An Adventure Serious Game for Teaching Effort Estimation in Software Engineering

Beatriz Marín[1][0000-0001-8025-0023], Matías Vera[1] and Giovanni Giachetti[2]

[1] Universidad Diego Portales, Facultad de Ingeniería y Ciencias, Santiago, Chile
beatriz.marin@mail.udp.cl, matias.vera@mail.udp.cl
[2] Universidad Tecnológica de Chile INACAP, Santiago, Chile
ggiachetti@inacap.cl

Abstract. Imagine that it is possible to learn effort estimation concepts and its application in an attractive manner, where complex and technical knowledge are presented through a playful experience. The serious game Back to Penelope face this challenge, in which the student is trying to save the life of a young girl astronaut from a desolated world. To do this, the player must correctly estimate the development effort for the vital systems of the spaceship of the astronaut. The game is based on the COSMIC measuring method for estimation. This paper presents the results obtained from a systematic review performed to identify the main features of the existent serious games related to estimation. Moreover, this paper presents the design of the game developed, which has elements to improve the motivation of the students that use Back to Penelope. Finally, we present the promising results obtained from the validation of the game when teaching software engineering courses.

Keywords: Serious Game, COSMIC, estimation, learning.

1 Introduction

Effort estimation is one of the key factors involved in the failure of software development projects according to the *Chaos Report* [1]. At the same time, proper planning is the fourth most important factor for the success of a software project. The works presented in [2] and [3] indicate that an incorrect planning can be produced by 1) lack of expertise in the planning process; 2) unrealistic assumptions and expectations; 3) lack of a systematic measurement process that has adequate methods and tools for the project involved.

In this context, effort estimation capability is one of the main aspects that must be properly learned by software engineers. Despite the relevance of performing appropriate effort estimations for software engineers, the process of teaching these kinds of complex technical concepts is always a hard task. This teaching process is especially difficult since comprehension of concepts and the ability to apply these concepts to practical problems are both needed. Furthermore, traditional teaching classes have demonstrated to be effective to introduce measurement estimation concepts; however, they are not effective in motivating students [4] or allowing them to put the knowledge learned into practice [5]. Teaching classes based on serious games can be a good alternative to capture students' attention and to improve their motivation in the application of more complex/technical software engineering concepts [6]. Thus, a serious game provides a risk-free environment that allows students to experiment and have re-learning experiences.

In this paper, we present a serious game developed to teach effort estimation using the COSMIC Function Points measuring method [7] applied to conceptual models. The game is oriented to courses related to software engineering and software project management. The game, *Back to Penelope*, is a supporting tool for the teacher (not a replacement), which is oriented to improve the learning skills related to the application of the estimation technique. Back to Penelope has been successfully applied to software engineering courses in a preliminary evaluation. Thus, the contribution of this paper is twofold: 1) to present the structure and design of an adventure serious game for teaching estimation, and 2) to show the results from the initial evaluation of this game with software engineering students.

The rest of the paper is organized as follows: Section 2 shows the related work to serious games and teaching effort estimation skills in software projects; Section 3 presents the design of the game; Section 4 briefly explains the validation of the game; and finally, Section 5 summarizes our conclusions and future work.

2 Related work and Background

The use of serious games to improve learning has been widely studied as we can observe in the references [8-10], which in summary show that students' behavior influences their learning [11]. In order to find previous works related to serious games for teaching effort or time estimation, we conducted a systematic mapping review process by following Kitchenham guidelines [12]. Thus, the review protocol considers: the formulation of the research question, the definition of the search strategy, the definition of inclusion and exclusion criteria, and the data extraction strategy.

2.1. Research question and Search Strategy

The research question that drives this review is the following:
What evidence about serious games in the field of teaching/learning time or effort estimation exists?

From this question, the following search string was defined using the main words of the research question and the AND and OR operators: *Serious Games* AND *Learning* AND *techniques* AND (*effort estimation* OR *time estimation*).

The search string was used in the digital libraries a) IEEE Xplore, b) ACM Digital Library, and c) Springer Link. We use these digital libraries since they store the majority of computer science research.

2.2. Inclusion and exclusion criteria, and Data extraction strategy

The definition of the search process first considered to execute the search string in the selected digital libraries. Then, we read the title and the abstract in order to identify the candidate studies. If the paper did not fulfill the inclusion criteria, or it fulfilled one of the exclusion criteria, then the paper was discarded. If the paper fulfilled the inclusion criteria, it was selected as a candidate study. For the selection of candidate studies, we defined inclusion and exclusion criteria.

Inclusion criteria:

- The paper must be a conference or journal manuscript written in English or Spanish.
- The paper is about serious games oriented to effort or time estimation.

Exclusion criteria:

- The paper is related to effort or time estimation for development of serious games.
- The paper is related to effort or time estimation only, without a serious game that subjects can use for learning.
- The paper is related to serious game design or implementation only, without presenting an approach for learning effort or time estimation.
- Grey literature (blogs, letters, book prologues, posters), books, master or doctoral thesis)
- Duplicate of papers among the target libraries.

After the application of the inclusion and exclusion criteria, the candidate studies were completely read in order to select them. For the data extraction strategy, we used a tabular form to store all the information extracted from the selected papers in order to easily compare all the approaches. To do this, we use the following criteria:

- **Game Name.** As indicated in the reference paper.
- **Game Type.** The type is directly related to the interaction with the player, for instance, this can be a simulation game, adventure game, role game, multiplayer, etc.
- **Learning Scope.** This is indicated when the serious game analyzed is oriented to teaching other aspects than effort estimation.
- **Game Design.** It refers to the game architecture, and it is mainly related to functional features.
- **Validated (Val).** Indicates with Yes or No, when the game has been validated. It considers type of validation, number of users, measures defined, and the validation process.
- **Effort Estimation (EE).** Indicates with Yes or No, when the game analyzed considered effort estimation activities to be done by the players (students).

2.3. Systematic Review Execution and Results

Table 1 summarizes the results obtained after the first execution of the research question in the different libraries. After the first iteration, a snowballing search was applied to increase the initial number of candidate studies obtained since it was reduced. The snowballing technique considers reviewing the references of each candidate study obtained from the execution of the search string in order to obtain more candidate studies.

Table 1. Results obtained from the execution of the search string.

Library Name	Search Date	Search result	Candidate Studies	Selected Papers
IEEEXplore	21-01-18	1152	5	2
ACM Digital Library	21-01-18	217	3	1
Springer Link	21-01-18	123	3	2
(Snowballing)	09-03-18	7	5	5

Table 2 shows the final articles selected and the data extracted according to the strategy defined above.

To answer our research question (*What evidence about serious games in the field of teaching/learning time or effort estimation exists?*), we found evidence of serious games related to the field of teaching effort or time estimation (see Table 2).

From the articles selected, 70% are related to software project management, and 30% are oriented to effort estimation. In detail, 20% consider an effort estimation process performed by students. The work proposed in [13] that is related to the *ProDec* approach, applies an effort estimation approach based on Albretch function Point. This is a simulation game that considers the ISO 21500 [14]. This work has been validated with students to estimate the effort of a set of pre-defined tasks.

The approach of *The Incredible Manager* presented in [15] considers effort estimation from the developer side; however, this approach does not use a specific method for effort estimation.

The other approaches analyzed do not consider effort estimation tasks, or indeed the effort is already predefined by the system without intervention of the player (student) in the estimation process.

Most of the games are related to general tasks of software engineering and project management. Only two games are oriented to more specific domains. In the case of [16], the domain is related to software process improvement, and the approach presented in [17] is related to the requirement engineering domain.

All the approaches analyzed are related to simulation games, where the players are involved in explicit software engineering challenges or planning tasks of project management. In some games, the players can take specific roles as project leader or developer such as in papers [16] and [18]. However, none of these approaches present novel gaming scenarios or a story thread that differs from software engineering tasks, which could better motivate the use of the game as a complementary learning technique. These strategies have shown that serious games benefit the learning process [5].

Later, some background about the Cosmic Function point approach and the MDA Framework for game design is provided to facilitate the comprehension of the approach presented.

Table 2. Selected papers with the data-extraction results.

Ref	Paper Title	Game Name	Game Type	Learning Scope	Game Design	Val	EE
[5]	Coverage of the ISO 21500 Standard in the Context of Software Project Management by a Simulation-Based Serious Game	*ProDec*	Simulation	Software Project Management	-	No	yes
[7]	Coverage of ISO/IEC 29110 Project Management Process of Basic Profile by a Serious Game	*ProDec*	Simulation	Software Project Management	-	No	yes
[6]	Integrating serious games as learning resources in a software project management course: the case of *ProDec*	*ProDec*	Simulation	Software Project Management	-	yes	No
[2]	Simulation in software engineering training	*SESAM*	Simulation	Software Engineering	Models	yes	yes
[10]	Challenges and issues in the development of a Software Engineering simulation game	*SPIAL*	Simulation, Role playing	Software Process Improvement	Models	No	No
[26]	Engendering an Empathy for Software Engineering	*SimjavaSP*	Simulation, Role Playing	Software Project Management	Models	yes	No
[23]	*AMEISE* – A Media Education Initiative for Software Engineering Concepts, the Environment and Initial Experiences	*AMEISE*	Simulation, Multiplayer, Competition	Software Project Management	Models	No	No
[21]	Model Driven Game Development: Experience and Model Enhancements in Software Project Management Education	*The Incredible Manager*	Simulation, Role Playing	Software Project Management	Models	No	No
[1]	A Simulation-Based Game for Project Management Experiential Learning	*The Incredible Manager*	Simulation, Role Playing	Software Project Management	Models	No	No
[12]	A Game for Taking Requirements Engineering More Seriously	*Software Quantum Game*	Simulation	Requirement Engineering	Models	No	No

2.4. The MDA Framework for Serious Game Development

The MDA framework [19], whose acronym comes from mechanics, dynamics, and aesthetics, involve not only software but also player interaction (see Fig. 1) . MDA framework is focused on bridging the gap between design and game development (programming). To do this, this framework divides the game architecture into three main components: mechanics, which correspond to the rules; dynamics, which correspond to the system acting in real time; and aesthetic, which corresponds to what makes a videogame attractive or fun, as well as the emotional response expected from the player.

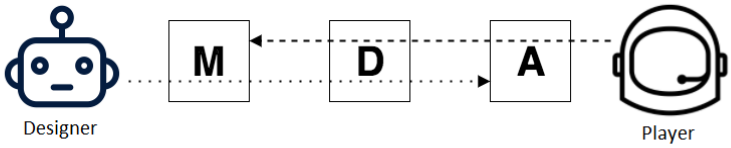

Fig. 1. MDA Schema for videogames development

Coinsidering the three MDA components for the development of a serious game, all the teaching techniques and learning skills are involved at the mechanics level. Hunicke in [19] states that each MDA component is a subtlety engaged isolated game-view. This means that in the context of this work, where teaching effort estimation concepts and their application are its focus, the *Dynamics* and *Aesthetic* components do not need to be related to software engineering simulation. Therefore, it is possible to present some fantasy elements to make a more attractive game, that is aligned with the mechanics components that provide the requiered knowledge to be transmitted to the student at the same time.

2.4. The Cosmic Function Point Method

The COSMIC Full Function Point (FFP) estimation method [20] is oriented to obtaining the functional size of an application by means of the identification of functional processes and the data movements involved. For its execution, this method considers the following phases: 1) a strategy phase in which the purpose and scope of the measurement task is defined; 2) a mapping phase in which the functional processes contained in the scope of the estimation are identified; and 3) an estimation phase in which all the data movements for each functional process are identified. Each data movement corresponds to 1 CFP (COSMIC Function Point), the sum of all the function points of a process corresponds to the functional size of the corresponding process. The aggregated sum of the functional size of all processes will correspond to the functional size of the piece of software measured [7].

Since the functional size of the piece of software is given in an abstract unit (CFP), for effort estimation purposes, it is necessary to translate this measure to a human effort-related value. For this purpose, the data collected by the International Software Benchmarking Standards (ISBSG) Group is used [21]. The ISBSG database collects data from software projects from different domains, and we use this data to estimate the size of the software products and the effort required to complete the software development project.

3 Back to Penelope - Serious Game Design

Back to Penelope is a videogame that was developed with the Unity game development engine, which uses C# as scripting language. Unity provides free packages to compile videogames for different platforms, such as Windows, OSx, Linux, iOS, or Android operating systems. In particular, *Back to Penelope* was developed for desktop platforms, and it is available for Windows, Mac, and Linux operating systems. Adobe's Photoshop was largely used for asset creation, and Steinberg's Cubase DAW was used for musicalization. The game is available at http://backtopenelope.me

Development of the first iteration of the game took between 18 and 20 weeks, and approximately 220 hours of programming. It is important to consider that character and graphic design was not included in these programming hours.

For the design of the serious game, we use fantasy elements in the dynamics and aesthetic components in order to provide a more attractive way to practice the measurement concepts, and therefore the effort estimation from conceptual models. Thus, in *Back to Penelope*, the player must take the role of *Ada*, a young girl astronaut on an exploration mission of a new solar system as part of the *Odyssey* project, whose mission is to find new planets. Ada travels in a recognition spaceship that is designed to travel large distances among planets; however, it is not designed to land on or launch from a planet itself since it does not have enough power to break the eventual planet gravity.

As part of her mission, Ada is orbiting the planet Omicron-IV, when suddenly, her ship is trapped by the planetary gravity causing Ada to force land on Omicron-IV. After the crash of her personal spaceship, she is trapped by the high gravity of this mysterious planet. This situation forces Ada to fix her ship in order to return to her mothership, which is named *Penelope*. To fix the ship, she must use the *disposable Utilities Droids* (dUD), which require the indication of the functional size of each task that needs to be accomplished.

We created the *Back to Penelope* videogame to help in the learning process of effort estimation through the application of the COSMIC measurement method. To do that, in each of the *Back to Penelope* scenes, the player has challenges that correspond to tasks to be performed, which are represented by one or more class diagrams. In order to perform these tasks, the player must assign a dUD to each of the classes presented in the diagrams, indicating the required CFP size of the selected class. This is measured by applying the COSMIC method. After the player has done this, the selected dUD will be in a working state, in which it will remain for the number of seconds that the player calculated, taking into account the amount of CFP estimated for the class assigned to the dUD. This conversion from CFP to time is done considering the following constant: 1 CFP corresponds to 3 seconds of implementation time for a dUD (namely 1 CFP = 3 dud-seconds). We use a ratio from the information of ISBSG regarding the time required to develop a project with the functional size of the project. Once the player has measured the size of the piece of software, it is possible to calculate the effort estimation using the data provided by ISBSG.

During the tutorial section of the Back to Penelope game, the player is guided through an estimation process for a simple class diagram. This tutorial shows the user a guide for the interaction of the player with the game, and it also shows details about how the estimation process works: i.e., it shows a simple definition of the COSMIC measurement method considering its three phases, how to identify data movements of the pieces of software, and how to obtain the size in COSMIC Function Points.

3.1. Back to Penelope Mechanics

In *Back to Penelope*, each scene presents to the player a set of challenges in terms of support systems that must be implemented for fixing the spaceship. These support systems are represented by class models. In order to complete a task, the player must assign a dUD to each class of each system (or sub-system) so it can fix its part of the spaceship. This is done by estimating the corresponding size for the corresponding class to be implemented. The size estimation is performed by applying the COSMIC method.

The difficulty related to estimation increases from one scene to another by adding new features such as different kinds of associations, more complex class methods, etc.

The first scene of the game provides a tutorial where the player can learn how to perform a proper effort estimation by using the COSMIC measurement method. After this, during each scene execution, the player can access a help view, which shows details about the game context as well as the application of the COSMIC method. At the end of each estimation process of the work that must be performed by each dUD, the game provides feedback to the player about the estimation performed, indicating if it is right or wrong. In case that the estimation made by the player is lower than what the task requires, the player has the possibility of correcting the initial estimation performed. Nevertheless, in case that player makes a higher estimation in regard to the task time, the player must wait for the dUD to finish the task in the time originally estimated.

3.2. Back to Penelope Dynamics

The dynamics are those user-interaction components that bring the game closer to the player. In *Back to Penelope*, the main menu of the game initiates this interaction (see Fig. 2). The scene selection screen shows the available scenes and those that are locked until the player achieves a certain level of progress (see Fig 3). During the game execution, each scene is divided into three stages, which are related to a section of the class diagram that represents the system to be developed through the scene (see Fig. 3).

The player can assign one of the dUDs to the implementation of each class of the systems of the Ada spaceship (Fig. 2). Once a dUD is assigned, the player estimates the functional size of the class, which is translated into the effort that is necessary to implement the class. The estimation performed by the player is compared to the estimation already calculated for each class to determinate if the player has over- or underestimated the functional size of the class, with a fault tolerance of +3 to -3

Cosmic Function Points. The fault tolerance level can be increased or reduced in order to change the difficulty level of the game by the teacher.

Fig. 2. Main Menu and Game Play view of Back to Penelope.

Fig. 3. Scene selection and class model view related to a scene of the game.

3.3. Back to Penelope Aesthetics

The main aesthetic aspect used in this game is the narrative. We create a fantasy story, which could motivate engineering students to play rather than merely simulate office tasks. The aesthetics of the game include fantasy aspects that are directly related to the story represented in each scene. For instance, the player is Ada, a time bar for each scene, different difficulty levels in the class diagram of each scene, feedback that is given to the player when the implementation of a class has been under- or overestimated, among others. The scenes of the story, explained below, are called *Crash Landing*, *Motion*, *Hard Cut*, and *Escape Velocity*.

Scene 1. Crash Landing: As a consequence of the collision, Ada's cargo cabin door gives way, causing materials and equipment to fall on the Omicron-IV’s surface, which seems to be a deserted planet, without traces of life. When Ada wakes up after the collision, the her suit sensors indicate that the atmosphere is not breathable and that her oxygen reserves are at 30% of its capacity. The scene starts at the moment that one of the dUDs starts to operate, apparently by error. During the three stages of this scene, Ada must to implement the control panel to operate the dUDs, to mount the vital support dome, and to configure a telemetric system to locate where *Penelope*, the mothership, is located.

Moreover, in the first scene there is also an initial tutorial stage that is used by students to understand how to interact with the dUDs, and it is also used to explain how effort estimation can be performed by using an example class diagram.

Scene 2. Motion. At the sunset, Ada sees a metallic glow on the horizon, just over one of the rock formations that surrounds the place where she landed. Using the long-distance function of her helmet's viewfinder, she sees a small group of humanoids, native aliens probably. They look muscular and are poorly dressed in rags. Their skin resembles the rocks of Omicron-IV. They carry primitive weapons in their hands, like spears. At the same moment, the leader of the group, who carries the largest spear, throws its weapon, hitting the side of the Ada's spaceship. Ada must use her dUDs to quickly implement a defense system, with activity sensors, shooting towers, and a centralized defense control to get through the night safely in the ship.

Scene 3. Hard Cut. The next morning, Ada notices that the aliens were smart enough not to approach the spaceship during the night. She only has two days to launch her spaceship and escape from the planet's gravity; otherwise, she would be too far away to reach *Penelope*. To achieve the necessary escape velocity, Ada uses a trick that she learned in the Astronaut Academy; she constructs a small nuclear reactor with one of the sources of power of her ship. The main problem with this solution is the radiation emitted by the nuclear reactor. Hence, she must be able to develop a mechanism to deactivate the reactor once the necessary speed to escape from the planet's gravity is achieved, and to use an alternative propulsion system for the rest of the journey.

Scene 4. Escape Velocity. After the launch, Ada observes that the metallic cover of the ship is cracking due to the damage caused by Omicron-IV's. The only alternative to impede the ship's collapse is to redirect part of the ship's energy to create an energy shield at the top of the ship. Moreover, while the escape speed is approaching, the radiation is also dangerously increasing. Thus, Ada must develop and activate an alternative ionic propulsion system, while she shuts down the nuclear reactor before it is too late.

As example of the game aesthetics, Fig. 4 shows the first scenario and the class model involved in the three stages of this scenario.

It is important to mention that each scene of the game has a time limit. This is of importance because the dUDs need different amounts of time to develop their tasks, depending on the effort involved. During the game, the time element is represented in different manners, depending on the scene involved. For instance, in Scene 1, time is related to the oxygen available; in Scene 4, it is related to the time needed to achieve the escape speed without being affected by the nuclear reactor radiation.

Other aesthetic elements that are important to mention are the penalties that the player obtains when he or she incorrectly estimates the time needed for a task to be done by a dUD. In fact, the player can unblock more scenes as a reward for making good estimations, but the player also receives penalties when the estimation is wrong. Thus, when the player estimates more effort than the necessary for performing a specific task, the dUD will stay in a stand-by state until the difference between the time proposed and the real time required by the task is achieved.

Fig. 4. First game scenario and class models involved.

4 Validation of the Game

The validation of *Back to Penelope* considered two aspects: usability and effectiveness. We understand the usability as the perceived ease of use of the game, and the effectiveness is related to the estimation made by the student with regard to the correct estimation. The validation process was performed on two occasions with students that have previously taken software development and software engineering courses.

The usability aspect has been validated by means of a UMUX survey (*Usability Metric for User Experience*) [22], which use a 5-point Likert scale to be answered (values go from 1: totally disagree to 5: totally agree). The UMUX survey presents 4 questions to evaluate the perception of usability of a system, which applied to *Back to Penelope* results in regard to the following four questions:

1) **Perceived Effectiveness:** Does *Back to Penelope* allow you to learn how to estimate by using the COSMIC method?
2) **Perceived Satisfaction:** Is playing *Back to Penelope* a frustrating experience?
3) **General vision:** Is *Back to Penelope* easy to use?
4) **Perceived Efficiency:** Do you spend too much time making corrections with *Back to Penelope*?

On the other hand, the validation of the effectiveness of *Back to Penelope* has been evaluated by means of a metric that considers the number of tries that are necessary by a player to perform a correct estimation. This measure is obtained from the results of the following two questions:

1) How many tries (on average) did it take to obtain a correct estimation of the functional size of a class?
2) What was the highest number of tries you had to do in order to estimate the functional size of a class?

Both questions have as possible answer 1, 2, 3, 4, 5, or more tries. These metrics allows to validate the effectiveness of the game as well as the perceived difficulty of the game.

The number of times that a player correctly answers a game challenge and the time used by each player session are registered once the player stops playing.

4.1 Results obtained from the first validation experience

In the first empirical trial, 10 subjects participated, which corresponds to graduate students from the Engineering major of Information Technology and Telecommunications. This is a five-year course. During this trial, only the UMUX survey was applied since the effectiveness measures were not implemented yet. Fig. 5 shows the results obtained from the survey application.

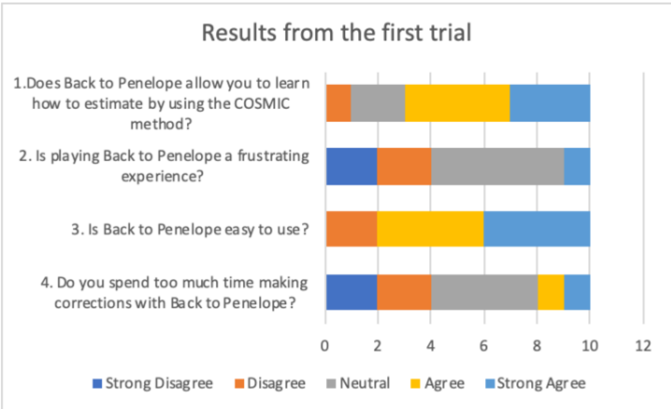

Fig. 5. Results of the survey in the first validation trial.

According to the results presented in Fig. 5, 70% of the players is agreed or totally agreed with the fact that Back to Penelope allowed them to learn how to the estimate by using the COSMIC method. Only 10% did not agree with this affirmation.

For the second question, only 10% of the students agreed that using *Back to Penelope* was a frustrating experience; that is, 90% of the players showed a tendency to not feel frustrated when using the game.

Regarding the question of whether *Back to Penelope* was easy to use, 20% disagreed, and 80% of the players agreed or totally agreed.

Finally, regarding the question about if they spent too much time making corrections with *Back to Penelope*, 80% is disagreed.

These results are positive and suggest that this method for learning estimation using COSMIC is a motivating experience for students.

4.2 Results obtained from the second validation experience

The second validation trial was carried out with 4th year Engineering students majoring in Information Technology and Telecommunications (a five-year course).

This trial had 30 subjects, which were divided into two groups. It is important to mention that this activity was not associated with a grade. The link to download the game *Back to Penelope* http://backtopenelope.me was given to the students in a computer lab, along with a brief 5-minute introduction.

For these students, *Back to Penelope* is the first time that they have to use the COSMIC method for effort estimation.

Furthermore, only five players agreed to register the time that they spent on a scene. Thus, only 27% of the players have a register, and we cannot use it in the analysis about playing time. Despite this, the time spent for each group to complete the different scenes was registered (see Table 3). As we can observe, passing from one scene to another takes a few minutes for the player, so that it is not boring for the student to wait until the dUD finishes the implementation of classes of the systems of each scene.

Table 3 Playing time by scene (in seconds).

Group	Scene 1	Scene 2	Scene 3	Scene 4
Group 1	85.8	142.2	136.8	197.0
Group 2	98.2	179.1		
Average	92.0	160.7	136.8	197.0

Table 4. Size Estimation measured in CFP.

Group	Scene 1	Scene 2	Scene 3	Scene 4
Group 1	39.0	92.0	94.0	120.0
Group 2	31.5	96.7		
Average	35.3	94.3	94.0	120.0
Real Value	34.0	91.0	97.0	102.0

With the data obtained from the second trial, it is observed that student estimations were 17.6% higher than the real value, with an average of 9.5% over the real value. This shows that the students estimated the functional size of the class diagrams with adequate accuracy even though this is the first time that they used the COSMIC method. This provides preliminary evidence of the effectiveness of using serious games for teaching effort estimation.

Fig. 6 shows the results obtained in the questions related to perceived effectiveness. As we can observe, 70% of the students answered that they use the maximum number of tries needed to perform a correct estimation. For instance, Student1 tried 3 times to correctly estimate a class, and this student also thought that the maximum attempts to obtain the correct measurement was 3. Twenty percent of students responded that they perceived that they were more effective in the measurement since their number of tries was lower than expected. Only 10% of students perceived that they were ineffective in the measurement exercise.

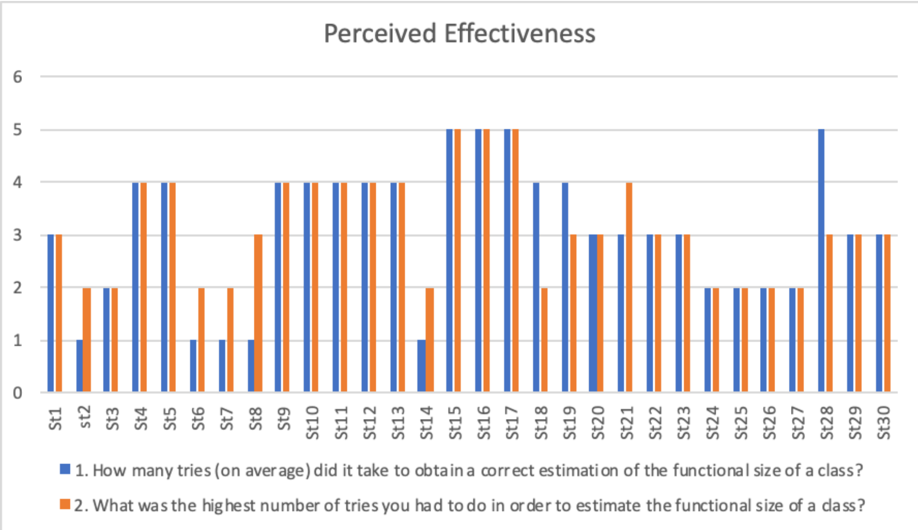

Fig. 6. Results of the perceived effectiveness in the second validation trial.

From the answers of the UMUX survey (see Fig. 7), it is possible to observe that players agreed that *Back To Penelope* allows them to learn to estimate using COSMIC. However, questions 2, 3, and 4 of the questionnaire do not indicate a positive or negative trend. This can be explained because it is the first time that students used a method for effort estimation, and this may cause nervousness at the time of playing. Also, there are some other factors related to the interaction with the game; for instance, the game is designed to be used with a mouse and keyboard, and the students indicated that they would prefer a touch interaction.

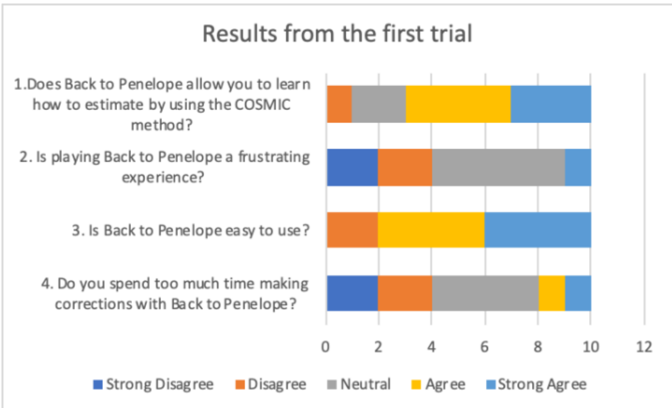

Fig. 7. Results of the survey in the second validation trial.

In relation to the effectiveness measures, the average number of tries to obtain a correct estimation is close to the maximum. This means that the players were required to correct their estimation at least once. This is also related to the fact that this is the first time that this group of students used the COSMIC method.

Despite the promising results obtained during the validation trials, we consider that replication is needed to corroborate and strengthen the results. To do that, we plan to conduct an experiment in order to compare traditional learning with gamified learning of the COSMIC method, applying both learning strategies in realistic settings. In this future experiment we plan to collect qualitative and quantitative data to measure the effectiveness of using a serious game for learning COSMIC method.

5 Conclusions and Future Work

Effort estimation is of paramount importance for properly planning the activities that must be performed in the development of software projects. In order to teach how to estimate the effort needed in development activities, theoretical lectures of the estimation methods are traditionally used in classroom. However, taking into account the lack of concentration of students of the XXI century, new techniques are needed to motivate students to focus on the content presented in the classroom. Thus, in this paper we present a serious game, which has an interesting story to attract students to practice the COSMIC measurement method.

The serious game, *Back to Penelope*, has been carefully designed in order to take into account the developer's viewpoint as well as the player's viewpoint. With this game, we provide a new way to practice how to measure the function points using class diagrams. We validated this game obtaining promising results. Future work includes engaging in new empirical studies to evaluate the F-measure, precision and recall of the measurements using the game *Back to Penelope*. Future work should consider expanding this game to use other measurement methods and estimation approaches that are commonly used in software development projects.

Acknowledgment. This work was funded by CONICYT project ENSE REDI170020, 2017-2019.

References

1. Standish Group: CHAOS Report.(1994). Available at URL http://www.standishgroup.com/sample_research/chaos_1994_1. php, (1994)

2. McGarry, J.: Practical software measurement: objective information for decision makers. Addison-Wesley Professional (2002)

3. Peixoto, C.E.L., Audy, J.L.N., Prikladnicki, R.: The importance of the use of an estimation process. Proceedings of the 2010 ICSE Workshop on Software Development Governance, pp. 13-17. ACM (2010)

4. Shaw, K., Dermoudy, J.: Engendering an empathy for software engineering. Proceedings of the 7th Australasian conference on Computing education-Volume 42, pp. 135-144. Australian Computer Society, Inc. (2005)

5. Marín, B., Frez, J., Cruz-Lemus, J., Genero, M.: An Empirical Investigation on the Benefits of Gamification in Programming Courses. ACM Transactions on Computing Education (TOCE) 19, 1 (2019)
6. Kenny, R.F., Gunter, G.A.: Endogenous fantasy-based serious games: Intrinsic motivation and learning. International Journal of Social Sciences 2, 8-13 (2007)
7. Marín, B., Pastor, O., Abran, A.: Towards an accurate functional size measurement procedure for conceptual models in an MDA environment. Data & Knowledge Engineering 69, 472-490 (2010)
8. Caponetto, I., Earp, J., Ott, M.: Gamification and education: A literature review. In: Proceedings of the 8th European Conference on Games Based Learning (pp. 50–57). ISBN 978-1-910309-55-1, (2014)
9. de Sousa Borges, S., Durelli, V.H., Reis, H.M., Isotani, S.: A systematic mapping on gamification applied to education. In: Proceedings of the 29th annual ACM symposium on applied computing (SAC), pp. 216-222. ACM, (2014)
10. Dicheva, D., Dichev, C., Agre, G., Angelova, G.: Gamification in education: A systematic mapping study. Educational Technology & Society 18, 75-88 (2015)
11. Landers, R.N.: Developing a theory of gamified learning: Linking serious games and gamification of learning. Simulation & Gaming 45, 752-768 (2014)
12. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. (2007)
13. Drappa, A., Ludewig, J.: Simulation in software engineering training. In: Proceedings of the 22nd international conference on Software engineering, pp. 199-208. ACM (2000)
14. Calderón, A., Ruiz, M., O'Connor, R.V.: Coverage of the ISO 21500 standard in the context of software project management by a simulation-based serious game. In: Proceedings of the International Conference on Software Process Improvement and Capability Determination, pp. 399-412. Springer (2017)
15. Dantas, A.R., de Oliveira Barros, M., Werner, C.M.L.: A Simulation-Based Game for Project Management Experiential Learning. In: Proceedings of the Sixteenth International Conference on Software Engineering & Knowledge Engineerin (SEKE), vol. 19, pp. 24 (2004)
16. Peixoto, D.C., Resende, R.F., Pádua, C.I.P.: Evaluating software engineering simulation games: The UGALCO framework. In: Proceedings of the 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, pp. 1-9. IEEE (2014)
17. Knauss, E., Schneider, K., Stapel, K.: A game for taking requirements engineering more seriously. In: Proceedings of the Third International Workshop on Multimedia and Enjoyable Requirements Engineering-Beyond Mere Descriptions and with More Fun and Games, pp. 22-26. IEEE (2008)
18. Barros, M.d.O., Dantas, A.R., Veronese, G.O., Werner, C.M.: Model-driven game development: experience and model enhancements in software project management education. Software Process: Improvement and Practice 11, 411-421 (2006)
19. Hunicke, R., LeBlanc, M., Zubek, R.: MDA: A formal approach to game design and game research. In: Proceedings of the AAAI Workshop on Challenges in Game AI, vol. 4, pp. 1722 (2004)
20. COSMIC-Group: The COSMIC Functional Size Measurement Method – Measurement Manual, version 4.0.2, December 2017, 115 pages. (2017)
21. isbsg.org: ISBSG (International Software Benchmarking Standards Group). (2019)
22. Finstad, K.: The usability metric for user experience. Interacting with Computers 22, 323-327 (2010)