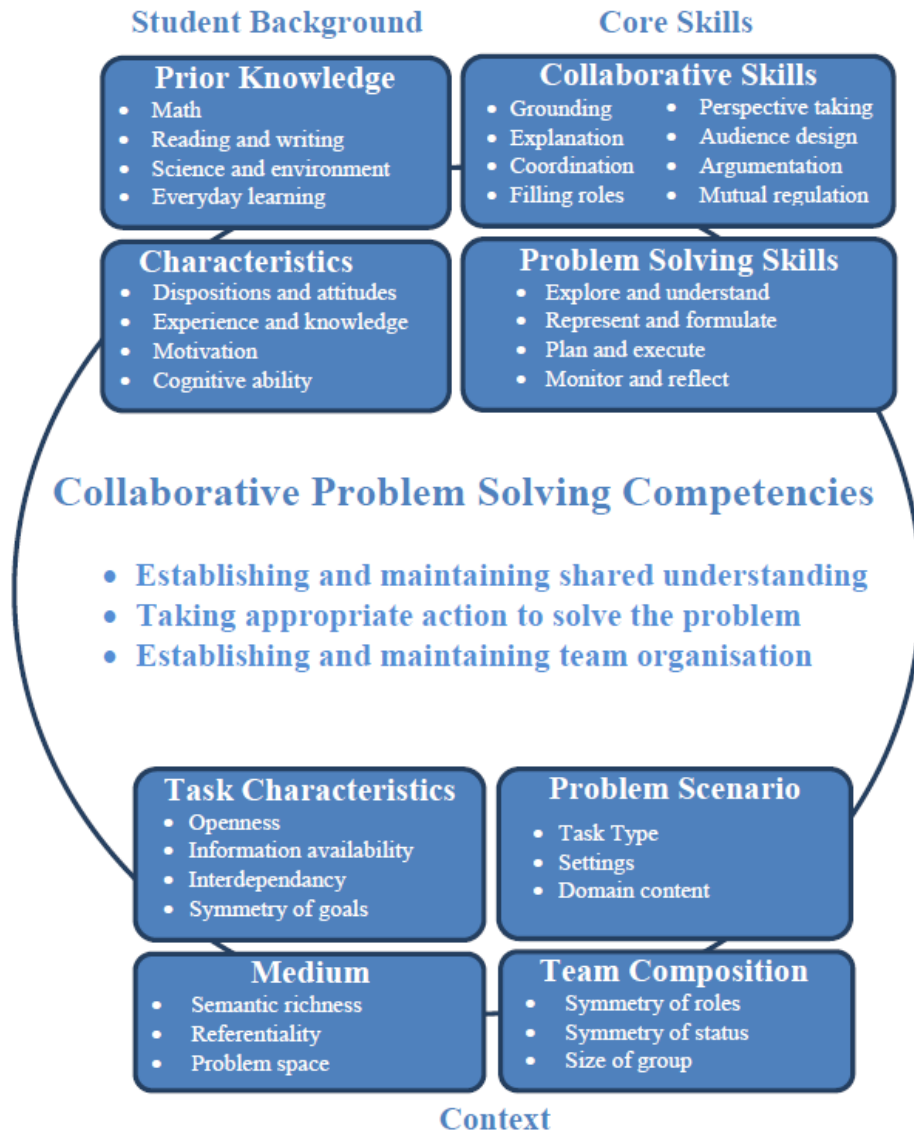
---

	(1) Establishing and maintaining shared understanding	(2) Taking appropriate action to solve the problem	(3) Establishing and maintaining team organisation
(A) Exploring and Understanding	(A1) Discovering perspectives and abilities of team members	(A2) Discovering the type of collaborative interaction to solve the problem, along with goals	(A3) Understanding roles to solve problem
(B) Representing and Formulating	(B1) Building a shared representation and negotiating the meaning of the problem (common ground)	(B2) Identifying and describing tasks to be completed	(B3) Describe roles and team organisation (communication protocol/rules of engagement)
(C) Planning and Executing	(C1) Communicating with team members about the actions to be/ being performed	(C2) Enacting plans	(C3) Following rules of engagement, (e.g., prompting other team members to perform their tasks.)
(D) Monitoring and Reflecting	(D1) Monitoring and repairing the shared understanding	(D2) Monitoring results of actions and evaluating success in solving the problem	(D3) Monitoring, providing feedback and adapting the team organisation and roles

**Figure 1.1:** Matrix of Collaborative Problem Solving skills for PISA 2015

- **Negotiation:** wherein the agents involved in the CPS task are expected to share their knowledge, express their ideas and come to a shared understanding leading to an agreement over the solution of the problem or the course of action to take in order to reach a solution. In some cases, an actor is expected to learn to become more flexible in their negotiations, while in other cases an actor may need to learn to be more assertive.
- **Planning:** this includes the ability to divide a given problem into smaller tasks and formulating as efficient a plan as possible in order to reach the final solution.
- **Communication:** this is the skill that makes the enhancement of the other skills possible. Negotiation, planning and organization can only be achieved through communication; therefore it is essential to develop the right type of communication in order to ensure the correct transmission of information and the effective interaction between the actors.
- **Organization:** wherein the agents are expected to take on the necessary roles in the team in order to structure and coordinate their efforts and therefore reach a solution in the least chaotic way possible.

The CPS enhancement tool must take on the form of a game in order to make the process more enjoyable and effective for the end user. Furthermore, the game is developed for more than one platform (namely, a tabletop, a multi-tablet environment and a TUI) in order to make it possible to not only evaluate the effectiveness of the tool itself but to also compare the possible additional benefits that different platforms may offer.



**Figure 1.2:** Overview of factors and processes for Collaborative Problem Solving in PISA 2015

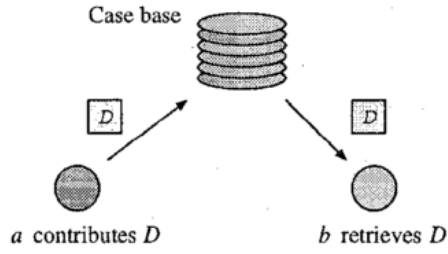


Figure 1.3: Indirect Collaboration

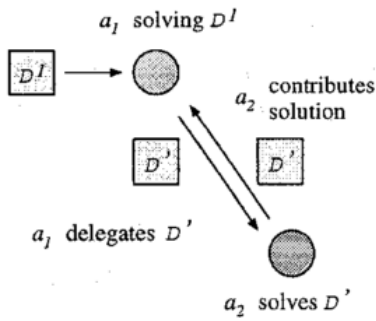


Figure 1.4: Direct Collaboration by Delegation

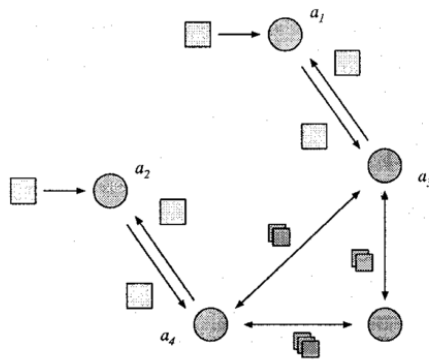


Figure 1.5: Direct Collaboration by Team Formation

## Related Works

Problem solving skills are highly valued and therefore there are many studies about nurturing and enhancing these skills. Works such as (Idrus et al., 2010) address how certain learning institutes have been adopting a Problem-Based Learning approach. PBL was developed in the 1970s and has been adopted by many disciplines (Savery, 2006) but it is now important to add a collaborative aspect to this skill as well. (Pinho-Lopes and MacEdo, 2013) showcases another university where a Problem-Based Learning approach was adopted, but with the addition of collaborative learning as well. In both cases where PBL was added to a real learning environment, it was concluded that not only did the students and lecturers find it important to work on enhancing these soft skills but also that this kind of active learning helped in “promoting and facilitating the construction of knowledge and in developing competencies by the students”.

Considering the amount of studies demonstrating the usefulness of game-based learning it is only natural for there to be works that aim to use videogames to help develop problem solving skills. (Hooshyar et al., 2015; Brophy and P., 1995; Sancho et al., 2008; Shang et al., 2006) are examples of games that were developed for the purpose of improving students’ CPS skills. While these games do serve their purpose, they are based on a technology that is now slowly but surely becoming obsolete. Instead of desktop applications, it is more interesting to use mobile and/or tangible interfaces nowadays. Works such as (Chang et al., 2009) highlight the need for new technologies and active methods to be introduced into learning by showcasing a framework that presents the notion of a ubiquitous game-based learning model.

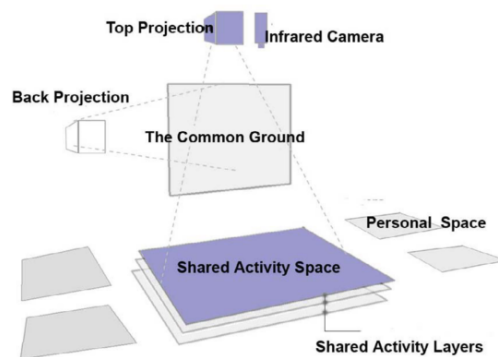


**Figure 2.1:** Left: tangible interface. Right: multitouch interface.

(Schneider et al., 2011) showcases the benefits of a TUI over a tabletop using a learning environment. The two different setups can be seen in figure 2.1. The work studies the impact of tangible interfaces and concludes that it helps promote constructive behavior such as exploration and collaboration, which are included in the CPS competencies, suggesting that TUIs may indeed be useful for CPS enhancement. On the other hand, (Catala et al., 2011b) is another work comparing tabletops to TUIs in order to explore creative learning and instead concludes that, while users are more productive using a tangible platform, they instead manage to design more elaborate and complex solutions using the tabletop. Unfortunately, neither works are aimed at studying either platform's benefits for fomenting CPS.

(Scharf et al., 2012; Schneider et al., 2012) are examples of tangible educational collaborative games. The former fosters algorithmic construction and reasoning as well as discussion in general, while the latter aims to help students explore, solve and understand probability problems in particular. Both works pretend to foster skills that are related to CPS, but neither focuses on the area in particular.

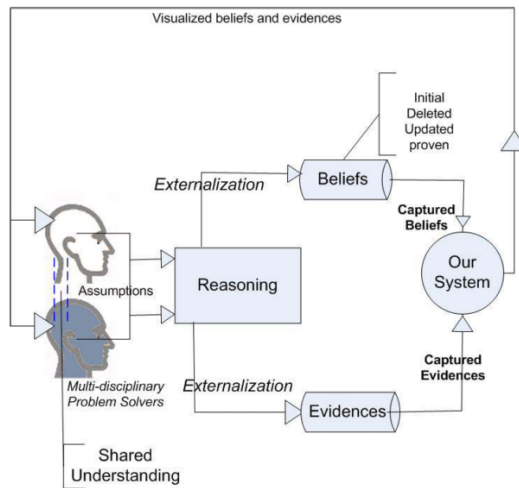
(Deshpande et al., 2005) is an example of an original setup using projectors and an infrared camera to create a Collaborative Problem Solving Space (CPSS). This space includes a shared activity space and a personal space as well as a common ground space, seen in figure 2.2, which solves any private vs. public space issues. However, this work focuses first and foremost on shared understanding, as seen in figures 2.3 and 2.4, and therefore ignores some of the other CPS competencies that are defined by PISA (Oecd, 2015).



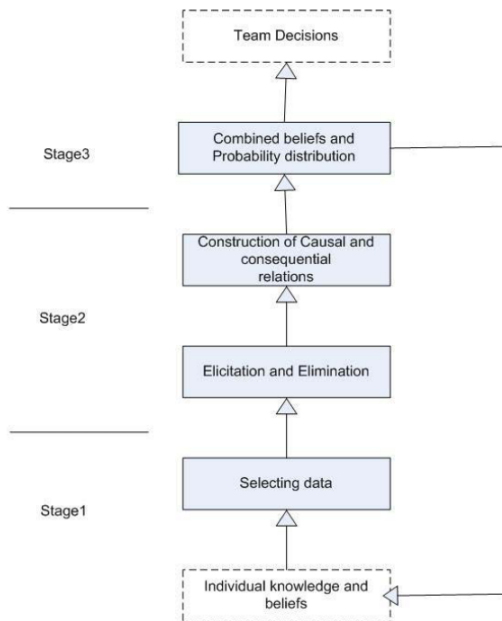
**Figure 2.2:** Schematic representation of CPSS.

Another example of game-based learning is (Shatte and Holdsworth, 2014), where a sorting application using Augmented Reality (AR) is tested individually and collaboratively. The study clearly indicates that tasks which are performed collaboratively are executed more efficiently due to the sharing of sub-tasks and communication, figure 2.5. However, the study also concludes that the AR technology itself provides no immediate benefit seeing as there was no significant improvement in the individual tasks, figure 2.6. A summary of the study's results can be found in figure 2.7.



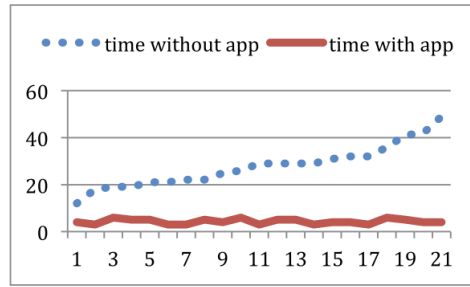


**Figure 2.3:** Conceptual framework for building and supporting shared understanding.

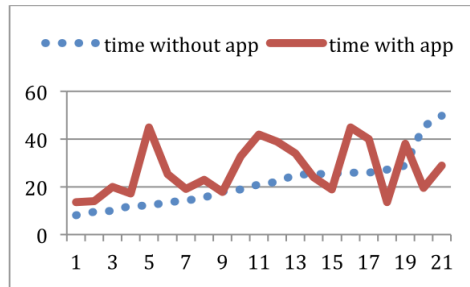


**Figure 2.4:** Process in the CPSS

(Hung et al., 2012; Cheng-Yu Hung et al., 2014) present a collaborative educational game using multi-touch technology. The works present an educational jigsaw puzzle game that can be played on a Microsoft Surface. The game’s interface can be seen in figures 2.8



**Figure 2.5:** Collaborative sorting group (x-axis: the number of pairs, y-axis: time in seconds).



**Figure 2.6:** Individual sorting group (x-axis: participant, y-axis: time in seconds).

Group	Task	Mean (sec)	StDev (sec)
Individual sorting	without app	21.62	10.84
	with app	27.18	10.87
Collaborative sorting	without app	27.81	4.29
	with app	8.99	1.06

**Figure 2.7:** Means and standard deviations of the sorting task.

and 2.9. After performing a pre-game test and a post-game test on 20 participants in (Hung et al., 2012) and 240 participants in (Cheng-Yu Hung et al., 2014), the study concludes that the game did indeed help in raising mean score on the tests, figure 2.10. While these works truly showcase the benefits of serious games using multi-touch technology for CPS enhancement, they do not try to test the developed tool using different technologies.

Overall, there is a clear lack of works studying the benefits of games using TUIs to enhance CPS. There is also a lack of studies comparing different platforms' effectiveness for CPS enhancement.



Figure 2.8: The design of the multi-touch interactive jigsaw game.

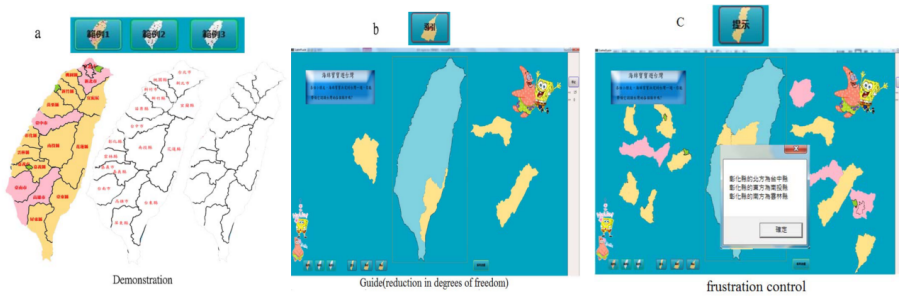


Figure 2.9: The three jigsaw game scaffold types.

Group	(1)		(2)		(3)		(4)		ANOVA	Post hoc
	Without Scaffold		A1~A3		B		C			
	N	M(SD)	N	M(SD)	N	M(SD)	N	M(SD)	F(1,221)	
Level-1	16	15(0.00)	18	15(0.00)	20	15(0.00)	20	15(0.00)	8.180**	(2)>(3)>(4)>(1)
Level-2	16	4.38(2.55)	20	12.05(3.03)	20	6.80(2.04)	18	7.00(2.85)		
Level-3	18	3.89(2.91)	18	7.22(4.11)	20	6.05(2.40)	19	4.32(2.06)		

\*\* $p < 0.01$ .

Figure 2.10: Results of the Problem-Solving Activity.

# Quizbot: A Robot Board-based Gamification Approach to Support CPS

For the purpose of this thesis, a game called Quizbot was developed for multi-touch tabletops, handheld tablets and a Tangible User Interface. Quizbot is, in essence, a board game with a robot as the main actor that the players can move. The board itself has an undetermined number of cells with “items” placed on some of those cells. One or more players can then utilize a set of movement commands in order to lead the robot to a certain item-containing cell. The board section of the game (which includes the board itself with all its cells, the robot as a movable actor and the different movement commands that control the movement of the robot) was developed with the intention of being reusable for other types for board games where the events that are triggered once an item-containing cell is reached may vary depending on the type of game that is being played.

Quizbot means to foster CPS by compelling the users to collaborate in order to solve the given problems. By splitting the movement commands among the users, they are then driven to cooperate and coordinate their efforts in order to plan and execute the robot’s track on the board.

## 3.1 Quizbot Gamification Dynamics

Quizbot is, at its core, a hybrid between a board game and a quiz game. Essentially, when the game is booted the player is presented with a board split up into an undetermined number of cells. At this point the game is considered to be in its normal mode (versus the quiz mode described further below). The board cells may be empty, or they may contain one of the following items:

- Key: this is the most relevant item in the game. Keys are used to activate the game’s

quiz mode, which presents the player with a question that must then be answered.

- **Block:** this is a mostly harmless item that simply serves as a blockade. These cells cannot be passed through by the robot that the player controls on the board.
- **Bomb:** this could be considered the game's main antagonist. Colliding with a bomb while the game is in quiz mode undoes any previously correct answers and the quiz is restarted from the beginning.

The board itself also contains a robot which acts as the player's agent. Four movement commands are associated to the robot and a player may have any number of these commands available to them. The commands are: go forward, turn left, turn right and stop. The reason behind this setup is so that in a multiplayer case, different players would control different commands and must coordinate between each other in order to move the robot efficiently. In the normal mode, the goal is to move the robot to a key-containing cell while avoiding blocks and bombs in order to activate the next quiz. Once that is done, the game enters into quiz mode.

In quiz mode, the blocks and bombs remain in place but all the key-containing cells minus the one that the robot reached in order to activate the quiz are turned into answer cells. The reached cell is turned into a question cell instead. A question cell, as the name suggests, contains a question that the player(s) must answer. A question, or quiz, is answered by guiding the robot to the correct answer cells. The game contains three types of questions that must be answered in different ways.

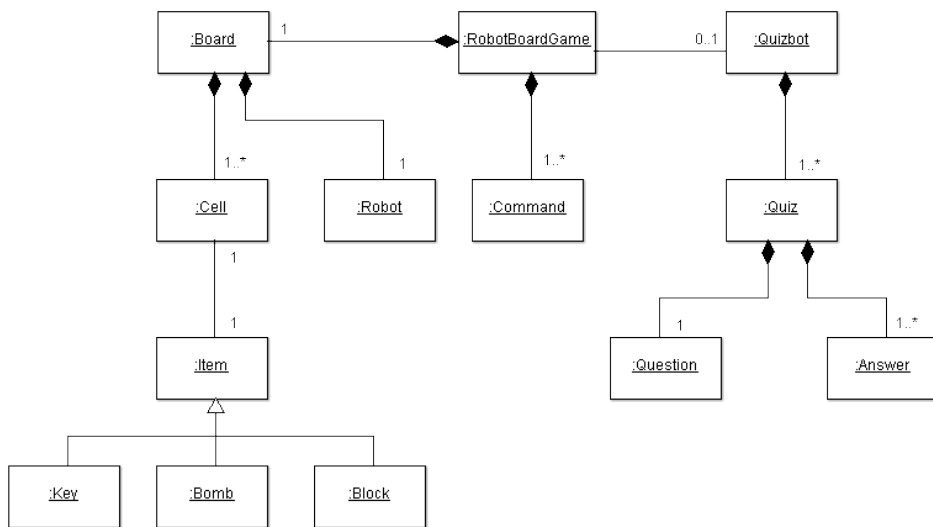
1. **Choice questions:** this type of question is basically a multiple choice type of question where the player is presented with several answers and must choose the correct one(s) out of those, visiting the cells containing these answers in any order of the player's choice.
2. **Ordering questions:** in this type of question, all the answers are correct but the cells containing them must be visited in a specific order dictated by the question itself.
3. **Accumulation questions:** these questions provide the player with a greater freedom of choice as far as answering is concerned. The player simply has to choose any number of answers wherein their sum equals the value given in the question.

Once a question is answered correctly, the quiz is considered ended and in the case of there being more questions available the game goes back to its normal mode with the previous keys (or answer cells), bombs and blocks being removed from the board and replaced with new ones scattered over different cells. If there are no further questions available, the game is considered to be completed. Table 3.1 shows the functional requirements associated with the game. Figure 3.1 shows the Quizbot conceptual model.

---

Requirement	Description
Choose movement commands	The user should be able to choose a subset of commands from the available list of movement commands.
Move robot	The player should be able to use their set of movement commands to move the robot on the board game.
Initiate quiz	The player should be able to initiate a quiz question by reaching a key item on the game board. The question can be a choice, style, ordering style, or accumulation style question.
Answer quiz question	During quiz mode, the player should be able to answer the initiated quiz by moving the robot to the answer cells.
View quiz element	During quiz mode, the player should be able to view the quiz question and the possible answers on the board.
Collide with bomb	The player should be able to cause the robot to collide with a bomb item, resetting the current quiz if it has been previously initiated.
Collide with block	The player should be able to cause the robot to collide with a block item, stopping its movement or changing its course.

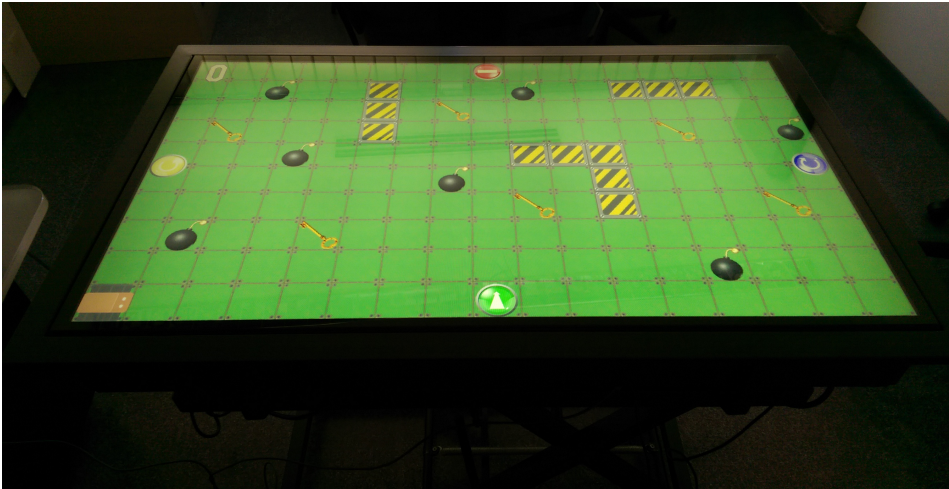
**Table 3.1:** Table of Quizbot Functional Requirements.



**Figure 3.1:** Quizbot Conceptual Model.

## 3.2 Quizbot for Tabletops and Handheld Tablets

Quizbot was developed based on a client-server architecture, where the tabletop or the handhelds would act as clients, making it possible to have the same game view on more than one device at a time. This way, each user could have their own private space while still seeing the game board with the results of the actions taken by everyone playing. While this is not particularly interesting for the tabletop platform, shown in figure 3.2, it is so for the handheld tablets, shown in figure 3.3.

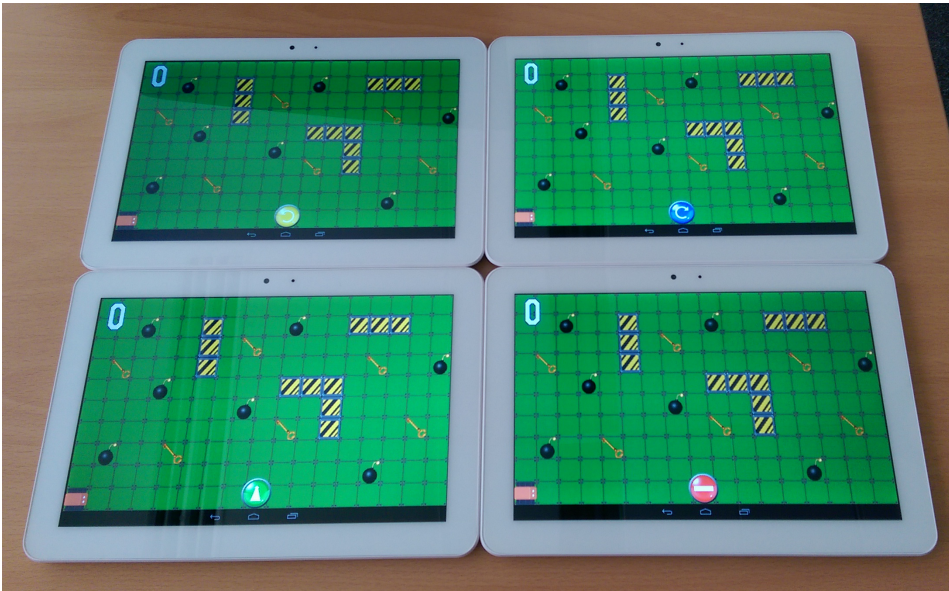


**Figure 3.2:** Instance of Quizbot running on a Windows Tabletop.

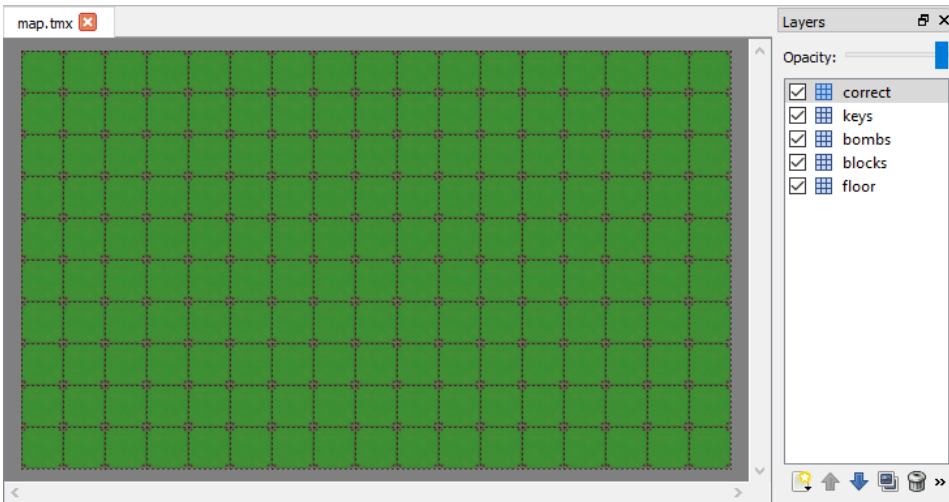
In order to create the game board on these two platforms, the Tiled map editing tool was used. Tiled is a 2D level editor that helps develop game content. “Its primary feature is to edit tile maps of various forms, but it also supports free image placement as well as powerful ways to annotate your level with extra information used by the game”<sup>1</sup>. This information includes things such as objects and collision layers. Figure 3.4 shows an initial empty board with the different map layers. The different items such as keys and bombs are added dynamically later on by the application itself, after their positions are read from a configuration file. This way, each quiz can be configured differently. Figure 3.5 shows what the game board looks like during normal mode and figure 3.6 shows the game board during quiz mode.

LibGDX was used to develop the client side of the game for the tabletop and handheld tablets platforms. LibGDX is a Java game development framework that provides a unified API that works across a variety of supported platforms, including Windows, Linux, Android, iOS, among others. This means that the same code can be used to deploy an application on any of the supported platform. In this case, the platforms in question were Windows for the tabletops and Android for the handheld tablets. For the multi-touch implementation of the game, the following LibGDX classes and libraries were used:

<sup>1</sup><http://doc.mapeditor.org/manual/introduction/>



**Figure 3.3:** Four connected instances of Quizbot running on Android Tablets.



**Figure 3.4:** Partial Tiled interface with game board and layers.

- Game class: an abstract class that implements the `ApplicationListener` interface and delegates to a `Screen`. This allows an application to easily have multiple screens.
- `ApplicationListener` interface: this is called when the `Application` is created, resumed, being rendered, paused or destroyed.



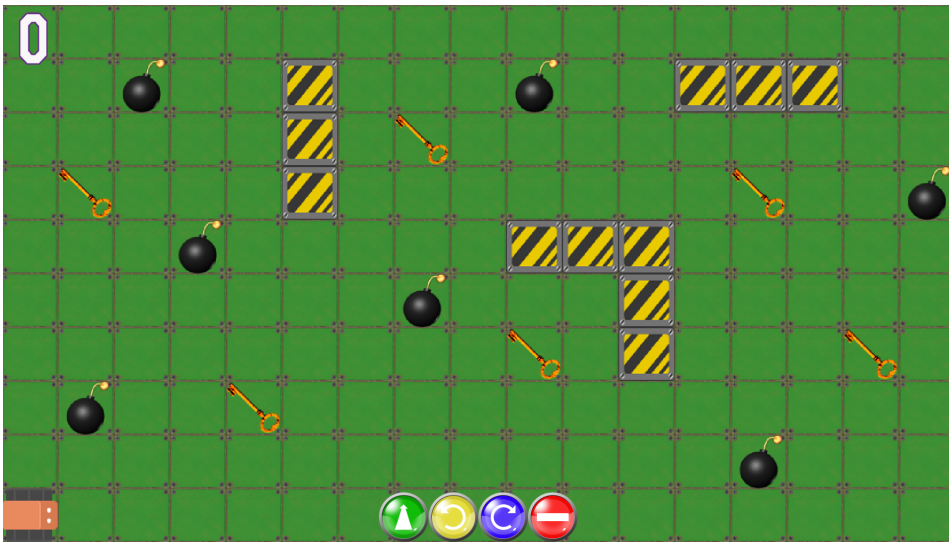


Figure 3.5: Quizbot for Tabletops and Handheld Tablets in Normal Mode.



Figure 3.6: Quizbot for Tabletops and Handheld Tablets in Quiz Mode.

- Screen interface: represents one of many application screens, such as a main menu, a settings menu, the game screen and so on.
- Stage class: extends InputAdapter and implements Disposable. This is a 2D scene graph containing hierarchies of actors (Actor class). It contains a Viewport that controls the coordinates used within the stage and sets up and camera used to convert

between stage coordinates and screen coordinates. A stage receives input events so it can distribute them to the actors.

- Viewport class: an abstract class that manages a Camera and determines how world coordinates are mapped to and from the screen.
  - Camera class: an abstract class used as the base class for OrthographicCamera and PerspectiveCamera. For this implementation, on the former was used.
  - OrthographicCamera class: extends the Camera class to create a camera with orthographic projection.
  - InputAdapter class: implements the InputProcessor interface and functions as its adaptor class from which methods can be derived and overridden.
  - InputProcessor interface: this is used to receive input events from the keyboard and touch screen (or mouse).
  - Disposable interface: an interface for disposable resources.
  - Actor class: this functions as a 2D graph node. An actor has a position, rectangular size, origin, scale, rotation, Z index, and color. This position corresponds to the unrotated, unscaled bottom-left corner of the actor. The position is relative to the actor's parent. The origin is relative to the position and is used for scale and rotation. An actor has a list of in-progress actions (from the Action class, described below) that are applied to it and are generally used to change its presentation (moving it, resizing it, etc.)
  - Action class: an abstract class that is attached to an Actor and performs some task, often over time.
  - TextureRegion class: defines a rectangular area of a texture.
  - Animation class: this stores a list of texture regions (TextureRegion class) representing an animated sequence. Each region of an Animation is called a key frame and multiple key frames make up the animation.
  - TiledMap class: represents a tiled map with the concept of tiles and tilesets. This class is used to load the map file created with the Tiled map editor onto the game and makes it possible to edit the layers and cells of this map.
  - OrthogonalTiledMapRenderer class: allows a TiledMap to be rendered.
  - AssetManager class: implements the Disposable interface and is used to load and store assets like textures, tile maps, sounds, music and so on.
  - GestureDetector class: an InputAdapter implementation that detects gestures (tap, long press, fling, pan, zoom, pinch) and hands them to a listener.
  - Dialog class: displays a dialog, which is a modal window containing a content table with a button table underneath it. Any type of widget can be added to the content table.
-

LibGDX itself makes use of OpenGL for graphic rendering. OpenGL is a widely used and supported 2D and 3D graphics API incorporating a broad set of rendering, texture mapping, special effects and other visualization functions. Furthermore, the framework makes use of the 2D physics engine Box2D to control the world physics. Box2D simulates rigid bodies in 2D and controls events such as collision detection, contact callbacks, friction, etc. Figure 3.7 shows the resulting class diagram for the multi-touch application implementation using the LibGDX framework. The following is a detailed description of each class.

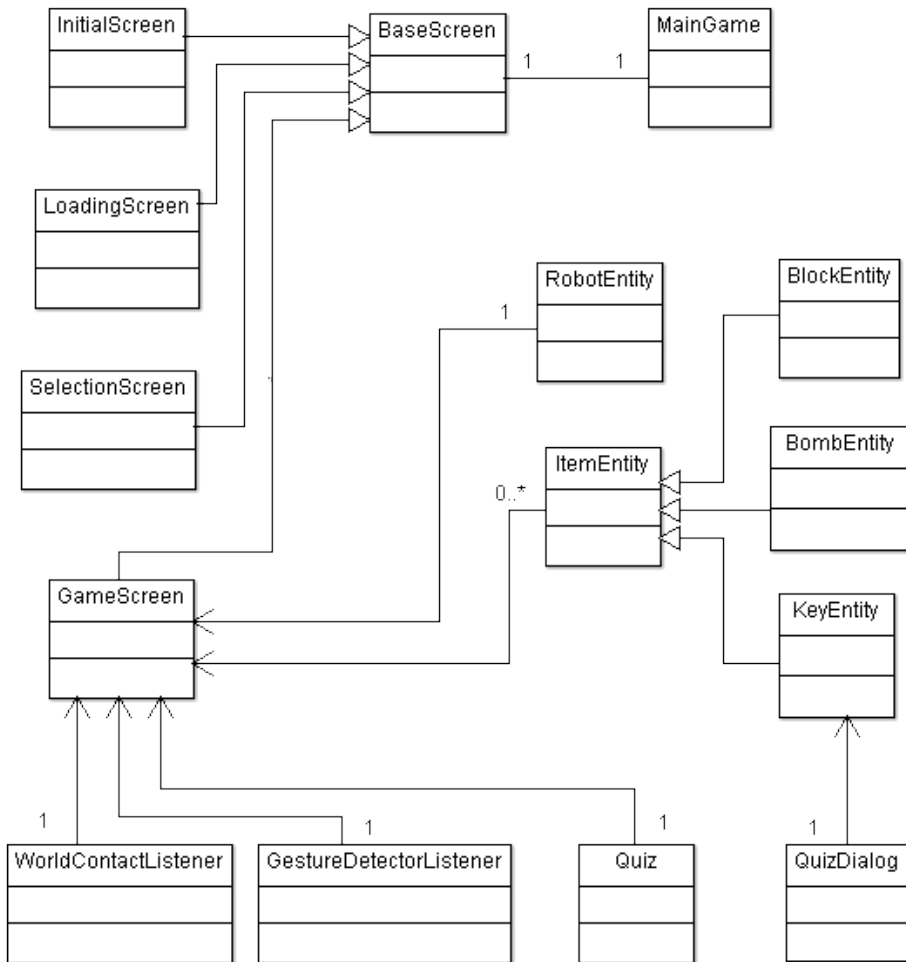


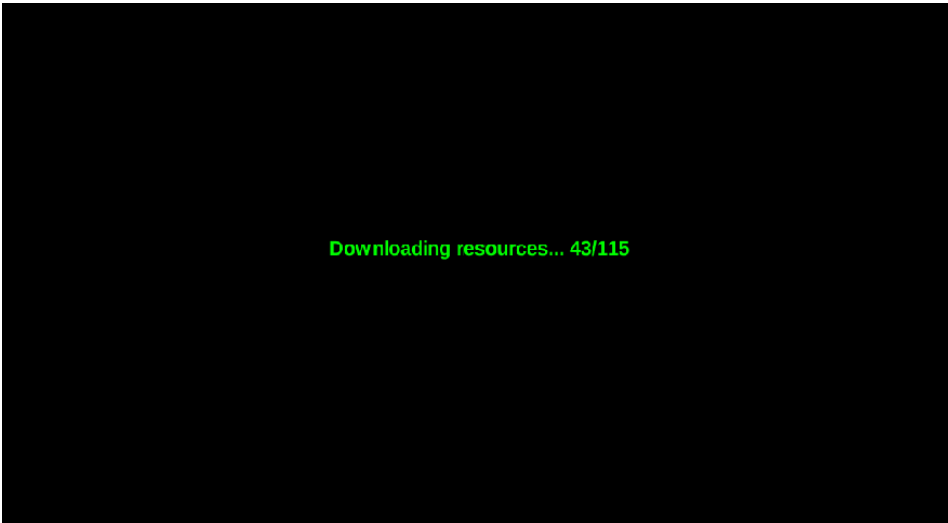
Figure 3.7: Quizbot for Tabletops and Handheld Tablets Class Diagram.

- **MainGame:** this is the class that is called by the application launcher (DesktopApplication for Windows, AndroidLauncher for Android) when the application is booted. The class serves several functions:
  - Set the application screen.
  - Establish a connection with the server.
  - Download resources (such as image and sound files) from the server and load them into an asset manager.
  - Get other information from the server, such as a list of other connected players.
  - Control connection and disconnection with the game server.
- **BaseScreen:** an abstract class that implements LibGDX's own Screen interface. It serves as a base for the other game screen classes.
- **InitialScreen:** as the name suggests, this is the screen that is called initially. It extends the BaseScreen class and simply offers a text field and a button. The text field is meant to be filled with the server address by the user. Once the button is pressed, the address text is sent to MainGame so the connection to the given server is established. An example of this screen can be seen in figure 3.8.



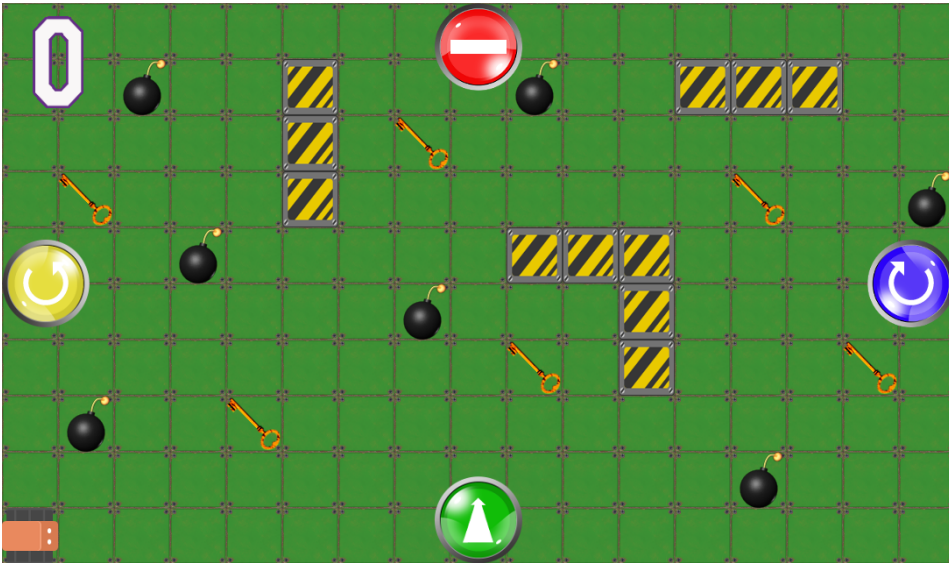
**Figure 3.8:** Quizbot for Tabletops and Handheld Tablets Initial Screen.

- **LoadingScreen:** extends BaseScreen and serves as a fill-in screen between InitialScreen and SelectionScreen. It simply shows the download progress of the resources followed by the loading progress of those resources into the asset manager. An example of this screen can be seen in figure 3.9.
-

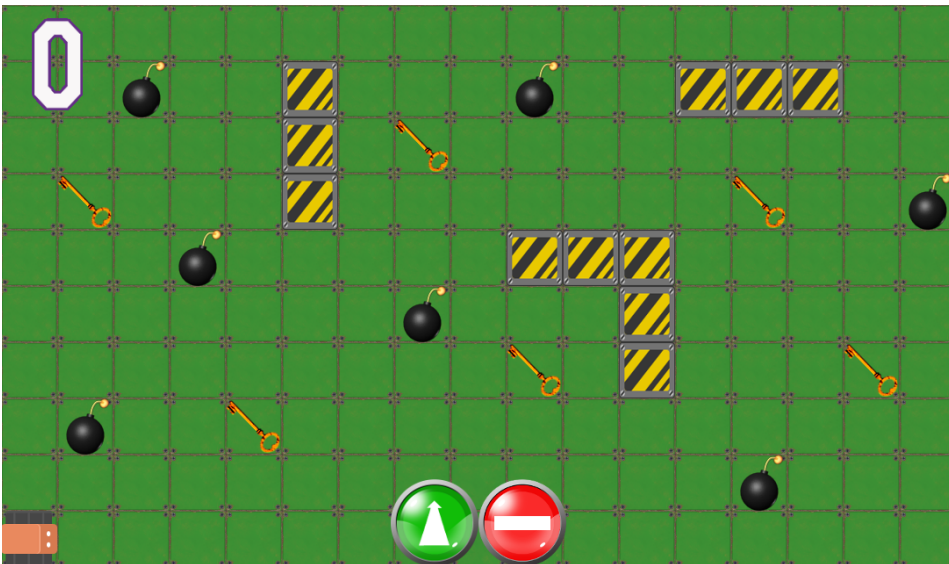


**Figure 3.9:** Quizbot for Tabletops and Handheld Tablets Loading Screen.

- **SelectionScreen:** extends `BaseScreen` and contains a text field where the player may introduce their name, as well as an image texture for each of the possible robot movement commands. Below each texture is a dropdown menu which the player may use to decide where to position the command button in question on the game screen. The positions are south (i.e. the bottom of the screen), east (right side of the screen looking from the bottom), west (left side of the screen), and north (top of the screen). For the tabletop, where players can be standing in any position around the table, this setup makes it possible to place any given button(s) right in front of the player. Figure 3.10 shows what the game screen looks like when the go forward button is positioned south, the turn right button is positioned east, the turn left button is positioned west, and the stop button is positioned north. The buttons can also be hidden from the screen, as can be seen in figure 3.11, in order to make it so that a player could have only a subset of the buttons available on their handheld tablet, while other players would have another subset of buttons. Figure 3.12 is an example of this screen.
  - **GameScreen:** extends `BaseScreen` and could be considered the game’s main screen, where all of the actual gameplay takes place. The gameplay elements that are contained in this class are as follows:
    - **TiledMap:** `GameScreen` creates a `TiledMap` using the Tiled map editor resource that was previously loaded into the `AssetManager` and renders it using `TiledMapRenderer`. This map occupies the entire screen, serving as a background of sorts, and all the events occur over it. The `TiledMap` object can also be edited dynamically during gameplay and `GameScreen` contains methods to do just that; add and remove items from the map by creating and setting new map cells.
-

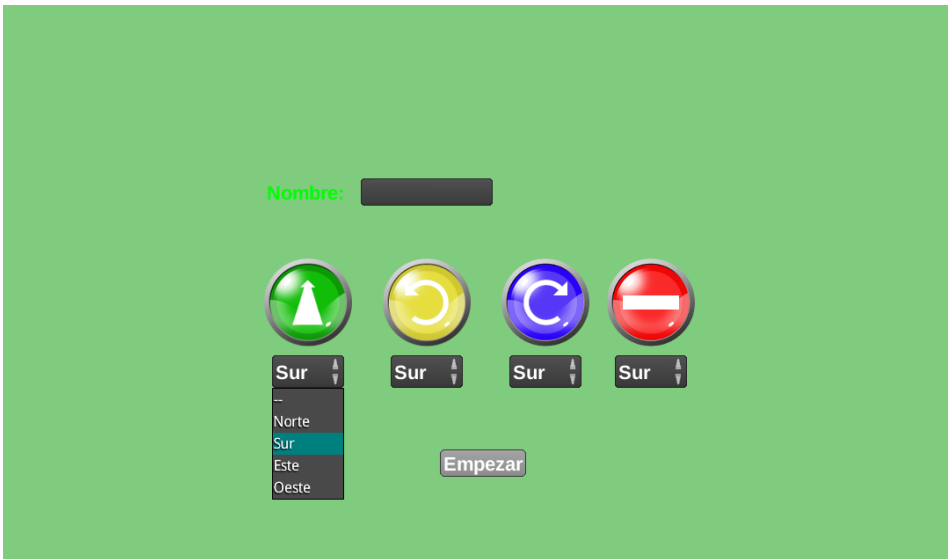


**Figure 3.10:** Quizbot game screen showing Go Forward at south position, Turn Right at east, Turn Left at west and Stop at north.



**Figure 3.11:** Quizbot game screen showing a subset of movement commands.

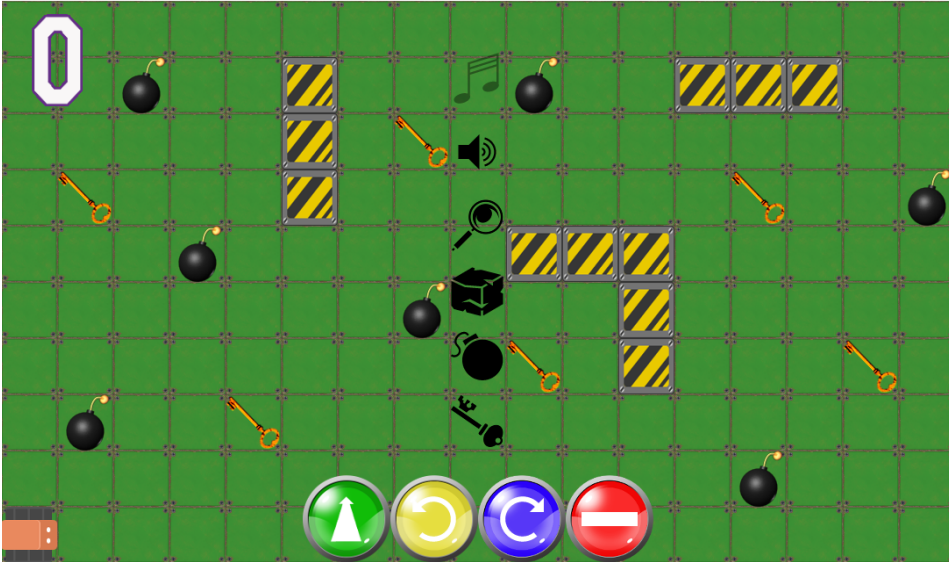
- Robot: the robot that the player controls is created using the RobotEntity class and is simply a LibGDX Actor. GameScreen creates and renders this actor, showing its movements and collisions.



**Figure 3.12:** Quizbot for Tabletops and Handheld Tablets Selection Screen.

- MyCamera: an auxiliary class extending LibGDX’s OrthographicCamera class, adding a bounding box to given coordinates (namely the TiledMap’s corner points) and ensuring that the camera cannot be panned outside of those coordinates (i.e. the player cannot pan infinitely through the game world).
- Stages: GameScreen contains several stages, each serving a different purpose. This is so that a combination of GestureDetecture and other InputProcessors can be used. The different stages are described below:
  - \* A main stage containing a viewport with a MyCamera class camera, and the robot actor.
  - \* A direction stage containing vertical and horizontal groups which in turn contain the robot movement command buttons. If in the SelectionScreen the user chose “south” for the button orientation, then the button would be added to a horizontal group positioned at the bottom of the stage. If the user chose “east” as the button orientation, the button would be then added to a vertical group positioned at the right side of the stage. If “west” was chosen, then the button would be placed in a vertical group at the left side of the stage. Finally, if “north” was chosen then the button would be added to a horizontal group positioned at the top of the stage.
  - \* A menu stage containing a table with menu buttons. These buttons include a music button to turn on or off the background music, a sound button to toggle the game sounds, a zoom button to toggle the ability to zoom in on the board or not, a bomb button to toggle the visibility of the bomb items on the board, a block button to toggle the visibility of the block items on the board, and a key button to toggle the visibility of the key items on the

board. The table itself can be made visible or invisible with a double tap gesture on the screen. Figure 3.13 shows the game screen with the menu visible.



**Figure 3.13:** Quizbot game screen displaying game menu.

- \* A count stage with a number indicating how many quizzes have been solved.
- \* A dialog stage containing a QuizDialog that is made visible when a quiz question or answer needs to be shown on screen.
- Animations: GameScreen contains two animations, one for when a wrong answer is selected and another for when the robot collides with a bomb. These animations can be seen in figures 3.14 and 3.15.
- Items: A list for each type of item (keys, blocks, bombs) is stored in order to control collisions.
- World: This is actually a Box2D class that manages all physics entities, dynamic simulation, and asynchronous queries.
- Sounds and music: A constant BGM is played during this screen. There are also sounds associated with selecting a correct answer, selecting a wrong answer, triggering a bomb explosion, and robot movement.

GameScreen is also in charge of several server communications, such as updating the robot's position. Only one client is actually in charge of the math and the physics to calculate the robot's trajectory and position, and this client then sends constant updates to the server about the robot's orientation and position. It is then the server that propagates these parameters to the rest of the clients.





Figure 3.14: Quizbot game screen displaying wrong answer animation.

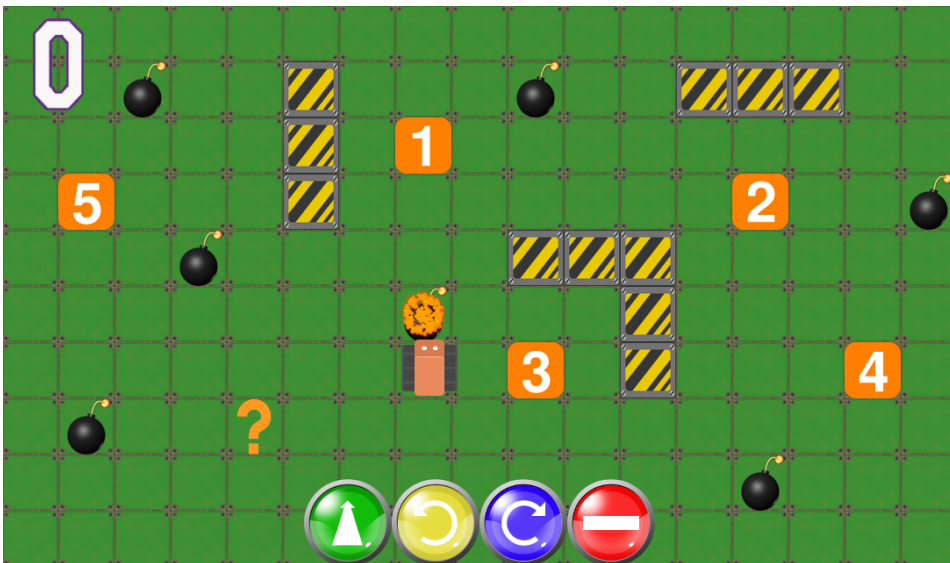


Figure 3.15: Quizbot game screen displaying bomb explosion animation.

- RobotEntity: a class that extends the LibGDX Actor class. The robot is the game’s main actor and the one that the player controls with the four movement commands. In order to calculate the robot’s new position in relation to its intended movement, a velocity and a direction vector are used, as well as the robot’s orientation (which is

calculated based on the turn left and turn right commands) and a fixed speed. This actor also has a texture that is simply the image of a cartoon robot. The robot texture is rotated in accordance with its orientation.

- **ItemEntity**: an abstract class which serves as a base to create all the game items. The class creates the necessary bodies, fixtures and shapes that Box2D needs for collisions and contacts.
  - **KeyEntity**: extends ItemEntity with the addition of a QuizDialog object. This entity represents the “key item” in the game.
  - **BlockEntity**: extends ItemEntity to represent the “block item” on the board.
  - **BombEntity**: extends ItemEntity to represent the “bomb item” on the board.
  - **QuizDialog**: a class that extends LibGDX’s Dialog class in order to insert an image and a text into the content table, as well a “close” button that closes the dialog window. The class also allows a second text to be added beneath the original text. This class is made visible when the robot actor makes contact with a key during normal mode. When the quiz mode is started, a QuizDialog is shown with the question image and text, as shown in figure 3.16. A QuizDialog is also made visible when the player double taps on the question cell or on an answer cell while in quiz mode, showing either the question image and text or the corresponding answer image and text. When the robot actor makes contact with a correct answer, a QuizDialog with the answer’s image and text is also shown. In some cases, the second text is also added to the dialog window, as shown in figure 3.17. This second text is meant to be informational only, providing additional information about the answer that aims to help the children learn.
  - **Quiz**: this class has three main items representing the current quiz. A state indicating whether the quiz is active (the game is in quiz mode) or inactive (the game is in normal mode and the quiz hasn’t been started yet), a question QuizDialog, and a list of QuizDialog answers.
  - **GestureDetectorListener**: as the name suggests, this class is an implemented listener for LibGDX’s GestureDetector class. The listener is implemented for the following gestures:
    - **Tap**: If the tap count is 1 and the game is in quiz mode, GameScreen is informed, checking if what has been tapped is a question or answer cell and making the corresponding QuizDialog visible in the positive case. If the tap count is 2, the visibility of the game menu table is toggled; if it was previously visible it becomes invisible, as vice versa.
    - **Pan**: This translates the game’s camera taking into consideration the bounding box and the camera zoom value.
    - **Zoom**: If enabled in the menu, this gesture zoom’s the game camera in or out.
-



Figure 3.16: Quizbot game screen displaying a Dialog with a quiz question.



Figure 3.17: Quizbot game screen displaying a Dialog with a correct answer and its informational text.

- WorldContactListener: implements Box2D's ContactListener interface, establishing the game's behavior when a collision is detected. If the collision is of the robot actor with a KeyEntity in normal mode, then the robot's movement is stopped and

the game goes into quiz mode. If the collision is of the robot actor with a `KeyEntity` in quiz mode, then the robot's movement is stopped and `GameScreen` is informed in order to check whether the selected answer is correct or not. If the collision is between the robot and a `BombEntity`, `GameScreen` is informed, initiating the explosion animation and resetting the quiz (and the robot's position, including stopping its movements). Finally, if the collision is between the robot and the `BlockEntity`, `GameScreen` is informed as well. In this last case, nothing is done to the robot's movement explicitly; `Box2D`'s physics are applied to the robot normally, stopping it if the collision was head on, or making it slide along the `BlockEntity` body otherwise.

### 3.3 Quizbot for Physical Spaces

Introducing physical spaces to traditionally sedentary games is a practice that is becoming increasingly popular. Physical body movements are proven to be essential for the enjoyment of life (Bowlby, 1969) and several works such as (Xie et al., 2008; Cheok et al., 2005; Schlieder et al., 2006) highlight the benefits of games which favor physical activity and make use of tangible objects. Therefore, a version of Quizbot using a mixture of physical and digital spaces for a Tangible User Interface experience was created.

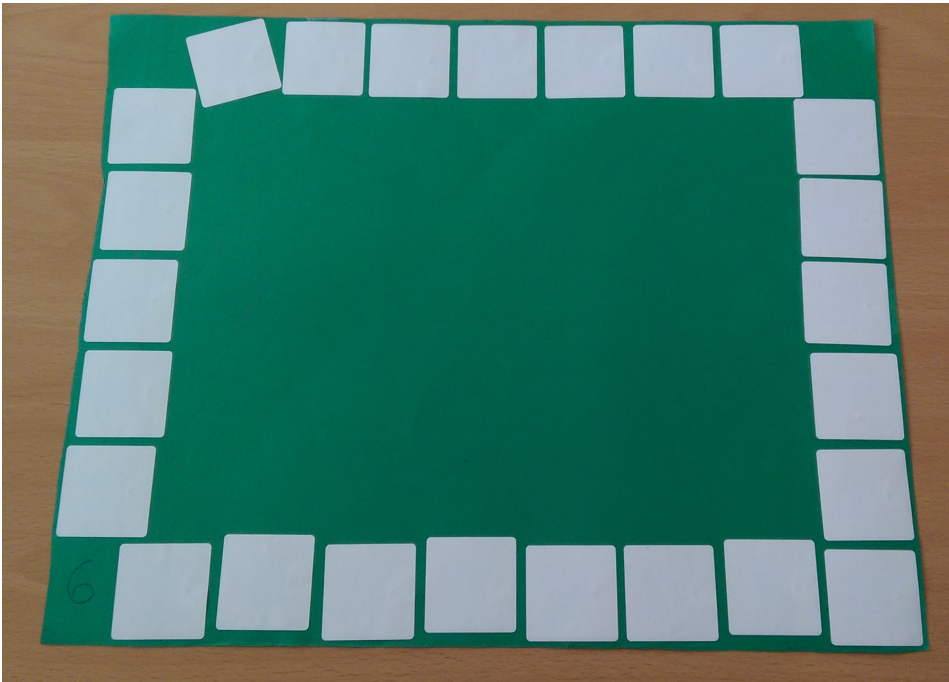
For this platform, several objects and devices were used to create Quizbot. The non-technological objects included interlocking foam mats for a 7m x 4m board, where each piece of the mattress represented a cell on the board. Foam tubes were used to represent “block items”, and inflatable rubber balls were used to represent the “bomb items” on the board. As for the technological aspect of the game, several Android handheld tablets were used as “key cells” to be placed on the foam mattress in their corresponding cell. Furthermore, a Lego Mindstorms robot was used as the actual robot actor to control on the board. Finally, in order to allow for communication between the board and the robot, RFID tags were placed around the “key cells” and the “bomb cells”, as seen in figure 3.18, and an Android phone connected to an RFID reader was mounted on the robot. The robot setup can be seen in figure 3.19. This communication is made through the game sever, where once a tag is read, the smartphone sends a message to the server about whether it was a key cell or a bomb cell (in case of the former, the ID of the cell is included), and the server then behaves in accordance to the message received. The setup with the game board can be seen in figure 3.20

A total of three native Android applications were developed in order to match the behavior of the TUI platform to the other two platforms. The first one, for the smartphone, uses the connected RFID reader in order to give each read tag an ID. The ID in this case refers to whether the tag belongs to a key cell or a bomb cell. In case of the former, each key cell has a different ID too. This information is stored in the server to be used by the other two applications during the actual gameplay.

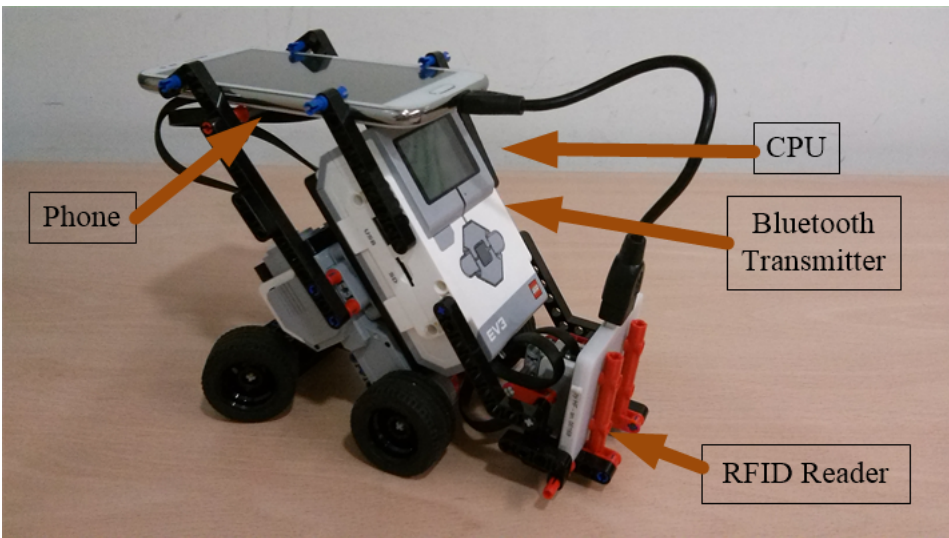
The second application is also for the robot-mounted smartphone. The application extends the already developed Tangibot application (Garcia-Sanjuan et al., 2016c), which allows the Lego Mindstorms robot to go forward, turn left, turn right or stop in accordance to the RFID tag that is read. These are the same movements that Quizbot can do. The RFID tags in this case are placed each in a different plastic stick, making each stick represent a different robot movement command. A player may then be able to use one or several of these sticks in order to move the robot. The movement module from Tangibot is extended with further functionality specific to Quizbot, which is notifying the server of any other type of read tags (namely, a “key item” tag or a “bomb item” tag). Furthermore, the application can receive certain notifications from the server in order to display information on the device screen. For example, after the server receives a bomb tag, it sends a return message to the smartphone application indicating that it should display an explosion video on the screen. The application also receives the same messages as the third application explained below.

The third and final application developed for this platform is for the Android handheld tablets which act as the “key item” cells on the board. The application consists in a first

---



**Figure 3.18:** RFID tags placed around a game item under the foam mattress.



**Figure 3.19:** Tangible Lego Mindstorms robot setup.

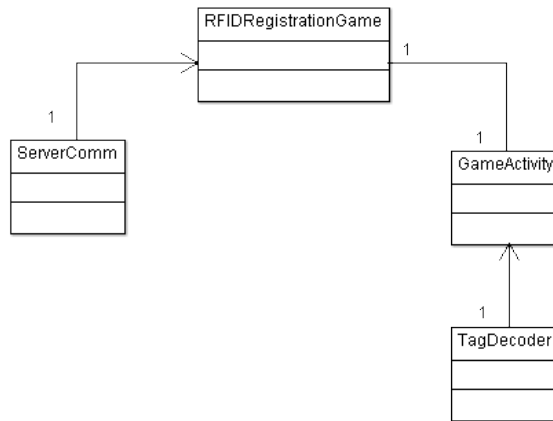


**Figure 3.20:** Quizbot in a Physical Space.

screen where the tablet is given an ID. This ID is then sent to the server and, if found (i.e. the first application created it and assigned RFID tags to it), the tablet is registered on the server as a key cell with that ID. After that, a second screen is displayed which acts much like the “key item” cells on the tabletop and handheld tablets platforms. When the game is in normal mode, the cell tablet displays the image of a key. After the robot reaches a key cell and reads one of the RFID tags surrounding the cell, the game enters into quiz mode and the reached cell displays the quiz question while the rest of the tablet cells display the possible answers. Once a quiz is completed, this application as well as the robot-mounted smartphone application display a “quiz ended” message. Finally, once the game is finished, both applications display a “game ended” message.

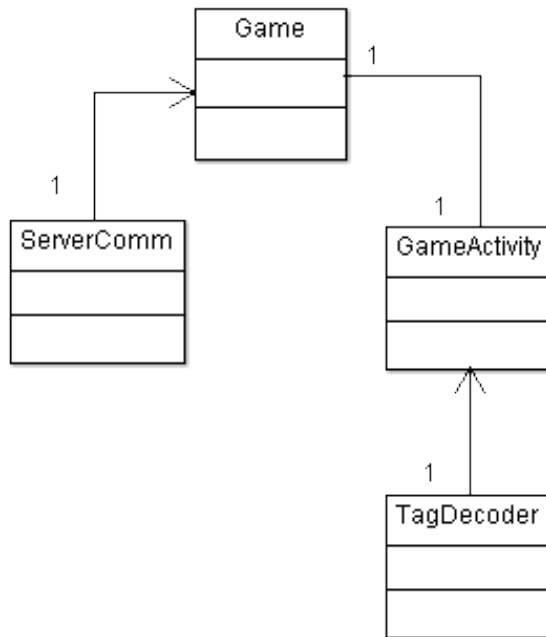
Figure 3.21 shows the class diagram for the first, RFID tag registration application. The following is an explanation of each class:

- **RFIDRegistrationGame:** The application’s main class which serves as a bridge between all the classes and controls the message flow between the application and the server.
- **GameActivity:** This class is in charge of the game display, where the user can see the read tags, assign them an ID and send them to the server. The user can also choose to clear the read tags. It also contains a listener for when a tag is read by the RFID reader.
- **ServerComm:** The class contains the events that the application can send to the server, such as connecting and disconnecting, as well as messages with the tag IDs.
- **TagDecoder:** This class is in charge of decoding the tags read by the RFID tag reader.



**Figure 3.21:** Quizbot for Physical Spaces Class Diagram: RFID Registration.

Figure 3.22 shows the class diagram for the second application, which is used during the game to read movement commands and game items. The classes are as follows:



**Figure 3.22:** Quizbot for Physical Spaces Class Diagram: Commands and Items Reader.

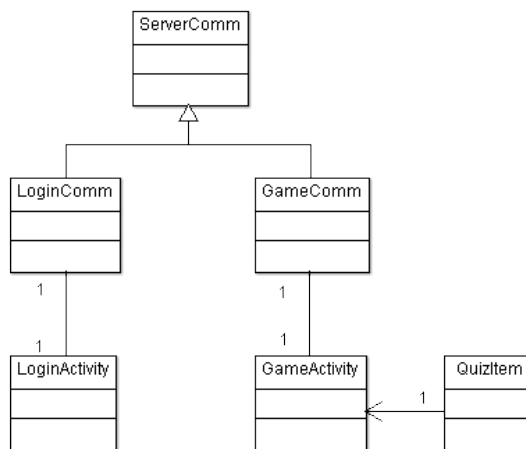
- Game: The application’s main class, which serves as a bridge between all the appli-



cation classes and processes the read RFID tags to decide which event and message to send to the server.

- **GameActivity:** The class is in charge of the game display, displaying “quiz ended” and “game ended” messages, as well as a bomb explosion video when Game processes a bomb tag.
- **ServerComm:** Contains the possible server communications, such as sending connecting, disconnecting and item cell reached events, as well as receiving events indicating that a quiz has ended or that the game is over.
- **TagDecoder:** The class is in charge of decoding the tags read by the RFID tag reader.

Finally, figure 3.23 shows the class diagram for the third application, in charge of the tablets’ behavior on the game board. The application classes are the following:



**Figure 3.23:** Quizbot for Physical Spaces Class Diagram: Key Item Cells.

- **ServerComm:** Contains the connection and disconnection events that are sent to the game server.
  - **LoginComm:** Extends ServerComm, adding login-related events such as getting resource names and downloading resources.
  - **GameComm:** Extends ServerComm, adding game-related events such as item distribution, quiz resetting, correct answer, wrong answer, quiz ended, and game ended.
  - **LoginActivity:** Displays the application’s initial screen, where the user must input the tablet’s ID. Furthermore, the class uses LoginComm to download the game resources from the server, displaying a download progress dialog as well.
-

- **QuizItem:** Class used to set the layout for the “key items”, whether displaying a key texture when the game is in normal mode or an image with the answer text (and possibly the informative text) during quiz mode.
- **GameActivity:** The class is called after LoginActivity is completed in order to control the actual game, where a QuizItem is displayed and a correct or incorrect answer image is displayed over it when the corresponding event is triggered by the server. The class is also in charge of changing the display when a quiz is ended or when the game is ended, as well as resetting the quiz by removing the correct answer displays when the robot collides with a bomb.

### 3.4 Quizbot Server

The server side of the game was developed using Node.js<sup>2</sup>, a JavaScript runtime built on Chrome’s V8 JavaScript engine that uses an event-driven, non-blocking I/O model. To establish the connection between the server and the game client(s), Socket.IO was used, enabling real-time bidirectional event-based communication. The server for the tabletop platform and the handheld tablets platform behaves in exactly the same way, but the TUI platform required some modifications.

The following are the events that are shared between all three platforms:

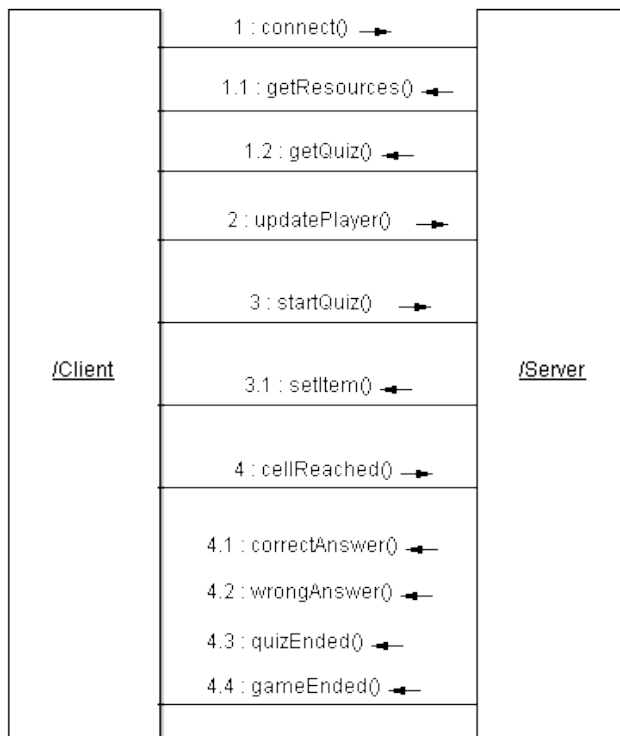
- **getResources:** a event that is sent to a client when it first connects with a list of resources to download from the server’s public directory. This list is created when the server is first booted by reading the file name of all the files in the public directory.
- **setItem:** triggered in response to an event from the client sent when the game is switching from normal mode to quiz mode. This event sent in a loop, once for each “key item” in the quiz, to all the connected clients with the information related to that item.
- **resetQuiz:** sent in response to an event from the client when the robot collides with a bomb.
- **correctAnswer:** when the robot collides with the question or with an answer cell on the board, the client sends a “cellReached” event to the server with that cell’s ID information. The server then checks whether that cell contains the correct answer and, if yes, broadcasts this correctAnswer event to all the clients.
- **quizEnded:** after establishing that a reached cell contains a correct answer, the server then proceeds to check whether no further correct answers remain in the quiz. If none remain, then the quiz is considered completed. However, the event is only broadcasted to the clients if there remain one or more quizzes in the list of quizzes, since the event contains the information of the next quiz to be answered.

---

<sup>2</sup><https://nodejs.org/>

- `gameEnded`: if, after establishing that a quiz has ended, the server also establishes that no further quizzes remain, then the game is considered ended and the `gameEnded` event is broadcasted to all the clients instead of the previously mentioned `quizEnded` event.
- `wrongAnswer`: if, instead of a correct answer, the cell reached contained an incorrect answer, then this event is broadcasted to all the clients with the cell ID so that a wrong answer animation is played over the cell in question.

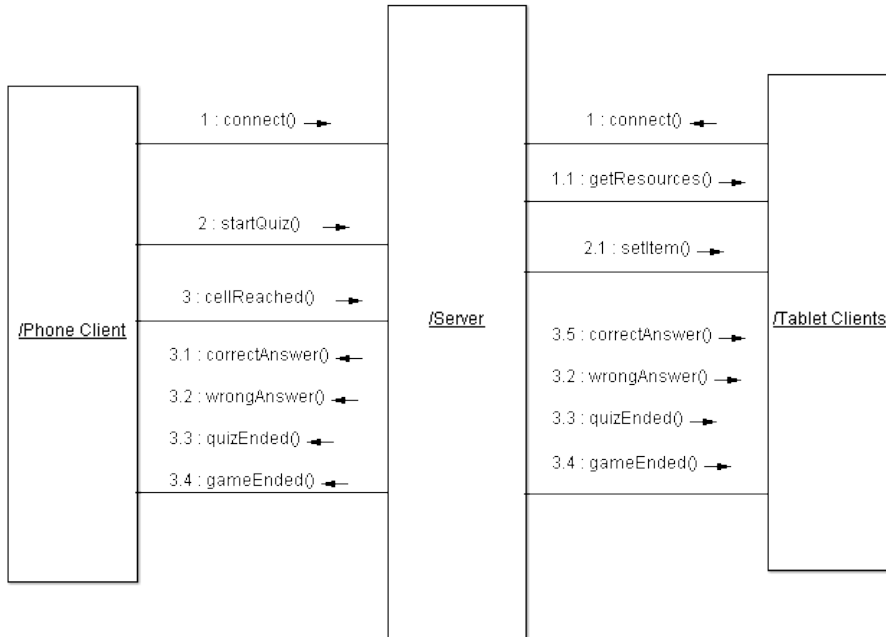
The following are the events that are specific to the tablet and handheld tablets platform. Figure 3.24 shows an example of some of part of the message flow between the server and a tablet or handheld tablet client.



**Figure 3.24:** Quizbot server communication with digital platforms.

- `socketID`: a message is sent to a client when it first connects to the server with its connection socket ID.
- `getPlayers`: a message is sent to a client when it first connects with the list of other players that are connected to the server at the moment of establishing the connection. If the list is empty, that means that client is the first to connect to the server,

making it the game’s “host” and in charge of doing the math and physics calculations mentioned in section 3.2.



**Figure 3.25:** Quizbot server communication with TUI platform.

- **newPlayer:** this event’s message is also sent when a client first connects, but it is broadcasted to all the other clients instead of being emitted to the newly connected client. The message contains the socket ID of the new client.
- **getQuiz:** this event is triggered in response to a message sent from the client when the game board is being loaded the first time. The message sent in response contains the quiz that is currently being answered (or is about to be answered if the game is still in normal mode and the quiz is inactive).
- **updatePlayer:** this event is triggered constantly by the host client through a homonymous event containing information related to the robot’s position and orientation. updatePlayer simply broadcasts the robot’s calculated position and orientation to the rest of the clients.
- **playerDisconnected:** this event is triggered whenever a client is disconnected in order to inform the other connected client about the occurrence. The event contains a message with the disconnected player’s ID.

- `newHost`: in the case where a disconnected player happened to be the game host and other players remain, a new host is selected from the list of still connected players and the corresponding client is then informed of their new role through this event.

Finally, there is only one event that is specific to the TUI platform. Figure 3.25 shows a partial example of the message flow between the server and the TUI clients during the game (i.e. after the RFID tag registration has been done).

- `bombReceived`: an event sent to smartphone app so it would display an explosion on the screen.

Besides enabling multi-player and multi-device experiences, the server allows for a lot of liberty as far as game content configuration is concerned. The downloadable resources can be changed at will at any given time; the new resources become available as soon as the server is restarted when the file names are read. Furthermore, the quiz content and configurations can be changed at will as well. All the quiz information is stored in a JSON file which is read every time a client is the first one to connect to the server. This way the server does not even have to be restarted whenever a new game is to be played. JSON (JavaScript Object Notation) is a lightweight data-interchange format and objects in this format can be sent in Node.js event messages without any issues. Listing 3.1 is an example of one quiz configuration in the game's JSON file. The file itself is a JSON array consisting of a number of "quiz" objects. A quiz itself consists of the following objects:

**Listing 3.1:** Quiz configuration in JSON file.

```
{
  "quiz": {
    "state": "inactive",
    "style": "choice",
    "values": {
      "count": 0,
      "visited": []
    },
    "question": {
      "text": "Select number 1.",
      "image": "question.png",
      "thumbnail": "questionT.png"
    },
    "answers": [
      {
        "text": "1.",
        "info": "",
        "value": { "correct": true, "id": 1 },
        "image": "1.png",
        "thumbnail": "1T.png"
      },
    ]
  }
}
```

---

```

    "text": "2.",
    "info": "",
    "value": { "correct": false, "id": 2},
    "image": "2.png",
    "thumbnail": "2T.png"
  }
],
"quizKeys": {
  "image": "key.png",
  "positions": [
    { "x": "1", "y": "6"},
    { "x": "4", "y": "2"},
    { "x": "7", "y": "7"}
  ]
},
"bombs": {
  "image": "bomb.png",
  "positions": [
    { "x": "2", "y": "8"},
    { "x": "3", "y": "5"},
    { "x": "7", "y": "4"}
  ]
},
"blocks": {
  "image": "block.png",
  "positions": [
    { "x": "5", "y": "6"},
    { "x": "5", "y": "7"},
    { "x": "5", "y": "8"}
  ]
}
}
}

```

- **state**: this is to indicate whether a quiz is inactive (it hasn't be loaded onto the game board and/or the game is in normal mode) or active (the quiz has been loaded onto the board and the game is in quiz mode). This value is "inactive" by default.
  - **style**: this value indicates whether the question is a choice, ordering or accumulation type of question. The server behaves differently for each type of question whenever it receives a "cellReached" event or a "resetQuiz" event. This aspect of the server was implemented in a modular way in order to make it easy to add different types of question styles. The main server module simply imports a "styles" module to which is passes the question style read from this value on three occasions: when a key cell is reached in quiz mode, after an answer is deemed correct, and when a quiz must be
-

Function	Style	Description
Pre-condition	Choice	Checks if the answer value is true or false, and the answer hasn't been selected before.
	Ordering	Checks if the answer value, indicating the position, is the same as the current quiz value.
	Accumulation	Checks if the current quiz accumulation value plus the answer value is less or equal to the goal value.
Post-condition	Choice	Checks if the quiz contains further true answers. If not, the quiz is ended.
	Ordering	Checks if the current quiz value is equal to the number of answers. If so, the quiz is ended.
	Accumulation	Checks if the current quiz accumulation value is the same as the goal value. If so, the quiz is ended.
Reset	Choice	Resets the quiz by clearing from the quiz values all the previously visited answers.
	Ordering	Resets the quiz by setting the quiz value to the initial value.
	Accumulation	Resets the quiz by setting the quiz accumulated value to 0.

**Table 3.2:** Description of how the server's Styles module works.

reset after a bomb collision. The styles module can be seen in more detail in table 3.2.

- **values:** this object serves to store the answering progress of the quiz. It contains the values needed to keep track of that progress, such as a counter for the current position an ordering style question is at, or the currently accumulated value in an accumulation style question.
- **question:** indicates the question to be answered in the quiz. The object includes the text of the question itself as well as the file name of the image to display in the Quiz-Dialog and the file name of the thumbnail image to display on in the corresponding cell on the board during quiz mode.
- **answers:** this object is a list of objects where each one represents a possible answer in the quiz, including both the right and wrong answers. Inside each answer object is the relevant information to that answer, including the answer text, the possible informational text to be displayed once the answer is selected, a value field which varies depending on the question style (in a choice style question, this would indicate whether the answer is a true or false one, in an ordering style question this would indicate the position of the answer in the given order, and in the accumulation style this would be the weight given to the answer), an image field with the file name of the associated image, and finally a thumbnail field indicating the file name of the thumbnail to be displayed on the board during quiz mode.

- quizKeys: this object represents the game's "key item" and contains the file name of the image to display on the board cells where a "key item" is located. It also contains a list of cell positions indicating where the "key items" are to be positioned on the game board.
- bombs: this represents the "bomb item" in the game and contains the file name of the image to display on the board cells where a "bomb item" is located. It also contains a list of cell positions indicating where the "bomb items" are to be positioned on the game board.
- blocks: this represents the "block item" in the game and just like the "quizKeys" and "bombs" objects, it contains the file name of the image to display on the board and a list of cell positions indicating where the "block items" are to be positioned on the board.

The quizKeys, bombs, and blocks values are ignored on the TUI platform since these items are physical objects that are placed manually on the physical board.



# Evaluation

In order to evaluate Quizbot on all three platforms, an experiment with 80 primary school students (of which 45% percent were girls and 55% were boys) between the ages of 9 and 10 was carried out at the Polytechnic University of Valencia Summer School, bringing with that the additional benefit of the children being from different schools with different curriculums. The children were separated into 10 groups of 8 and were made to test the three platforms in different rotations. For example, one group would start with the tabletop then move onto the handheld tablets and then onto the TUI, while another group would start with the TUI platform then move onto the tabletop and then onto the handheld tablets. This ensured that the order in which a platform is tested would not affect factors such as the enjoyment or learning factor. The questions to resolve on the platforms were also rotated in order to ensure that any possible variability in problem difficulty would not have an effect on the children’s impression of the platform. The questions themselves were taken from third and fourth grade school textbooks.

For each group on each platform, four children were playing at any given moment while the other four would observe from the sidelines. They would then switch after three minutes of gameplay and then back again after another three and so on, for a total of 18 minutes of gameplay. This does not include the time it took for them to complete a trial question at the beginning of each platform’s session. The children were each given control over one command at a time and were left to their own devices as far as everything else was concerned. Throughout all the activities, several observations were made of the children’s behavior. Furthermore, at the end of each group session, after a group would have tried out Quizbot on all three platforms, a questionnaire was passed out to each child in order to get their feedback on the experience. The questionnaire itself is a Fun Toolkit (Read and MacFarlane, 2006; Read, 2008) questionnaire adapted to this experiment. Table 4.1 shows the questions that were asked in the questionnaire. Questions 1 to 6 use a Smileyometer in order to measure how much fun the children had on each platform and how easy they found controlling the robot was on each platform. Questions 7 to 10 use a Fun Sorter in order to measure on which platform the children thought they performed better and worse, and on which platform the children had the most and the least fun. Questions 11 and 13 use

#	Question
1	How much fun did you have with the game on the floor?
2	How much fun did you have with the game on the tablets?
3	How much fun did you have with the game on the big table?
4	How easy was it to handle the robot on the floor?
5	How easy was it to handle the robot on the tablets?
6	How easy was it to handle the robot on the big table?
7	With which version do you think you did best?
8	With which version do you think you did worst?
9	With which version did you have most fun?
10	With which version did you have least fun?
11	Would you like to play again in class?
12	In what subjects would you play?
13	Would you like to play again outside class?
14	Would you prefer to play alone or with friends?
15	What would change in the game to like it better?

**Table 4.1:** Post-game session questionnaire

an Again-Again Table where the children can report the likelihood with which they might play the game on each platform inside a classroom and outside of one. Finally, questions 12, 14 and 15 are additional, Quizbot-specific questions in order to have a better grasp on the type of quiz questions the children prefer, as well whether they prefer playing in collaboration with friends or whether they prefer playing alone. The last question is simply for future reference, in order to make Quizbot more appealing and therefore possibly more effective.

## 4.1 Qualitative Results

The results obtained from the Fun Toolkit questionnaire are reported in this section. The questions were split into groups where the same factor was being measured for the different platforms in order to see how the children perceived the platforms.

A Wilcoxon test was used on the Smileyometer results in which the questions were paired by platform (tabletop, tablets, tangible) for each measurement factor (fun, ease of use). In this case, because three independent variables are being compared, the Bonferroni adjustment was used, making the significance level  $0.05/3 = 0.017$ .

For the tabletop-tablets fun comparison, the test concluded that there was no statistically significant difference ( $p = 0.461$ ). For the tabletop-tangible fun comparison, the test once again concluded that there was no statistically significant difference ( $p = 0.06$ ). Finally, for the tablets-tangible fun comparison, the test concluded that there was no statistically significant difference ( $p = 0.019$ ).

The same tests were made for the ease of use Smileyometer results. For the tabletop-tablets comparison, the test concluded that there was no statistically significant difference ( $p = 0.02$ ). For the tabletop-tangible comparison, the test once again concluded that there

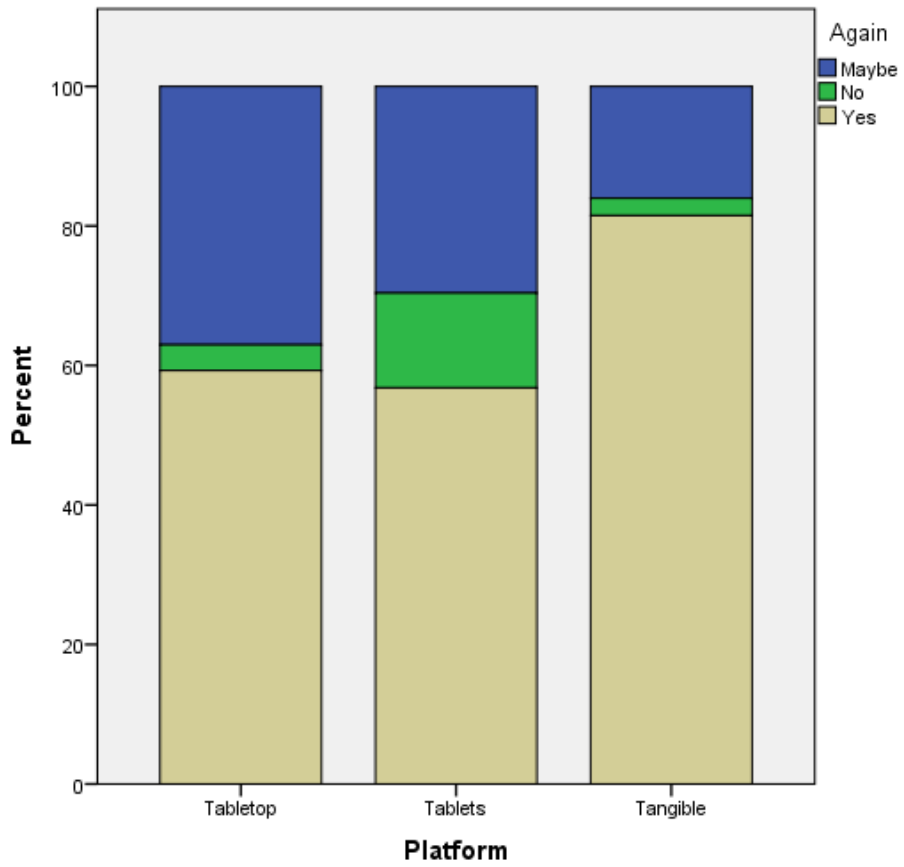
	<b>Best</b>	<b>Intermediate</b>	<b>Worst</b>
Easy to use	TUI (1.55)	Tabletop (2.14)	Tablets (2.26)
Fun	TUI (1.36)	Tabletop (2.04)	Tablets (2.53)

**Table 4.2:** Results for questionnaire Fun Sorter. Mean score for each platform is shown between parenthesis.

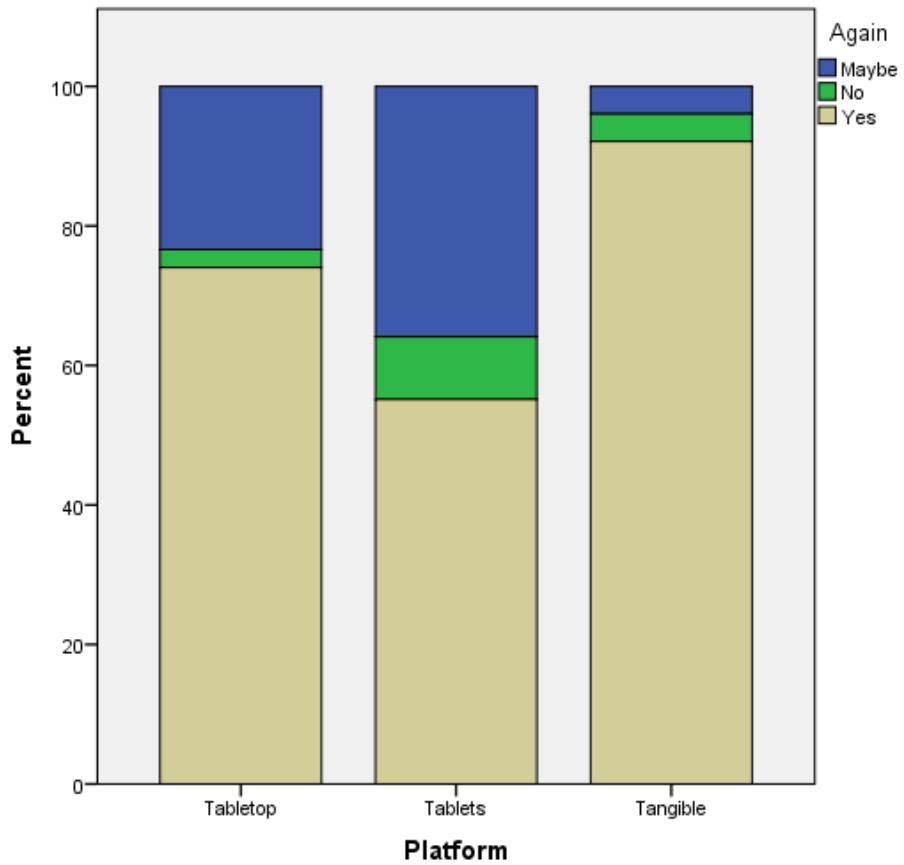
was no statistically significant difference ( $p = 0.438$ ). Finally, the tablets-tangible comparison is the only one where the test concluded that there was a statistical difference ( $p = 0.005$ ).

The results from the Fun Sorters where the children's platform preferences for the fun and the ease of use factors were asked explicitly are shown in 4.2. The average score is shown for each platform. This score was established by assigning 1 point to the platform that was chosen as the best, 2 points for the platform that was chosen as second best, and so on. This means that the closer the score is to 1, the better it is.

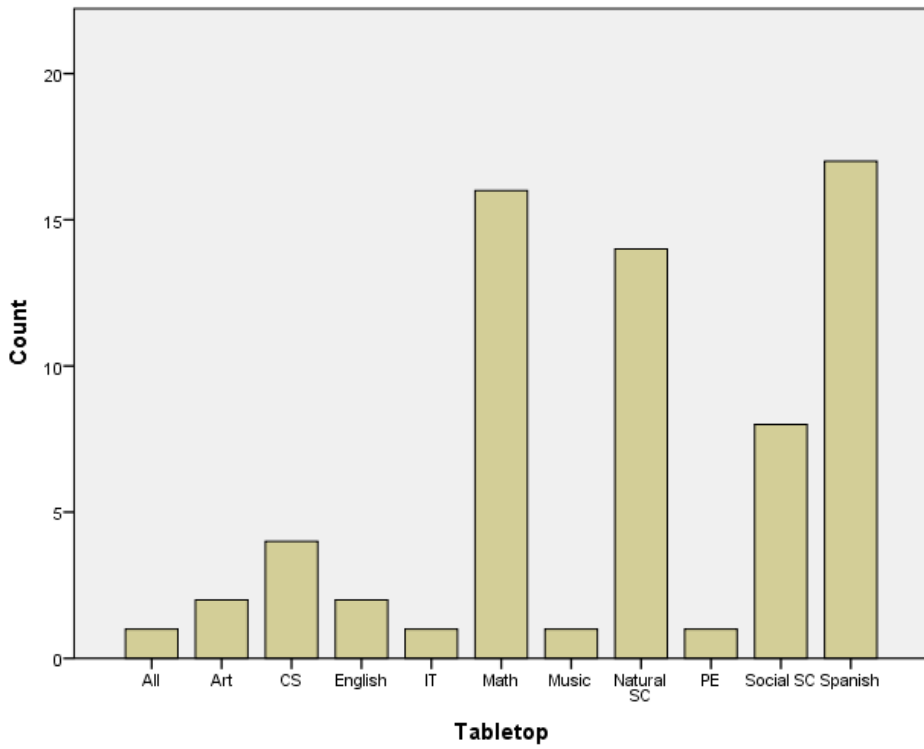
Figures 4.1 and 4.2 show the results of the Again-Again tables in which the children state their intention of playing again on each platform in class and outside. Figures 4.3, 4.4 and 4.5 show which school subject the children prefer for the quiz questions on each platform. Figures 4.6, 4.7 and 4.8 show the ratio of children who prefer playing alone vs. with friends on each platform. Finally, figure 4.9 shows some of the changes that the children suggested for Quizbot.



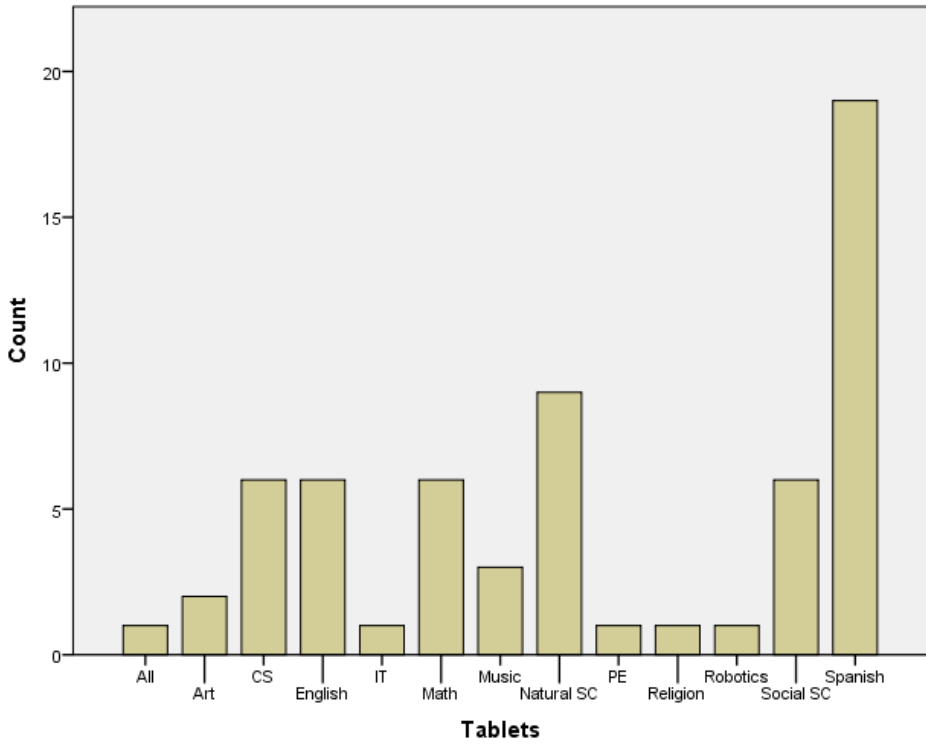
**Figure 4.1:** Results for questionnaire Again-Again table, stating desire to play again in class.



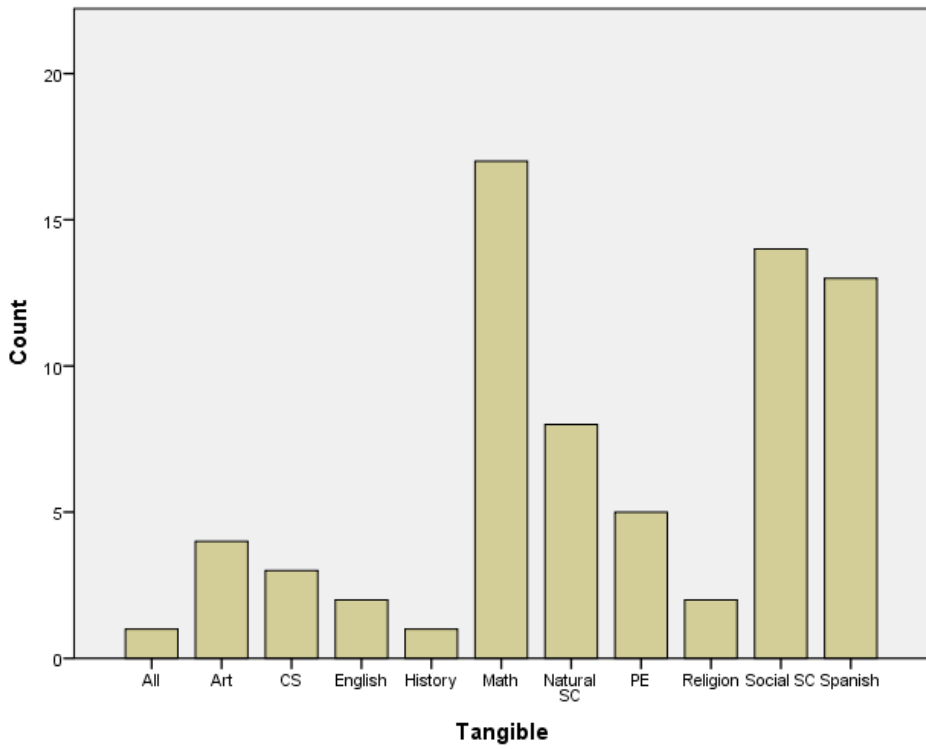
**Figure 4.2:** Results for questionnaire Again-Again table, stating desire to play again outside class.



**Figure 4.3:** Results for which school subjects are preferred for questions on the Tabletop platform.

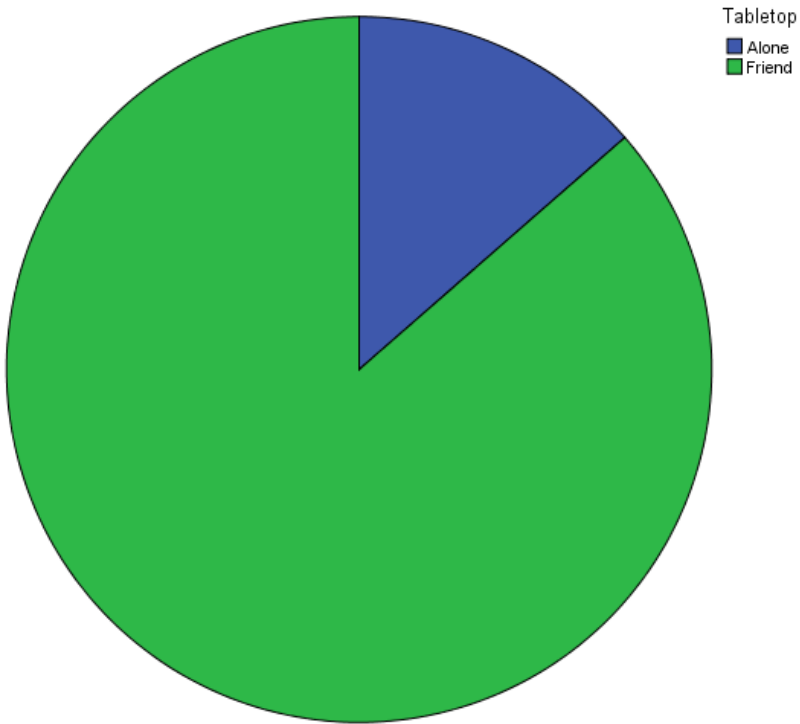


**Figure 4.4:** Results for which school subjects are preferred for questions on the Tablets platform.

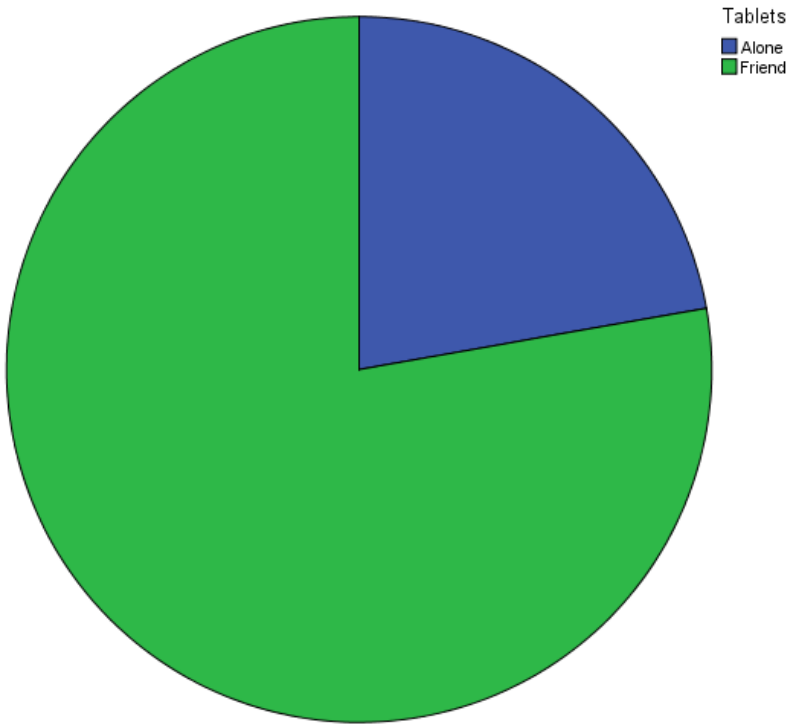


**Figure 4.5:** Results for which school subjects are preferred for questions on the TUI platform.



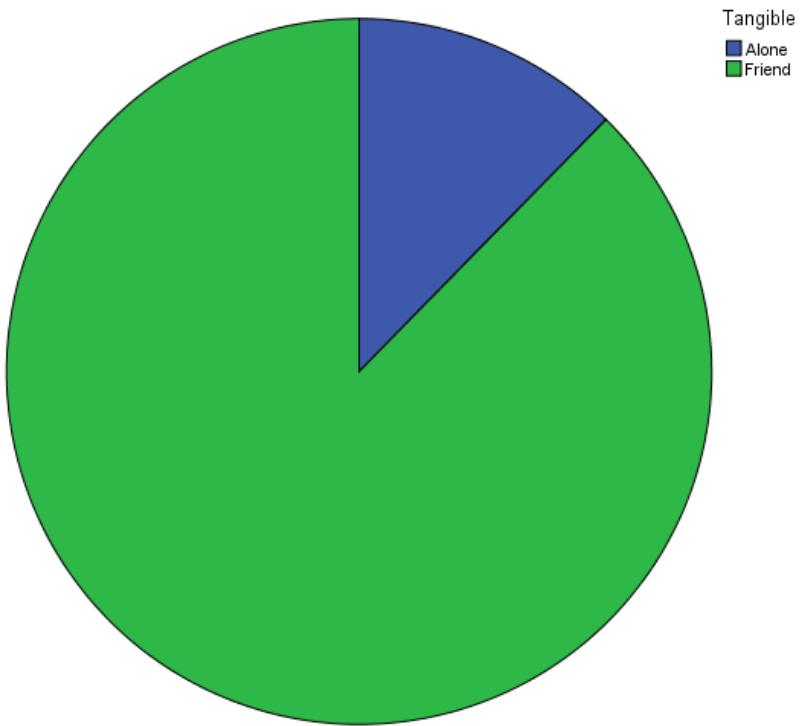


**Figure 4.6:** Company preference for Tabletop platform: 10% alone, 90% with friends.



**Figure 4.7:** Company preference for Tablets platform. 16.25% alone, 83.75% with friends.

---



**Figure 4.8:** Company preference for TUI platform: 7.5% alone, 92.5% with friends.

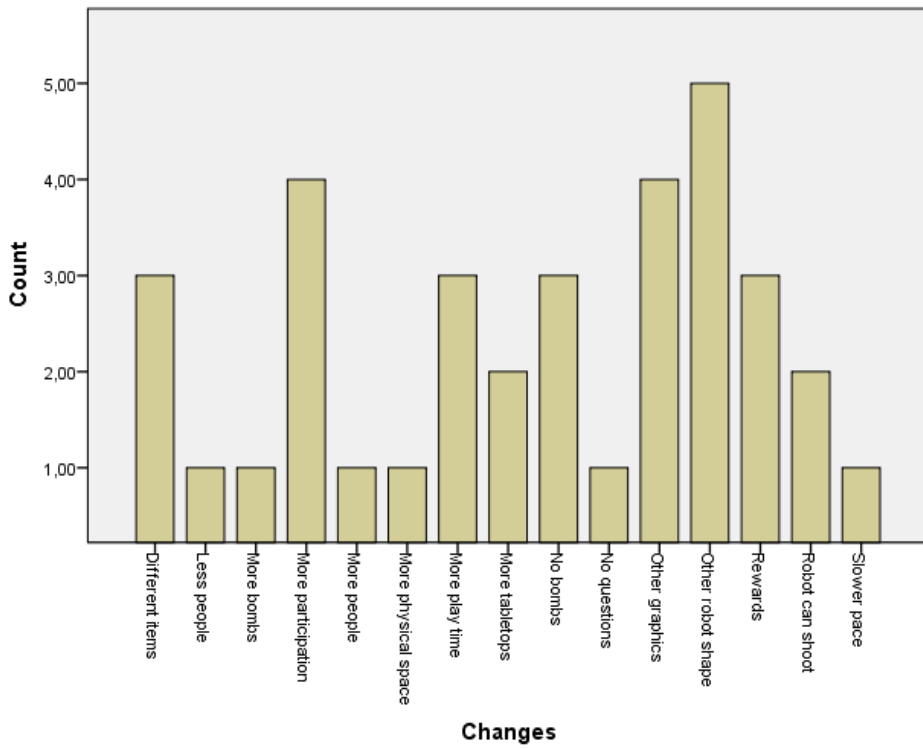


Figure 4.9: Changes suggested for Quizbot.

## 4.2 Observational Results

Throughout the game sessions, several observations were made of the children's general behavior and some patterns were extracted from these observations. Overall, the patterns were very similar over the three platforms, but some were repeated more frequently in one or two compared to the remaining platforms. However, a few platform-specific observations were also made. All of these observations are described below.

The action that was most frequently observed on all three platforms was that of planning. Whether it was at the beginning of each quiz or after a correct (and sometimes incorrect) answer, the children would stop and discuss which path to take to get to the next question. Some of the discussion revolved around whether the robot would be able to pass between two items on the board or not. Sometimes, they would plan ahead for several answers. However, there were also some cases where no planning was done and a couple of children would take charge and try different answers randomly. It was not only the children who were playing at the moment who planned; the children watching from the sidelines were also observed planning in hushed voices for when it was their turn to play.

Another action that was observed frequently was that of exploration. Whenever a new quiz would start, the children would check out all the answers before starting the planning process. This was observed most frequently on the TUI platform, especially among the children watching from the sidelines. During the exploration and planning processes, a lot of knowledge sharing took place too, especially if a child was sure of an answer or if someone asked a question.

A lot of negotiation in different forms took place on all three platforms too. For example, sometimes the children would discuss whether a set of answers is correct or not and would then agree to check one answer out and then another. Negotiations related to path planning also took place, where they would evaluate whether it was worth risking a shorter path containing bombs or if it was better to play it safe and take a longer path. Some sub-groups would also negotiate which movement command each person would have whenever it was their turn to play. This last type of negotiation was observed most frequently on the TUI platform and sometimes on the tablets platform, but rarely on the tabletop.

In most groups, one of the children would eventually take on a leadership role, ordering movements constantly. Most of the children would shout for the robot to be stopped, especially when it was about to collide with a bomb, making some children either avoid having that movement command or purposely ask for it, but the group leaders would shout out all the movements, telling the others when to go forward or when to turn.

Team Wranglers also appeared in some groups, where children would get fed up with waiting for someone to perform a movement command and would either invade the other's workspace (in the tabletop and tablets case) or grasp the other child's hand to force them to perform the wanted command.

In some groups, the children waiting on the sidelines would collaborate with those currently playing by telling them the answer or warning them about a bomb. This occurred most frequently on the handhelds platform, but also sometimes on the other two platforms. However, the children on the sidelines were more frequently found trying to annoy those playing by counting down the time for their turn to end, taunting them, asking for them to

---

collide with a bomb or to choose a wrong answer, giving them wrong answers, or actually sabotaging by invading their interface.

There were also cases where one of the players would sabotage the rest by constantly turning the robot or stopping it as soon as it started moving. In these cases, the other children would either tell them off or, in a few cases, physically stop them by grabbing their hand.

Overall, there were several groups with good coordination and groups with bad coordination. Sometimes a person would know and say the correct answer but the others would ignore them, causing them to sulk and ignore the game. In some cases, after answering wrongly, part of the group would sulk and momentarily stop playing. There were also cases where someone would try to cheer up the rest of the group and encourage them to try another answer.

As far as individual platform observations go, a couple of children complained about the warm air the tabletop releases, as well as about having to read the question and the answers upside down (for those standing in the north position). In the latter case, the person standing in the south position would help by reading the text out loud.

While playing on the tablets platform, the children would sometimes stand up when they get excited (such as when they answer something correctly or, in the case of the children on the sidelines, a wrong answer is chosen). The children on the sidelines would also stand up sometimes to have a better view of all the tablets, even though they can view one or two tablets easily from their position. A lack of coordination was also observed when it came to the two children with the turning movements; they would often turn the robot left and right at the same time, causing it to stay in the same position. They would also often call out an answer to go to by saying “This one!” while pointing at their own tablet, causing the others to ask “Which one?” in return.

Finally, when faced with the TUI platform, several children would make joyous exclamations such as “That’s so cool!” or “This is great!” and so on. In a few cases, the children would make the robot purposefully collide with the blocks. There were also cases where the robot would be dismantled because of the children’s rough handling (whether because of the colliding or because they would move the robot manually). On some occasions, the children who are supposed to be on the sidelines would stay on the board to observe the actions of those who are playing, while on others they would move around the board to play with the foam blocks or the rubber balls.

### 4.3 Discussion

The main purpose of the qualitative results using the Fun Toolkit was to compare the three platforms in order to see if any one platform would stand out from the rest. Overall, it seemed like the children’s preference was the TUI, but only one significant result was obtained that partially confirms that.

The Smileyometer showed that the tangible interface was easier to use than the handhelds, and that is in accordance to the Fun Sorter results shown in table ???. This could be due to a combination of lack of a public workspace which would make knowledge sharing easier and a general higher difficulty observed with the entirely digital version of the game. The reason only one significant result was obtained from the Smileyometer could be the

---

fact that, according to (Read and MacFarlane, 2006), the Smileyometer is not very useful with children younger than 10 years old because the variability of their responses is very low; they tend to choose the highest (most positive) score. In that regard, the Fun Sorter results showed the differences between the three platforms more clearly. Table ?? shows that the tangible platform was both the most fun and the easiest to use, while the tablets were both the least fun and the least easy to use, which suggests a correlation between the two variables. The reason behind these results could be that the TUI was more intuitive for the children. The tangible game being a generally rarer type of activity might also affect the fun factor in this case.

The results of the Again-Again tables (figures 4.1 and 4.2) show a mostly positive reaction to all three platforms, which could possibly be related to the children's age and their eagerness to play most of the time. This could be considered a positive thing since the intention was to make CPS skill enhancement fun so that the activity would be repeated willingly, thus helping to further enhance the children's Collaborative Problem Solving skills.

Figures 4.3, 4.4 and 4.5, which display what subjects the children would like to study using the three platforms, do not show much variety between the subjects the children chose based on platform, but there is somewhat more of a variety of subjects on the TUI platform. This could be due to the wider options this platform provides. For example, Physical Education related activities would be harder on the digital-only platform.

When asked whether the children would rather play Quizbot alone or with friends, an overwhelming number responded that they would prefer to play with friends. This is a positive result considering the purpose of Quizbot is to enhance Collaborative Problem Solving, which requires the participation of more than one agent. The handheld tablets might have the highest number of replies indicating they would rather play alone due to children perceiving tablets as generally private devices.

On the last question of the questionnaire, where the children were asked about any changes they would make to Quizbot, it can be noted that most of the changes suggested by the children are aesthetic, suggesting that visually pleasant items are more appealing, which is important to take into consideration when creating something with the intention of being used repeatedly.

As for the observations that were made during the experiment, a lot of it involved seeing communication, negotiation and planning taking place, which is in accordance with this work's goal. Organization varied between the different teams, mostly depending on whether there was one or two children sabotaging the activity or not, which could be attributed to children simply acting their age. Sometimes, better organization simply took longer, waiting instead for a leader to manifest in the group. The other roles discussed in section 1.2 were also present in different measures in each group.

The three main CPS competencies also discussed in section 1.2 were clearly observed taking place during the experiment. The children would share their knowledge where required, take action to solve the given questions and maintain some level of organization. The fact that improvement in some of these aspects could be observed already shows that Quizbot fulfills its intended purpose of encouraging the practice of the CPS sub-skills and CPS skills in general.

On a platform-specific level, the reason more exploring took place on the TUI platform

---

could be the fact that the children had to move around to explore, and that is precisely what the children wanted. It would also explain the constant standing up on the other platforms. More negotiation was observed on the TUI platform as well, at least when it came to negotiating what movement command (which could be considered a tool) each child would have. Since this was observed on the handhelds as well, albeit to a lesser extent, it could be related to the fact that it is easier to move the movement commands around on these two platforms. The only slight drawback related to the TUI is that it could be a little distracting for children in that age range still.

The tabletop platform had the one flaw of having to read text upside down depending on where a person was placed, which could be attributed to its limited workspace dimensions mentioned in section 1.1.

Finally, the handheld tablets provided a mixed bag of results. On the one hand, the private space seemed to have made coordination more difficult for the children because they would point at their own tablet a say “here” or “there” when referring to a point on the board to go to. However, this can be seen as an opportunity to improve the children’s communication skills by encouraging them to be more specific and descriptive with their language.

---



# Conclusions and Future Work

## 5.1 Conclusions

This work starts by presenting the many soft skills that are required of a student today, and the consolidation of said skills into what is referred to as Collaborative Problem Solving skills. These skills can be nurtured and enhanced in many ways, but one way that has been proven to be effective for learning in general is through video games. However, depending on the platform used, the activity may prove to be more or less effective.

Reviewing other works that are related to the subject at hand revealed that, while it is generally agreed upon that collaborative problem solving skills need to be developed in all students, there are few works documenting a skills enhancement process using anything but obsolete technologies. TUIs in particular seem to be used for learning and education in general but never for CPS skills enhancement, even though the types of activities that could be done on this platform are virtually endless. Thus, a multi-platform CPS skills enhancement tool called Quizbot was developed in order to not only encourage using CPS skills, but also to make use of the different advantages each platform may offer.

Quizbot is a mixture between a board game and a quiz-solving game, where the users control a robot, moving it on a board with cells containing different game items. Some game items trigger quizzes that the players must answer, also by guiding the robot to the correct answer(s). The game presents a CPS scenario by urging the players to coordinate their actions to make the robot move, plan the robot's route and share their knowledge to answer the quiz questions.

The first of the three platforms Quizbot was developed for is a multi-tactile tabletop, which provides a public space where players can share their knowledge with more ease. The second is a multi-tactile handheld platform where the board can be viewed on several tablets, making it possible to give each player their own private space. The third and last platform is based on a Tangible User Interface where the robot, the game board, and even the robot movement commands became physical objects.

An experiment with 80 summer school students was performed where the students were split into groups of 8 to try out the three platforms in a random order. The children

---

were observed without interference while they played, and at the end of each group session, a questionnaire was handed out. The questionnaire itself provided little insight on the differences between the three platforms beside the fact that the TUI was both the most fun and the easiest to use. However, it did show eagerness to play again. The observational results of the experiment on the other hand provided feedback on concrete differences between the three platforms, as well as verification of Quizbot serving its intended purpose by encouraging the usage of the skills associated with CPS.

## 5.2 Future Work

In the future, further experiments using Quizbot will be performed in order to test the effectiveness of the tool itself by preparing CPS tests and comparing students' performance on these tests before and after playing with Quizbot. Experiments in order to compare the effectiveness of Quizbot on each platform individually will also be performed. The changes suggested by the children in the experiment described in this work will also be taken into consideration in order to make Quizbot more appealing.

A complementary tool will be developed, allowing teachers and educators to manage educational activities, add new content and define specific CPS tasks. Furthermore, the three platforms will be tested in a pediatric hospitalization context for fomenting emotional intelligence (perception, assimilation, understanding and emotional regulation).

Lastly, the board game created as the base for Quizbot will be used to create other tools for stalling cognitive deterioration in the older population or for developing cognitive abilities in younger children.

## 5.3 Publications

This work is presented in an international conference paper:

- F. Garcia-Sanjuan, J. Jaen, and S. Jurdi. Towards Encouraging Communication in Hospitalized Children through Multi-Tablet Activities. In *Proc. of XVII International Conference on Human Computer Interaction (Interacción'16)*, page In press, 2016a

## 5.4 Acknowledgements

I would like to thank Fernando García Sanjuan for being this work's experimental director. This work is supported by Spanish Ministry of Economy and Competitiveness and funded by the European Development Regional Fund (EDRF-FEDER) with Project TIN2014-60077-R.

# Bibliography

- A. N. Antle and A. N. Exploring how children use their hands to think: an embodied interactional analysis. *Behaviour & Information Technology*, 32(9):938–954, Sept. 2013. ISSN 0144-929X. doi: 10.1080/0144929X.2011.630415. URL <http://www.tandfonline.com/doi/abs/10.1080/0144929X.2011.630415>.
- R. Ballagas, R. Ballagas, M. Rohs, J. G. Sheridan, and J. Borchers. BYOD: Bring Your Own Device. *IN PROCEEDINGS OF THE WORKSHOP ON UBIQUITOUS DISPLAY ENVIRONMENTS, UBICOMP, 2004*. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.9939>.
- J. Bowlby. *Attachment and loss*. Basic Books, New York, 1969. ISBN 9780465097166.
- M. Brayshaw and N. Gordon. Using motivation derived from computer gaming in the context of computer based instruction. In *2016 SAI Computing Conference (SAI)*, pages 828–832. IEEE, July 2016. ISBN 978-1-4673-8460-5. doi: 10.1109/SAI.2016.7556074. URL <http://ieeexplore.ieee.org/document/7556074/>.
- S. P. Brophy and S. P. Computer partner in the classroom. In *The first international conference on Computer support for collaborative learning - CSCL '95*, pages 40–44, Morristown, NJ, USA, 1995. Association for Computational Linguistics. ISBN 0805822437. doi: 10.3115/222020.222051. URL <http://portal.acm.org/citation.cfm?doid=222020.222051>.
- A. Catala, F. Garcia-Sanjuan, J. Azorin, J. Jaen, and A. Mocholi. Exploring Direct Communication and Manipulation on Interactive Surfaces to Foster Novelty in a Creative Learning Environment. In *Proceedings of the 6th International Conference on Virtual Learning - ICVL 2011*, pages 373–379, 2011a. URL [http://www.icvl.eu/2011/disc/icvl/documente/pdf/tech/ICVL\\_Technologies\\_paper01.pdf](http://www.icvl.eu/2011/disc/icvl/documente/pdf/tech/ICVL_Technologies_paper01.pdf).
- A. Catala, J. Jaen, A. a. Martinez-Villaronga, and J. a. Mocholi. AGORAS: Exploring creative learning on tangible user interfaces. *Proceedings - International Computer Software and Applications Conference*, pages 326–335, 2011b. ISSN 07303157. doi: 10.1109/COMPSAC.2011.50.

- 
- W. C. Chang, T. H. Wang, F. H. Lin, and H. C. Yang. Game-based learning with ubiquitous technologies. *IEEE Internet Computing*, 13(4):26–33, 2009. ISSN 10897801. doi: 10.1109/MIC.2009.81.
- Cheng-Yu Hung, F.-O. Kuo, J. C.-Y. Sun, and Pao-Ta Yu. An Interactive Game Approach for Improving Students’ Learning Performance in Multi-Touch Game-Based Learning. *IEEE Transactions on Learning Technologies*, 7(1):31–37, Jan. 2014. ISSN 1939-1382. doi: 10.1109/TLT.2013.2294806. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6687273>.
- A. D. Cheok, S. Lee, S. Kodagoda, K. E. Tat, and L. N. Thang. A social and physical inter-generational computer game for the elderly and children: Age invaders. In *Proceedings - International Symposium on Wearable Computers, ISWC*, volume 2005, pages 202–203. IEEE, 2005. ISBN 0769524192. doi: 10.1109/ISWC.2005.6. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1550811>.
- W. Conen and G. Neumann. Prerequisites for collaborative problem solving. *Proceedings of WET ICE '96. IEEE 5th Workshop on Enabling Technologies; Infrastructure for Collaborative Enterprises*, pages 310–315, 1996. ISSN 1080-1383. doi: 10.1109/ENABL.1996.555249. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=555249>.
- N. Deshpande, B. De Vries, and J. P. Van Leeuwen. Building and supporting shared understanding in collaborative problem-solving. In *Proceedings of the International Conference on Information Visualisation*, volume 2005, pages 737–742, 2005. ISBN 0769523978. doi: 10.1109/IV.2005.29.
- A. Donker and P. Reitsma. Young children’s ability to use a computer mouse. *Computers & Education*, 48(4):602–617, May 2007. ISSN 03601315. doi: 10.1016/j.compedu.2005.05.001. URL <http://linkinghub.elsevier.com/retrieve/pii/S0360131505000710>.
- A. G. Fallis. The instructional gaming literature: Implication and 99 sources. *Journal of Chemical Information and Modeling*, 53(9):1689–1699, 2013. ISSN 1098-6596. doi: 10.1017/CBO9781107415324.004.
- S. B. Fan. Roles in Online Collaborative Problem Solving. *Cognition*, pages 265–266, 2010. doi: 10.1109/VLHCC.2010.51.
- F. Garcia-Sanjuan, J. Jaen, and S. Jurdi. Towards Encouraging Communication in Hospitalized Children through Multi-Tablet Activities. In *Proc. of XVII International Conference on Human Computer Interaction (Interacción'16)*, page In press, 2016a.
- F. Garcia-Sanjuan, J. Jaen, and V. Nacher. From Tabletops to Multi-Tablet Environments in Educational Scenarios: A Lightweight and Inexpensive Alternative. In *Proceedings of the 16th International Conference on Advanced Learning Technologies*, number 1, page In press, 2016b. doi: 10.1109/ICALT.2016.69.
-

- 
- F. Garcia-Sanjuan, J. Jaen, and V. Nacher. Tangibot: A tangible-mediated robot to support cognitive games for ageing people—A usability study. *Pervasive and Mobile Computing*, 2016c. ISSN 15741192. doi: 10.1016/j.pmcj.2016.08.007.
- A. a. Gokhale. Collaborative learning enhances critical thinking. *Journal of Technology Education*, 7(1):22–30, 1995. ISSN 1045-1064. URL [http://scholar.lib.vt.edu/ejournals/JTE/v7n1/gokhale.jte-v7n1.html?ref=Sawos.Org%5Cdelimiter"026E30F%5Cnhttps://vtechworks.lib.vt.edu/bitstream/handle/10919/8591/JTEV7N1.pdf?sequence=1#page=23](http://scholar.lib.vt.edu/ejournals/JTE/v7n1/gokhale.jte-v7n1.html?ref=Sawos.Org%5Cdelimiter).
- I. Granic, A. Lobel, C. M. E, and R. C. M. E. Engels. The benefits of playing video games. *American Psychologist*, 69(1):66–78, 2014. ISSN 1935-990X. doi: 10.1037/a0034857. URL <http://www.ncbi.nlm.nih.gov/pubmed/24295515>.
- C. Gutwin and S. Greenberg. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work (CSCW)*, 11(3-4): 411–446, Sept. 2002. ISSN 0925-9724. doi: 10.1023/A:1021271517844. URL <http://link.springer.com/10.1023/A:1021271517844>.
- I. Hatzilygeroudis, F. Grivokostopoulou, and I. Perikos. Using game-based learning in teaching CS algorithms. In *Proceedings of IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) 2012*, pages H2C–9–H2C–12. IEEE, Aug. 2012. ISBN 978-1-4673-2418-2. doi: 10.1109/TALE.2012.6360338. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6360338>.
- D. Hooshyar, R. B. Ahmad, M. Yousefi, M. Fathi, S.-J. Horng, M. Hooshyar, A. R. Dooraki, and Dooraki. Improving web-based problem solving skills of students with a novel game-based Intelligent Tutoring System. In *2015 International Conference on Science in Information Technology (ICSITech)*, pages 63–66. IEEE, Oct. 2015. ISBN 978-1-4799-8384-1. doi: 10.1109/ICSITech.2015.7407778. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7407778>.
- E. Hornecker, P. Marshall, N. S. Dalton, and Y. Rogers. Collaboration and interference. In *Proceedings of the ACM 2008 conference on Computer supported cooperative work - CSCW '08*, page 167, New York, New York, USA, 2008. ACM Press. ISBN 9781605580074. doi: 10.1145/1460563.1460589. URL <http://portal.acm.org/citation.cfm?doid=1460563.1460589>.
- J. Huizinga. Homo Ludens: A Study of the Play-Element in Culture. *European Early Childhood Education Research Journal*, 19(2):1–24, 2007. ISSN 1350-293X. doi: 10.1177/0907568202009004005. URL [http://www.ncbi.nlm.nih.gov/pubmed/17036547%5Cdelimiter"026E30F%5Cnhttp://unesdoc.unesco.org/images/0014/001474/147470e.pdf%5Cdelimiter"026E30F%5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2807809&tool=pmcentrez&](http://www.ncbi.nlm.nih.gov/pubmed/17036547%5Cdelimiter)
-

---

rendertype=abstract&\delimiter"026E30F\$nhhttp://www.  
psych.utoronto.ca/users/peterson/Psy230.

- C.-y. Hung, T.-W. Chang, P.-T. Yu, and P.-J. Cheng. The Problem Solving Skills and Learning Performance in Learning Multi-Touch Interactive Jigsaw Game Using Digital Scaffolds. *2012 IEEE Fourth International Conference On Digital Game And Intelligent Toy Enhanced Learning*, pages 33–38, 2012. doi: 10.1109/DIGITEL.2012.13. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6185577>.
- H. Idrus, H. Mohd Dahan, and N. Abdullah. Integrating critical thinking and problem solving skills in the teaching of technical courses: The narrative of a Malaysian private university. In *2010 2nd International Congress on Engineering Education*, pages 258–263. IEEE, Dec. 2010. ISBN 978-1-4244-7308-3. doi: 10.1109/ICEED.2010.5940803. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5940803>.
- L. Y. Li, C. W. Chang, and G. D. Chen. Researches on using robots in education. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5670 LNCS, pages 479–482. Springer Berlin Heidelberg, 2009. ISBN 3642033636. doi: 10.1007/978-3-642-03364-3\_57. URL [http://link.springer.com/10.1007/978-3-642-03364-3\\_57](http://link.springer.com/10.1007/978-3-642-03364-3_57).
- Y. H. Liao and C.-Y. Shen. Heuristic Evaluation of Digital Game Based Learning: A Case Study. In *2012 IEEE Fourth International Conference On Digital Game And Intelligent Toy Enhanced Learning*, pages 192–196. IEEE, Mar. 2012. ISBN 978-1-4673-0885-4. doi: 10.1109/DIGITEL.2012.54. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6185617>.
- N. Obeid and A. Moubaidin. On the role of dialogue and argumentation in collaborative problem solving. In *ISDA 2009 - 9th International Conference on Intelligent Systems Design and Applications*, pages 1202–1208, 2009. ISBN 9780769538723. doi: 10.1109/ISDA.2009.60.
- Oecd. PISA 2012 Field Trial Problem Solving Framework. *Oecd*, pages 1–47, 2012.
- Oecd. Pisa 2015 Draft Collaborative Problem Solving Framework March 2013. *Oecd*, (March 2013):89, 2015.
- M. Pinho-Lopes and J. MacEdo. Promoting problem solving and high order thinking skills in geotechnical courses. In *Proceedings - 2013 1st International Conference of the Portuguese Society for Engineering Education, CISPEE 2013*, 2013. ISBN 9781479912216. doi: 10.1109/CISPEE.2013.6701987.
- R. Raman, A. Lal, and K. Achuthan. Serious games based approach to cyber security concept learning: Indian context. In *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, pages 1–5. IEEE, Mar. 2014. ISBN 978-1-4799-4982-3. doi: 10.1109/ICGCCEE.

- 
- 2014.6921392. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6921392>.
- J. C. Read. Validating the Fun Toolkit: an instrument for measuring children's opinions of technology. *Cognition, Technology & Work*, 10(2):119–128, Apr. 2008. ISSN 1435-5558. doi: 10.1007/s10111-007-0069-9. URL <http://link.springer.com/10.1007/s10111-007-0069-9>.
- J. C. Read and S. MacFarlane. Using the fun toolkit and other survey methods to gather opinions in child computer interaction. In *Proceeding of the 2006 conference on Interaction design and children - IDC '06*, page 81, New York, New York, USA, 2006. ACM Press. doi: 10.1145/1139073.1139096. URL <http://portal.acm.org/citation.cfm?doid=1139073.1139096>.
- N. Reski, S. Nordmark, and M. Milrad. Exploring New Interaction Mechanisms to Support Information Sharing and Collaboration Using Large Multi-touch Displays in the Context of Digital Storytelling. In *2014 IEEE 14th International Conference on Advanced Learning Technologies*, pages 176–180. IEEE, July 2014. ISBN 978-1-4799-4038-7. doi: 10.1109/ICALT.2014.59. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6901430>.
- J. Rick, P. Marshall, and N. Yuill. Beyond one-size-fits-all: how interactive tabletops support collaborative learning. *Proceedings of the 10th International Conference on Interaction Design and Children*, pages 109–117, 2011. URL <file:///Users/me/Documents/references.sente6lib/Contents/Attachments/Rick/2011/Beyondone-size-fits-allhowinteractivetabletop.pdf>.
- G. Romeo, S. Edwards, S. McNamara, I. Walker, and C. Ziguras. Touching the screen: issues related to the use of touchscreen technology in early childhood education. *British Journal of Educational Technology*, 34(3):329–339, June 2003. ISSN 0007-1013. doi: 10.1111/1467-8535.00330. URL <http://doi.wiley.com/10.1111/1467-8535.00330>.
- P. Sancho, R. Fuentes-Fernández, and B. Fernández-Manjón. NUCLEO: Adaptive Computer Supported Collaborative Learning in a Role Game Based Scenario. In *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, pages 671–675. IEEE, 2008. ISBN 978-0-7695-3167-0. doi: 10.1109/ICALT.2008.147. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4561801>.
- J. R. Savery. Overview of Problem-based Learning : Definitions and Distinctions. *Interdisciplinary Journal of Problem-based Learning*, 1(1):13, 2006. ISSN 15415015. doi: 10.7771/1541-5015.1002. URL <http://dx.doi.org/10.7771/1541-5015.1002>.
- F. Scharf, T. Winkler, C. Hahn, C. Wolters, and M. Herczeg. Tangicons 3.0. In *Proceedings of the 11th International Conference on Interaction Design and Children*
-

- 
- *IDC '12*, page 144, New York, New York, USA, 2012. ACM Press. ISBN 9781450310079. doi: 10.1145/2307096.2307113. URL <http://dl.acm.org/citation.cfm?doid=2307096.2307113>.
- Schlieder, Kiefer, and Matyas. Geogames: Designing Location-Based Games from Classic Board Games. *IEEE Intelligent Systems*, 21(5):40–46, Oct. 2006. ISSN 1541-1672. doi: 10.1109/MIS.2006.93. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1705427>.
- B. Schneider, P. Jermann, G. Zufferey, and P. Dillenbourg. Benefits of a tangible interface for collaborative learning and interaction. *IEEE Transactions on Learning Technologies*, 4(3):222–232, 2011. ISSN 19391382. doi: 10.1109/TLT.2010.36.
- B. Schneider, P. Blikstein, and W. Mackay. Combinatorix. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces - ITS '12*, page 129, New York, New York, USA, 2012. ACM Press. ISBN 9781450312097. doi: 10.1145/2396636.2396656. URL <http://dl.acm.org/citation.cfm?doid=2396636.2396656>.
- M. Schubert, A. Serna, and S. George. Using Collaborative Activities on Tabletops to Enhance Learning and Knowledge Transfer. In *2012 IEEE 12th International Conference on Advanced Learning Technologies*, pages 610–612. IEEE, July 2012. ISBN 978-1-4673-1642-2. doi: 10.1109/ICALT.2012.73. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6268190>.
- J. Shang, M. S. Y. Jong, F. L. Lee, and J. H. M. Lee. VISOLE: A new game-based situated learning paradigm. In *Proceedings - Sixth International Conference on Advanced Learning Technologies, ICALT 2006*, volume 2006, pages 1082–1083. IEEE, 2006. ISBN 0769526322; 9780769526324. doi: 10.1109/ICALT.2006.1652635. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1652635><http://www.scopus.com/inward/record.url?eid=2-s2.0-34247139267&partnerID=40&md5=7bd74d31db1c653e0aad6e0f1cc55447>.
- A. Shatte and J. Holdsworth. Hand-Held Mobile Augmented Reality for Collaborative Problem Solving: A Case Study with Sorting. *2014 47th Hawaii International Conference on System Sciences*, pages 91–99, 2014. doi: 10.1109/HICSS.2014.20. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6758615>.
- A. Siang and Radha Krishna Rao. Theories of learning: a computer game perspective. In *Fifth International Symposium on Multimedia Software Engineering, 2003. Proceedings.*, pages 239–245. IEEE, 2003. ISBN 0-7695-2031-6. doi: 10.1109/MMSE.2003.1254447. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1254447>.
- A. Strawhacker, A. Strawhacker, and M. U. Bers. "I Want My Robot to Look for Food": Comparing Kindergartner's Programming... *International Journal of Technology and Design Education*, 25(3):293–319, 2015.
-



- 
- L. S. Vygotsky. *Mind in society: The development of higher psychological processes*. page 159, 1978. ISSN 9780674576292. doi: 10.1007/978-3-540-92784-6. URL <http://www.amazon.com/dp/0674576292>.
- P. Wouters, C. van Nimwegen, H. van Oostendorp, and E. D. van der Spek. A meta-analysis of the cognitive and motivational effects of serious games. *Journal of Educational Psychology*, 105(2):249–265, 2013. ISSN 1939-2176. doi: 10.1037/a0031311. URL <http://doi.apa.org/getdoi.cfm?doi=10.1037/a0031311>.
- L. Xie, A. N. Antle, and N. Motamedi. Are tangibles more fun? In *Proceedings of the 2nd international conference on Tangible and embedded interaction - TEI '08*, page 191, New York, New York, USA, 2008. ACM Press. ISBN 9781605580043. doi: 10.1145/1347390.1347433. URL <http://portal.acm.org/citation.cfm?doid=1347390.1347433>.