Exploring Game Elements in Learning Programming: An Empirical Evaluation

Adriano Lages dos Santos[1], Maurício R. de A. Souza[1], Marcella Dayrell[2], Eduardo Figueiredo[1]
[1] Department of Computer Science, Federal University of Minas Gerais – Belo Horizonte, Brazil.
[2] Department of Computer Science, State University of Montes Claros – Montes Claros, Brazil.
{adrianolages, mrasouza, figueiredo}@dcc.ufmg.br , mdayrell@unimontes.br

***Abstract*— The worldwide demand for software developers are increasing, however, students are facing problems to learn programming at universities. To make things worse, the failure and dropout rates are high, especially in introductory computer programming courses. To address this type of problem, new strategies have been proposed to engage students in programming courses. One strategy is to use game elements to learn programming. Game elements are important for the success or failure of an educational serious game. In the same way, the students' learning process may benefit from proper use of game elements. In this work, we aim to identify and evaluate which game elements contribute to the students' learning in programming education. We performed 19 user studies to investigate the impact on learning of game elements present in two serious games for learning programming. The study was carried out with students from introductory periods of the undergraduate program in Information Systems. Our results identify some game elements that have positive effect on learning programming. The students' feedback also indicates that game elements help them staying focused, engaged and that games are useful complementary resource for learning process.**

Keywords—Game Elements, Programming, Serious Games.

I. INTRODUCTION

The worldwide demand for software developers is increasing, however, students are facing problems to learn programming at universities [1]. To make things worse, the failure and dropout rates are high, mostly in introductory computer programming courses [2]. One reason is the high amount of training required to learn programming and the need for high students' engagement on additional classroom exercises [3,4]. However, frequently students lack of the required motivation to engage in these tasks.

Game elements have been used as a tool to aid educators in the learning process. Game elements are a set of components that compose a game [5]. In other studies, game elements are also called game attributes [7]. Techniques, such as gamification, use game elements in non-game contexts [6]. Some studies have been investigating the relationship among the use of game elements in the learning process and the learning outcomes [7]. In Electrical and Electronic Engineering and Information Technology (IT) related courses, for instance, educators are using serious games and gamification to motivate and to attract the students' attention for several disciplines, especially for disciplines related to learning programming [8].

In programming education, several serious games have been proposed as complementary tools to the traditional lectures [20]. Serious games are composed by several elements [5], such as leaderboards, levels, badges, bosses and so on. However, there is the need to know if these elements have influence on learning, and which ones are more relevant.

Our recent studies show that serious games for learning programming are only indirectly evaluated based on the game as a whole, without considering its specific elements [9] However, there are studies in literature in other areas of knowledge that show that some elements of games contribute in different ways to learning [7] [20].

The goal of this study is to identify and investigate the impact of game elements on the process of learning programming. We evaluate the elements in two serious games for learning programming: Code Combat[1] and Code Hunt[2]. The games were played by 19 undergraduate students enrolled in the third and fourth academic semesters in an undergraduate program in Information Systems.

The evaluation is performed in two steps. First, the students played the serious games according to the educator instructions after the traditional lesson. Second, we surveyed the 19 student volunteers to collect the feedback about the effectiveness of the 26 game elements present in the two selected serious games, as well as questions about serious games.

Our results indicate that, in the students' perception, some game elements are more relevant to learning programming than others. Students indicated some game elements they believe contribute to self-learning, such as Achievement, Chance and Level. On the other hand, other game elements are blamed to contribute for players abandoning the game. Game elements like "Hint" help the user to solve the levels of the game in a constructive way by acting as an educator providing tips for the student. The students also indicate that the serious games for learning programming contribute to maintain them motivated and that the games favor the learning process.

The remainder of this paper is organized as follows. Section II provides the background about learning programming, game-elements and serious games for learning programming. In Sections III and IV, we describe the settings of the study. Section V presents the results of the study. Section VI discusses the results on how game elements for learning programming can be used and improved. Section VII discusses the related work. Section VIII concludes this research paper and points out directions for future work.

[1] https://codecombat.com/
[2] https://www.codehunt.com/

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

II. Learning Programming with games

This section presents how students typically learn programming. We also present background of game elements.

Programming is usually taught at universities through traditional classes. Traditional classes consist of presentation of algorithms and data structures in slides or blackboard and the educator discusses the functionalities of the given algorithm or data structure [1]. Associated with classes in the classroom, students receive practical and theoretical exercises to reinforce the knowledge in classroom. During the learning process of programming outside the classroom, the student has some barriers to face. For example, the direct efforts to set up a programming environment for solving the exercises passed by the educator in the classroom. This type of effort could be avoided focusing on programming learning.

Serious games have been used as motivational strategies in introductory programming courses, as they may allow interactive entertained and engaged learning [11]. Two strategies have been used: creating or playing games. In the first strategy, students are asked to develop small games in order to apply the programming concepts [12]. In the second one, students play serious games to reinforce and practice concepts and programming skills [1].

Terminology and description for game elements are not uniform in the literature. Some works discuss the lack of a standard definition and nomenclature for game elements [13]. For instance, emblem [14] and badge [15] are two names for the same game element, which are visual rewards given to the user and identify user achievements in the game.

Since the 80s, research on which elements constitute the core of a game has been conducted. Previous work argues that elements, such as challenge, curiosity, control, and fantasy, constituted a core of a game [16]. Other work expanded this view to incorporate other elements, such as roles of a player, conflicts, even rules, goals, and constraints [17].

III. Using Game Elements Through serious games

This section introduces information about the courses where the study was conducted (Section A), and the serious games and their elements evaluated in this study (Section B).

A. The Programming Courses

The study described in this study was conducted in the Programming I and Programming II courses (P1 and P2, hereafter). P1 aims to introduce students to the basic concepts of programming using a language with a paradigm different from the discipline of algorithms and data structures using the Java language. In P1, students learn basic concepts of the object-oriented programming using the Java language. In addition, in course P1, students learn the basic syntax of the Java language and how to create algorithms and data structures in this language. This course is a prerequisite for P2.

The P2 course seeks to bring students advanced concepts of Java language and object orientation. P2 prepares students to develop software with a higher level of complexity, initiating the development of applications that integrate different data structures and programming techniques seen in the P1 course. In addition, students are introduced to the advanced concepts of object-oriented programming, such as multiple inheritance, interfaces, etc.

The two programming courses are taught by the same professor. It was the first experience of using serious games within the learning program of the P1 and P2 courses in this university. The serious games for programming did not replace the traditional lectures. The tasks in the games were assigned after the teacher explained the theoretical content in the classroom. The programming courses P1 and P2 are mainly based on two textbooks: “Java, How to Program”, by Deitel, H. and Deitel, P. [18] and “Thinking in Java” by Eckel, B. [19].

B. The Evaluated Serious Games and Game Elements

The selected games were Code Combat[3] and Code Hunt[4]. The purpose of both selected games is to teach programming using challenges for students. Both games are available for free to play on the internet. In addition, during the period of the P1 and P2 courses, students reported no problems regarding their availability. Both games increase the difficulty of levels as the player completes earlier stages with a lower level of difficulty. These games have been chosen because, besides being cited in scientific works [20], they have a satisfactory number of elements of games. Both games have a total of 26 game elements. However, excluding repeated elements among the two games, there are 19 unique game elements.

Table I shows the 19 unique game elements identified in the Code Combat and Code Hunt. Table I also shows a brief description of the game element. The game elements evaluated in this study were selected as follows: (i) two researchers played the two selected games. (ii) The researchers identified all game elements present in both games according to the definitions present in [6]. (iii) The two researchers discussed the identified game elements, to verify that both selected the game elements correctly.

As the goal of this work is to evaluate elements of games, we selected two serious games which cover topics taught in the classroom and that contain as many elements as possible. This choice was supported by our previous work [9], in which game elements were identified in several serious games for learning programming in higher education.

Code Combat provides programming learning for students and teachers with little or no previous programming knowledge in the Python and JavaScript languages. The game has elements, such as levels, progression, challenge, restriction, goals, avatar, tutorial, suggestions, badges, points, fantasy, health, boss, and multiplayer. In this serious game, the students follow a well-structured narrative with well-defined goals.

Code Hunt is an educational game in which learning to code is a product of solving a problem that is represented as the combination of inputs and outputs. Resulting in the improvement of the student computational logic. The game has elements, such as levels, leaderboards, restrictions, hints,

[3] https://codecombat.com/

[4] https://www.codehunt.com/

evaluation, challenge, point, and win state. The game has well-defined levels with sub-levels. Each sub-level has a puzzle to be solved. In addition, the game provides tips for the player to have a guide on how to solve the puzzles.

TABLE I. GAME ELEMENTS AND THEIR DESCRIPTION.

Game Element	Description
Achievement	A reward for completing a clear and desirable goal
Avatar	Visual representation of player character.
Badge	Visual representation of user conquests inside the game.
Boss	Represent hard challenges at the culmination of a level.
Challenge	Puzzle or other tasks that require effort to solve.
Chance	It allows the player to have new attempts to win a level in which he was defeated, without losing his current progress.
Fantasy	Context in which the player is inserted. For example, a space theme or a middle age theme.
Health Point	Health points of the player avatar. It adds difficulty to the game and causes the player to remain attentive to the game's restrictions so as not to reach a state of defeat.
Hint	Element that provides clues to player to solve a specific problem.
Item	Items acquired during the game that help the player complete missions and unlock features.
Leaderboard	Show the player progression regarding challenges completed. They also indicate how the player are performing against other players.
Level	Levels of the game are divisions of the game to make it more organized and to give the feeling of progression to the player.
Multiplayer	Allows players to cooperate at the same level of game.
Music	Music theme of the game that may or may not vary during the different levels of the game.
Point System	Used to rank the player against other users. In addition, can be used as a threshold for the user to unlock certain levels of the game.
Rating	Estimation or evaluation of the player's response during a stage of the game. It works as a measure of player performance during the game.
Restriction	Limitations imposed on the player during a game level, such as solving a part of the problem to go to a second part.
Sound	Sound of some movement or interaction of the character in the game or any indication that the user is doing a correct or incorrect action.
Timer	Timer whose goal is to create difficulties, limiting the player to complete the level in a certain time.

IV. STUDY SETTINGS

This section presents the goal of this study and its experimental steps. Section A presents the study goal and research questions. Section B explains the research method and steps we followed.

A. Study Goals and Research Questions

The goal of this study is to investigate what game elements contribute to motivate students in learning programming. To achieve this goal, we formulated two Research Questions (RQ) presented below:

RQ1. What is the students' perception on the relevance of each game element present in the serious games they played, regarding their learning experience?

RQ2. What are the student perceptions on the use of serious games for learning programming?

As a restriction of this study, the comparison of the two games is out of our scope. Therefore, our research questions are not meant to promote a discussion on which game is more adequate or effective for learning programming.

B. Study Design and Research Methods

To answer the research questions, we conducted an experiment using the games selected in two courses: P1 and P2.

The experiment was carried out between May and July 2017 at a public university. Participants are attending an Information Systems course. The P1 class had a total of 17 students and all participated in the experiment and the P2 class had a total of four students. All students were asked to play both games, but two students from course P2 played only one game. Participation in the experiment was not mandatory, if a student did not choose to participate, he would not lose points in the discipline. The students who participated earned extra points in the discipline and were able to choose activities they would not like to do outside the scope of our experiment.

First, we explained to students the purpose of the experiment, which is to evaluate the effectiveness of game elements for learning programming. We explained how the two games work and we played some levels during class. In addition, we explained to students' what game elements are, how they are divided and what is the state of the art research on elements of games. Students are asked to evaluate game elements in two ways: (i) Whether the game element was important for self-learning. (ii) Whether the game element, even if it was not important for self-learning, may be important for the learning of other students. (iii) If the element is definitely not effective for him or for other students.

Table II shows the questions regard the background of the participants of the experiment. The background questions were named BQ1 to BQ4. Table II also describes the possible answers for each question. All questionnaires of this study were applied in physical paper in the classroom under supervision of the authors of this study.

TABLE II. QUESTIONS ON THE PARTICIPANTS BACKGROUND

ID	Questions
BQ1	Do you like playing games? ( ) Yes. ( ) No.
BQ2	What is your programming knowledge? ( ) Basic. ( ) Intermediate. ( ) Advanced.
BQ3	Have you worked professionally with programming? ( ) Yes. ( ) No.
BQ4	How long you worked professionally with programming? ( ) Less than a year. ( ) Between 1 year and 3 years. ( ) More than 3 years.

Second, students who wanted to participate filled a consent form, allowing us to use the data they provided in a scientific study and preserving the anonymity of the participants would be maintained. After completing the consent form, we asked students to provide us with background on their programming skills and whether they enjoy serious games or not.

Third, we followed the P1 and P2 course plans and asked students to play a certain number of levels according to the

topics discussed in classroom. Students had to send screenshots to the instructors as evidence of completing each assignment.

In the last two weeks of the experiment, students were asked to answer two questionnaires (one for each game). In each questionnaire, student had to answer whether each of the game elements had positive effect on their learning experience. In addition, for each game element they had to write their opinion about the pros and cons of each element. All questionnaires have been scanned and are available online at a Google Drive folder: https://goo.gl/UdcJoM. There is no identification of the authors of this work in the shared folder, as well as in the scanned questionnaires.

TABLE III. QUESTIONS ON THE PARTICIPANT PERCEPTION ON THE EFFECTIVITY OF GAME ELEMENTS TO LEARNING PROGRAMMING

ID	Question
GEQ1	What is your assessment about game element "Level" to learning programming? (Open question) ( ) Effective. ( ) Not Effective.
GEQ2	What is your assessment about game element "Boss" to learning programming? (Open question) ( ) Effective. ( ) Not Effective.
GEQ3	What is your assessment about game element "Avatar" to learning programming? (Open question) ( ) Effective. ( ) Not Effective.
GEQ4	What is your assessment about game element "Fantasy" to learning programming? (Open question) ( ) Effective. ( ) Not Effective.

Due to page limits of this paper, Table III only presents questions for four elements. However, the questionnaires covers all 19 game elements identified in Code Hunt and Code Combat. Table III presents questions about game elements. The questions were named GEQ1 to GEQ19. For each game element, each student had to evaluate whether the game element is effective or not effective for learning programming. In addition, for each GEQ, the student should write an open answer stating the strengths and weaknesses of each element (open question). Due to the size of the questionnaires, 200 minutes of the course were used so that the students could answer the questionnaires without time restrictions.

Table IV shows the questions applied in the questionnaires about the two serious games selected for the experiment of this study. The questions were named GQ1 to GQ6. The questions in this table are the same for the two selected serious games. Table IV also describe the possible answers for each question.

To evaluate the open responses of each student, we use one stage of grounded theory called open coding [25]. This stage is a procedure for developing categories of information [25]. The open coding stage consists in examining the text for salient categories and the codes applied to the text is a labelling process. The codes work like keywords. They are not simple description of the text. For example, in some part of text there is a fragment containing the information "reading the schedule". This fragment is a description of some procedure. The refined way to coding this fragment, for example, is using the code "information gathering". This code can represent other fragments of text that are descriptions of process of gathering information.

TABLE IV. QUESTIONS ON THE PARTICIPANT PERCEPTION ABOUT THE USE SERIOUS GAMES IN LEARNING PROGRAMMING

ID	Question
GQ1	What do you think of the strategy of using games for learning programming? ( ) Does not yield results. ( ) Neutral. ( ) Yields results.
GQ2	Has the game taught you new knowledge? ( ) Yes. ( ) No.
GQ3	Does the game keep you focused? ( ) Yes. ( ) No.
GQ4	Does the game help you to reinforce the knowledge acquired in the classroom? ( ) Yes. ( ) No.
GQ5	Do you think that the game will make you achieve better results in disciplines that involve learning programming, algorithms and data structures? ( ) Yes. ( ) No.
GQ6	Do you think the game will make you achieve better results in a work environment? (If you work with programming). ( ) Yes. ( ) No.

V. RESULTS

In this section, we present the results of the present study. Section A provides an overview of the students background. Section B describes the results for the research question RQ1. Sections C describes the results for the research question RQ2.

A. Overview and Students Background

From the 19 students in the two programming courses P1 and P2, 17 participants answered the questionnaires of both games. Two participants answered one questionnaire. Three students choose to not answer the questionnaires. Some participants had to leave earlier in the class, leaving some answers blank and other students did not answer the questionnaires. We obtained 18 answered questionnaires about each game, totaling 36 answered questionnaires.

We consider only responses in which participants inform their personal opinion about the game element in relation to the effectiveness of programming learning. Responses from participants who wrote only the function of the element in the game were disregarded. This occurred because some participants may have thought the questionnaire was a traditional assessment of the course and, so, they answered in questions GEQ1 to GEQ19 with respect to the game element. In addition, we use as exclusion criterion, poorly formulated answers, not presenting clarity regarding the question.

Before we check the results about the game elements evaluated by the participants, it is important to check information about the background of the participants, such as programming knowledge and also about their preferences for games. Checking students' preference for games is important, since if most of the class does not like games, it is not a good strategy to use games to learning programming in the class.

The participants of the experiment are aged between 19 and 45 years. One participant is 45 years old, five participants aged 21 years, seven participants aged 19 years and six participants aged 20 years. There are six female participants and 13 male

participants. Nine participants are enrolled in the third semester and ten students in the fourth semester.

About the question BQ1 described in Table I, where we ask if the participant likes to play games. 18 participants (95%) said they liked games and one participant (5%) said they did not like it. Regarding the BQ2 question, we sought to know the programming knowledge of the participants (basic, intermediate or advanced). Eight participants (42%) stated that they had basic programming knowledge and 11 (58%) stated that they had intermediate knowledge. No participant claimed to have advanced programming knowledge.

In question BQ3, we ask the participants if they have professional experience. Five participants (26%) said they worked with programming and 14 (73%) said they did not work with programming. Complementing the BQ3 question, in BQ4 we asked how long they worked with programming. All five participants (100%) who claimed to have worked with programming reported that they had one to three years of professional experience.

The results show that most of the participants have basic programming knowledge and little work experience, which is explained by the fact that the participants are still in the initial periods of the course. In addition, few participants worked with programming. The classes have a homogeneous distribution of age group and sex. Only one participant of the entire population of participants said did not like games.

B. Perceptions on Game Elements Effectiveness (RQ1)

To evaluate participants' open answers, we used the labeling mechanism, which are keywords that identify information the participant wanted to express with their response. We call these keywords as code, just as they are called in the grounded theory technique.

In Table V we present the codes extracted from the open answers. The most recurring codes were: “self-learning”, cited by 18 students (94.74%); “attract attention”, cited by 17 students (89.47%); “keep focus” and “guidance”, mentioned by 16 students (84.21%). Other keywords mentioned by participants include: “motivation”, “challenging”, “competitiveness” and “game element improvement”. The least mentioned code by students was “challenging”, mentioned only once (5.26%). The code “increased difficulty” is used in two contexts. There are participants who said that some levels of the game are too easy and that the difficulty would need to be increased. On the other hand, some students said that the difficulty of the game increases considerably in the last levels so that the difficulty is proper to complete the given level. These different contexts for the same code are respectively exemplified in the following excerpts:

Subject 7: "*During the progress of the game, challenges could have increased difficulty*."

Subject 16: "*As the game brings an increased difficulty through the game element challenge, makes me think more about solving the challenge*."

Table VI presents the results of objective answers about effectiveness of each game element present in Code Combat. Each participant had to answer in questions GEQ1 to GEQ19 if their perception about the game element in question was effective or not effective. Before we detail the results is important to clarify that we do not want make game comparison. We aim to identify what game elements are effective and those that are not effective.

TABLE V. CODES EXTRACTED FROM THE OPEN ANSWERS GEQ1 TO GEQ19.

Code	# participants	% participants
Self-learning	18	94,74%
Attract attention	17	89,47%
Guidance	16	84,21%
Keep focus	16	84,21%
Motivation	14	73,68%
Game element improvement	10	52,63%
Increased difficult	10	52,63%
Irrelevant	8	42,11%
Competitiveness	4	21,05%
Discourage	4	21,05%
Players Interaction	4	21,05%
Under pressure	2	10,53%
Challenging	1	5,26%

**Sample size = 19 participants*

TABLE VI. PARTICIPANTS PERCEPTIONS ON THE USE OF GAME ELEMENTS IN THE CODE COMBAT GAME.

Game Element	Effective	Not Effective
Achievement	19	0
Chance	19	0
Level	19	0
Challenge	18	1
Health Point	18	1
Item	18	1
Rating	18	1
Leaderboard	17	2
Restriction	17	2
Badge	16	3
Fantasy	16	3
Avatar	15	4
Point System	15	4
Sound	15	4
Boss	14	5
Hint	14	5
Music	14	5
Multiplayer	13	6
Timer	13	6

As can be seen in Table VI the majority of participants answer that the 19 game elements are effective. The game elements: Achievement, Chance and Level were considered effective by 100% of participants. The game elements Hint and Timer were the elements that received most not effective evaluation.

In Table VII we present the results of objective answers about effectiveness of each game element present in Code Hunt. The elements Evaluation, Challenge, and Leaderboard were evaluated as effective by the highest number of participants. Point and Restriction had mixed opinions regarding their effectiveness. Only the element Sound was considered not effective by the participants.

TABLE VII. PARTICIPANTS PERCEPTIONS ON THE USE OF GAME ELEMENTS IN THE CODE HUNT GAME.

Game Element	Effective	Not Effective
Challenge	11	6
Leaderboard	12	7
Level	10	7
Point	8	7
Rating	14	3
Restriction	9	8
Sound	3	14

Table VIII presents maps the codes extracted from the open answers for each game element evaluated in this study. The code "attract attention" appears 38 times in participants responses for the game elements: Level, Chance, Challenge, Avatar, Achievement, Badge, Point System, Sound, Music, boss, Multiplayer and Fantasy. In total attract attention is related to 13 game elements. The elements that attract more attention of participants was: fantasy, multiplayer and avatar. In Table V the code attract attention was cited by 17 different participants, but one participant can cite the code for more than one element. Continuing in Table VIII, the code "keep focus" was related to nine game elements. The element that most maintain participants focus was Health Points. The code "motivation" was related to 11 game elements, where Achievement and Challenges were the game elements most related to this code.

C. Perceptions on Use of Serious Games for Learning (RQ2)

In conjunction with the evaluation of Code Combat and Code Hunt game elements, we surveyed participants about the use of serious games for learning programming. The purpose of this investigation is to know if the use of games, as a whole, contributes to the learning of programming.

The first question GQ1 ask participants about what they think about the strategy of using serious games to learning programming. Participants had two options to answer this question, that is, if they think that the strategy yields results, not yield results or if they think the strategy is neutral, that is, the participant does not have an opinion about the strategy. For this question, 16 (84%) participants answer that the strategy yields results to learning programming. The remainder of participants 16% of them affirm that the strategy of use games to learning programming is neutral. None participant affirm that the strategy does not yield results.

TABLE IX. PARTICIPANTS PERCEPTIONS ON THE USE OF SERIOUS GAMES TO LEARNING PROGRAMMING.

Question	Yes	No
GQ2	71%	29%
GQ3	100%	0%
GQ4	73%	27%
GQ5	68%	32%
GQ6	42%	58%

Table IX concentrates the remaining questions about use of serious games to learning programming. In all questions participants has two options to answer: yes or no. In the question GQ2 71% of participants answer that the use of serious games taught new knowledge and 39% that the use of serious games does not taught new knowledge. Regarding GQ3, 100% of participants affirm that the game keeps them concentrated.

For GQ4, 14 participants (73%) affirm that the use of serious games helps them to reinforce the knowledge acquired in classroom. Five participants (27%) said that serious games do not help to reinforce knowledge acquired during traditional class. This result of students that answer those serious games does not help to reinforce knowledge acquired in classroom, makes sense with the negative answers of GQ2, because, if students said that games does not taught new knowledge (29%), it is expected that games does not help to reinforce the knowledge acquired in classroom.

For GQ5, 13 participants (68%) answer that the serious games helps to achieve better results in other courses that involves learning of programming or data structures. On the other side, six participants (32%) thinks that the serious games

TABLE VIII. GAME ELEMENTS RELATED TO CODES EXTRACTED FROM PARTICIPANTS RESPONSES.

Elements / **Codes**	Level	Restriction	Rating	Chance	Challenge	Hint	Avatar	Leaderboard	Achievement	Badge	Point System	Sound	Music	Health Point	Timer	Boss	Ranking	Multiplayer	Fantasy	**Total**
Attract attention	2			2	1		5	1	3	3	2	2	2			4		5	6	38
Keep focus	3		2	3	3			1		2	2			7		1				24
Motivation			1		4	1		1	4	2	2			1		1	3	1		21
Self-learning	3	3		4	6	2														18
Guidance			4	1		9		2												16
Game Elem. Imp.			1	1		2	2	1			3				1					11
Increased difficult	2	2			1									1	1	3				10
Competitiveness																	4			4
Discourage		4																		4
Players interaction																	2	2		4
Irrelevant							2		1	1	1	7	7		1		2			22
Under pressure															2					2
Challenging														1						1

does not help to achieve better results in other courses that involve learning programming and data structures. In GQ6, eight students (42%) affirm that the use of serious games in classroom can contribute to achieve better results if they work professionally with programming. On the other hand, 11 students (58%) answer that use of serious games does not help to achieve better results in a professional work environment.

VI. DISCUSSION

This section discusses the results of Section V.

A. Perceptions on Use of Serious Games for Learning

Regarding the students' perceptions about use of serious games to learning programming in higher education GQ1, the results were positive. 84% of participants answer that the strategy of using serious games yields results to learning programming. That indicates that students would like to have their classes blended with new strategies. This result comes in conjunction with the motivation of this work, which is to investigate new strategies for teaching programming, seeking to motivate students to programming learning, which is a discipline that has a high number of dropouts.

The participants also answered that serious games taught them new knowledge GQ2. In total, 13 participants (71%) answer that serious games taught new knowledge. This result can motivate experiments in higher education, using only serious games to learning programming without the need of traditional classes, using the professor as a resource to guide students through game levels. This type of experiment can also be supported by the answer of GQ3, that 19 participants (100%) answer that the games used the experiment keep them concentrated. Still on GQ2, the game taught new knowledge to 13 participants (71%), but all game content was seen in the classroom. This indicates that it can often occur that the participants did not understand some concept during the traditional class and during the game, they saw something that he had not noticed during the class and therefore said that the game taught him new knowledge.

In GQ6, 11 participants (58%) stated that games would not make them achieve better results in a professional working environment. This result may be related to some students answering the open questions that they think that game Code Combat has little difficulty, and that in a professional environment they would face more complex challenges.

About serious games, one participant said he did not like it. However, the subject stated that the games used in the experiment kept him focused. This shows that perhaps using games can motivate students who do not like games to start liking it. Since if the student replied that he does not like games but that the game kept him focused, there are indications that after playing games, their opinions may change.

B. Perceptions on Relevance of Game Elements

Participants in the experiment reported interesting thoughts about the game elements for learning programming. Participants not only reported their perceptions about the game elements, but also suggested improvements for some elements, such as improvement in the game element point system.

Regarding game element restriction, some participants report that restrictions on the game increase the difficult and others report that these restrictions in conjunction with game element Level makes the game tedious, due to the reason that some participants have some previous knowledge about the game. Below some quotes about the participants` perceptions about game element restriction:

Subject 2: "*I think that element restriction is effective for those who are starting in programming, but for me that already have a background in programming is boring.*"

Subject 5: "*In my case that have more knowledge about programming, the game became tedious, due to the fact that I have to pass through initial restrictions of the game. There could be a way for the player to choose which levels he wants to go. Separate levels by difficulty and the player goes on the difficulty he wants.*"

The code hunt and code combat games do not offer players the option of difficulty selection. In order for the player to reach levels of greater difficulty, it is mandatory for the player to pass through the levels of least difficulty.

Subject 9: "*The task makes me try harder to solve a level, with the goal of knowing the next levels and finish the game.*"

Subject 15: "*Level constraints are important because they ensure linear learning, making less experienced players understand function by function and build knowledge slowly.*"

Some participants suggested changes in the element point system, claiming that the point system of the games only serve to rank players in the overall leaderboard. Some participants suggest creating different point systems based on skill and level of each player. They also suggested creating a different Leaderboards according to the player level. In addition, participants suggested creating a system that users can battle each other according to their ranks.

Subject 4: "*I think that leaderboards could be classificatory, if player take determined time to solve a level, the player will be ranked in a certain leaderboard, for example, leaderboard of gold players, silver players, etc.*"

Subject 5: "*Another prize for the player, the points obtained in levels could also be applied to the characters, gaining each level new skills to use.*"

Some game elements not only play the role of reinforcing players' knowledge and keeping them motivated. Several participants indicated that they feel guided by the game. The sense of orientation may be related to the student's ability not to need help from others to continue doing the tasks of the game.

Subject 4: "Game element hint *encourages the player to follow some solution that he imagined for the problem and was not sure if he should implement it. Also, encourage the player to keep trying if he does not know what to do in any way.*"

Subject 10: "*In case of doubt, hint game element guarantees the player the understanding of the doubts. Not needing anyone to help.*"

Subject 13: "*The hints of the game are accurate, comprehensive and this makes me motivated with the game.*"

The results of Table VI shows that, achievement, level and chance were 100% effective, these elements are important in the game, since they are related to similarities of academic life, that is, students want to obtain achievements in their academic tasks, want to have opportunities to try again some task after an error. Just like in the game, during the academic career, the knowledge being acquired in levels.

Badges, Achievements and Challenges maintain some participants focused and aware about they do in the game. Some thoughts about these elements according to the subjects of the experiment:

Subject 5: "*badges, as well as, achievements keep me focused and happy with the game.*"

Subject 18: "*challenges keep me focused and it makes me spend a lot of time playing the game.*"

According to subject 18, the challenges make the player spend a lot of time in the game. That is a good point. Since it is important that the player is dedicated to solve the challenges of the game. Since in a professional working environment challenges of this type will arise and this type of challenge requires that time be spent for it to be resolved.

VII. Related Work

In this section, we discuss the related research on the evaluation of students' perceptions about effectives of game elements to learn programming in higher education.

Some researchers evaluates the relationship between game elements and learning outcomes for all educational purposes [5] [21]. However, these works have some shortcomings, such as, making conclusions of learning outcomes and game elements through case study using one game in non-academic environments. In addition, there is a lack of experiments with a considerable number of students, as well as evaluating the results using statistical tools.

Malone and Lepper [22] conducted experiments using a serious game with children to see which game elements make learning more fun. The authors identified that the elements that contribute most to the learning in these experiments were: curiosity, sensory, fantasy, cooperation, and competition. They evaluate elements regardless area of knowledge.

Wilson et al. [7] conducted a systematic review in the literature to understand the "state of play" regards to learning outcomes and game attributes. The authors also seek out what specific game attributes have an impact on learning outcomes. Furthermore, authors identify gaps in this research area. Wilson et. Al. [7] found 18 game elements in literature review, but they focused the discussions of the work on a subset of seven game elements. Their work focuses the discussions on how game elements are related to learning objectives, regardless of area of knowledge. However, the authors do not evaluate the game elements with the students, as done by our work.

Souza [13] performed an experiment introducing two game elements, namely badges and leaderboards, in an introductory Software Engineering course. The goal of the study is to evaluate the students' perception on the impact of these elements in their motivation towards the course. The study of Souza is not related to learning programming and it uses concepts of gamification; that is, the use of game elements in non-games contexts [6].

Tihomir [20] conducted a study that aimed to examine the quality of serious games designed for learning programming from students' perspective and an empirical study was carried out. Study participants were students enrolled in different courses at two Croatian higher education institutions. Authors used two games to evaluate the students' opinions: LightBot and Code Combat. The work focuses on evaluating games in aspects like: learnability, reliability, accessibility, helpfulness, effectiveness, playfulness, and satisfaction. The authors did not focus in evaluate the game elements.

VIII. CONCLUSION

In this study we performed an experiment to verify new strategies for learning programming. Our experiment consisted of adopting a hybrid strategy that involved adding the use of game elements in conjunction with traditional classes. In this work, we verify what are the students' perception about the use of game elements to learning programming. In total, we evaluate 19 game elements with students

We noticed that several game elements contribute positively to learning programming. Participants in the study said that some elements such as achievements, challenges, levels, and hints, help them stay focused on the game, as well as make them spend more time in the game by addressing the challenges posed by these serious games. The game element Hint helps students to solve challenges inside the game without consult a professor.

All game elements were considered effective by about 68% of the participants in the Code Combat game. In the Code Hunt game, the sound element was considered not effective for 73% of participants. Regarding the use of serious games for learning programming, students said that games taught them new knowledge, kept them focused and that the games helped them to reinforce the knowledge acquired in classroom. This shows us indications that the use of game elements in conjunction with traditional classes brings positive results in learning programming. We did not evaluate students' grades compared to previous semesters. For future work, we plan to conduct these experiments with other serious games that contain other elements and evaluate student performance using hybrid versus traditional classrooms for learning programming.

ACKNOWLEDGEMENTS

This work was partially supported by CAPES, CNPq (grant 424340/2016-0), and FAPEMIG (grant PPM-00651-17).

REFERENCES

[1] M. Rahmat, S. Shahrani, R. Latih, N. F. M. Yatim, N. F. A. Zainal and R. Ab Rahman, Major Problems in Basic Programming that Influence Student Performance, Procedia - Social and Behavioral Sciences,Volume 59, 2012, Pages 287-296.

[2] Carla De Lira. 2017. Improving the Learning Experiences of First-Year Computer Science Students with Empathetic IDEs. In Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17). p. 293-294.

[3] Christopher Watson and Frederick W.B. Li. 2014. Failure rates in introductory programming revisited. In Proceedings of the 2014 conference on Innovation & technology in computer science education (ITiCSE '14). p. 39-44.

[4] P. Kinnunen and L. Malmi. 2006. Why students drop out CS1 course?. In Proceedings of the second international workshop on Computing education research (ICER '06). p. 97-108.

[5] Bedwell, W. L., Pavlas, D., Heyne, K., Lazzara, H. E., Salas, E., 2012. Toward a Taxonomy Linking Game Attributes to Learning: An Empirical Study. Simulation & Gaming An Interdisciplinary Journal, p. 729–760.

[6] Deterding, S and Dixon, D., 2011. Gamification: Using game design elements in non-gaming contexts. In Extended Abstracts on Human Factors in Computing Systems (CHI).

[7] Wilson, K. A., Bedwell L. W., Salas, E., Burke, S. C, Estock, J. L., Orvis, K. L., Conkey, C., 2009. Relationships Between Game Attributes and Learning Outcomes: Review and Research Proposals. In Simulation & Gaming: An Interdisciplinary Journal, 40 (2) 217-266.

[8] G. Petri, C. G. von Wangenheim, How games for computing education are evaluated? A systematic literature review, Computers & Education, 2017, p. 68-90.

[9] Lages dos Santos A., R. de A. Souza M., Figueiredo E. and Dayrell M. (2018). Game Elements for Learning Programming: A Mapping Study.In Proceedings of the 10th International Conference on Computer Supported Education - Volume 2: CSEDU, pages 89-101.

[10] ACM and IEEE, 2013. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science.

[11] Aldrich, C. 2009. Learning Online with Games, Simulations, and Virtual Worlds: Strategies for Online Instruction. San Francisco: John Wiley & Sons.

[12] Leutenegger, S. and Edgington, J. "A games first approach to teaching introductory programming". 2007. Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education. p. 115–118.

[13] M. R. d. A. Souza, K. F. Constantino, L. F. Veado and E. M. L. Figueiredo, "Gamification in Software Engineering Education: An Empirical Study," 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T), Savannah, GA, 2017, pp. 276-284..

[14] García, F., Mario, P. P., Cerdeira-Pena, A., Penabad, M., 2017. A framework for gamification in software engineering. Journal of Systems and Software (JSS), 132, 21-40.

[15] Hamari, J., 2017. Do badges increase user activity? A field experiment on the effects of gamification. In Computers in Human Behavior, 469-478, Elsevier.

[16] Malone, T. W., 1981. Towards a theory of intrinsically motivation instruction. Cognitive Science, 4, 333-369.

[17] Gredler, M. E., 1996. Educational games and simulation: A technology in search of a research paradigm. Handbook of research for educational communications and technology (pp. 521-540).

[18] Harvey M. Deitel and Paul J. Deitel. 2001. Java how to Program (4th ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[19] Bruce Eckel. 2005. Thinking in Java (4th Edition). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[20] Tihomir, Orehovacki. Babic, Snjezana., Zaphiris, Panayiotis., Ioannou, Andris. Inspecting Quality of Games Designed for Learning Programming. In Learning and Collaboration Technologies, 2015. Springer International Publishing.

[21] Garris, R., Ahlers, R., Driskell, J. E. (2002). Games, motivation and learning: A research and practice model. Simulation & Gaming: An Interdisciplinary Journal, 33, 441-467.

[22] Malone, T. W., Lepper, M. R., 1987. Making learning fun: A taxonomy of intrinsic motivations for learning. In Aptitude, learning and instruction: Vol. 3. Cognitive and affective process and analyses (pp. 223-253).

[23] Mauricio R A Souza, Lucas Veado, Renata Teles Moreira, Eduardo Figueiredo, and Heitor Costa. 2017. Games for learning: bridging game-related education methods to software enginering knowledge areas. In Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track (ICSE-SEET '17). pp, 170-179.

[24] Strauss, A. L., & Corbin, J. M. 1990. Basics of qualitative research: Grounded theory procedures and techniques. Newbury Park, Calif: Sage Publications.

[25] J. Bishop, R. N. Horspool, T. Xie, N. Tillmann, and J. de Halleux. 2015. Code hunt: experience with coding contests at scale. In Proceedings of the 37th International Conference on Software Engineering - Volume 2 (ICSE '15), Vol. 2. IEEE Press, 398-407.