Climbing Up the Leaderboard: An Empirical Study of Applying Gamification Techniques to a Computer Programming Class

Panagiotis Fotaris[1], Theodoros Mastoras[2], Richard Leinfellner[1] and Yasmine Rosunally[3]
[1]University of East London, School of Arts and Digital Industries, London, UK
[2]University of Macedonia, Department of Applied Informatics, Thessaloniki, Greece
[3]University of the West of England, School of Computing and Creative Technologies, Bristol, UK
p.fotaris@uel.ac.uk
mastoras@uom.gr
r.leinfellner@uel.ac.uk
yasmine.rosunally@uwe.ac.uk

Abstract: Conventional taught learning practices often experience difficulties in keeping students motivated and engaged. Video games, however, are very successful at sustaining high levels of motivation and engagement through a set of tasks for hours without apparent loss of focus. In addition, gamers solve complex problems within a gaming environment without feeling fatigue or frustration, as they would typically do with a comparable learning task. Based on this notion, the academic community is keen on exploring methods that can deliver deep learner engagement and has shown increased interest in adopting gamification – the integration of gaming elements, mechanics, and frameworks into non-game situations and scenarios – as a means to increase student engagement and improve information retention. Its effectiveness when applied to education has been debatable though, as attempts have generally been restricted to one-dimensional approaches such as transposing a trivial reward system onto existing teaching materials and/or assessments. Nevertheless, a gamified, multi-dimensional, problem-based learning approach can yield improved results even when applied to a very complex and traditionally dry task like the teaching of computer programming, as shown in this paper. The presented quasi-experimental study used a combination of instructor feedback, real time sequence of scored quizzes, and live coding to deliver a fully interactive learning experience. More specifically, the "Kahoot!" Classroom Response System (CRS), the classroom version of the TV game show "Who Wants To Be A Millionaire?", and Codecademy's interactive platform formed the basis for a learning model which was applied to an entry-level Python programming course. Students were thus allowed to experience multiple interlocking methods similar to those commonly found in a top quality game experience. To assess gamification's impact on learning, empirical data from the gamified group were compared to those from a control group who was taught through a traditional learning approach, similar to the one which had been used during previous cohorts. Despite this being a relatively small-scale study, the results and findings for a number of key metrics, including attendance, downloading of course material, and final grades, were encouraging and proved that the gamified approach was motivating and enriching for both students and instructors.

Keywords: gamification, game-based learning, learning and teaching, technology enhanced learning, virtual learning environment, classroom response system, Kahoot, assessment, higher education

1 Introduction

According to research on the dynamics of attention spans during lectures, the typical learner's attention increases during the first ten minutes of lecture and diminishes after that point (Hartley and Davies, 1978). One way to address this issue and recapture the attention of learners is by changing the environment during a lecture, e.g., via a short break (McKeachie, 1999). This is almost the opposite of the dynamic experienced by video gamers. The latter are kept at high levels of attention, which in some cases can last for many hours (Green and Bavelier, 2006). They also have a distinct characteristic where they strive to be on the verge of what Jane McGonical (2010) describes as an "epic win". Gamers also share common factors such as urgent optimism, social fabric, blissful productivity, and epic meaning, which in turn make them super empowered hopeful individuals (Huang and Soman, 2013). On the other hand, when confronted with complex learning, students are more likely to feel overwhelmed; there is no instant gratification or short term wins to keep them engaged and motivated. A promising way to address these counterproductive feelings is to design them out using techniques similar to ones found in successful gaming environments.

Reference this paper as Fotaris P, Mastoras T, Leinfellner R and Rosunally Y, "Climbing Up the Leaderboard: An Empirical Study of Applying Gamification Techniques to a Computer Programming Class" *The Electronic Journal of e-Learning Volume 14 Issue 2 2016*, (pp94-110) available online at www.ejel.org

Rather than assuming that the rapid proliferation of sophisticated technologies such as smartphones, tablets, and laptop computers into every facet of society is the cause of student attention deficit (Griffin, 2014), educators should be open to new possibilities to teach and educate (Squire, 2003; de Aguilera and Mendiz, 2003). Findings of independent experiments performed in secondary and higher education settings showed that students who were subjects to learning with video games reported significant improvements in subject understanding, diligence, and motivation (Barata et al., 2013; Coller and Shernoff, 2009; Kebritchi et al., 2008; Lee et al., 2004; McClean et al., 2001; Squire et al., 2004).

In the same way that games help stimulate the production of dopamine, a chemical that is considered to play a key role in motivation, affect and learning (Wimmer et al., 2014), educational techniques which access the same methodologies could result in learning-reward cycles (Gee, 2003) by reinforcing neuronal connections and communications during learning activity (NMC Horizon Report, 2013). Additionally, unlike the one-size-fits-all lecture, these game-based techniques can be balanced to be appropriate to the learners' skill level (Koster, 2004) in order to prevent them from becoming frustrated or bored, thus allowing them to experience "flow", i.e., a user's state of "optimal experience" (Barata, 2013; Chen, 2007; Csikszentmihalyi, 1990).

Gamification for learning should use game mechanics, dynamics, and frameworks to non-game processes along the following principles, which were adapted from Self-Determination Theory (Ryan and Deci, 2000):

- Relatedness – the universal need to interact and be connected with others;
- Competence – the universal need to be effective and master a problem in a given environment;
- Autonomy – the universal need to control one's own life.

These elements have been shown to affect intrinsic and extrinsic motivation, which in turn can have a big impact on student engagement and motivation (Deterding et al., 2011). Intrinsic motivation (e.g., altruism, competition, cooperation, sense of belonging, love or aggression) is driven by an interest or enjoyment in the task itself and inspires people to initiate an activity for its own sake (Ryan and Deci, 2000). Students who are intrinsically motivated are more likely to engage in a task willingly, as well as work to improve their skills, which will increase their capabilities (Wigfield et al., 2004). In contrast, extrinsic motivation comes from outside the individual and refers to the performance of an activity in order to attain an outcome (e.g., earn grades, levels, points, badges, awards) or to avoid punishment (Muntean, 2011). Typical extrinsic incentives include competitions, cheering crowds, and desire to win trophies.

Individual student fatigue could be taken into account so as to determine the optimal combination of intrinsic and extrinsic motivators; this would automatically re-captivate students and provide a rewarding break without producing any detrimental effects. By introducing game mechanics into generally unpopular activities such as assessments, students would enjoy the tasks first and, in the process of completing them, they would deliver the required assessment.

However, despite the fact that gamification of education is gaining support among an increasing number of academics who recognise that effectively designed games can stimulate large gains in productivity and creativity among learners (NMC Horizon Report, 2014), opponents argue that what is lacking is concrete empirical data to support or refute these theoretical claims (Annetta et al., 2009; Barata et al., 2013). Some of the negative experiences include disappearance of collaboration among students and overstimulation of competitiveness. The balance between learning, social collaboration, creativity, and competitiveness which is apparent in mainstream commercial games seems to be hard to achieve in tools specifically designed for education (Zaha et al., 2014). As a result, gamification is often reduced into a behaviour model leveraging human need for positive reward system and instant gratification, which is applied to a traditional teacher-centred classroom.

Annetta et al. (2009) and Britain and Liber (2004) suggest that both teachers and researchers need to evaluate video games and gamification from an educational perspective, in order to determine whether they can be embedded into teaching practices. Based on this notion, the present paper aspires to make a contribution to the empirical evidence in the gamification field by designing, implementing and evaluating a gamified learning experience in a higher education setting. This research effort tries to bridge the gap between theory and practice, as well as to study the educational impact of gamification in a real educational setting. The specific research questions were:

- Are students who use Codecademy and play “Who Wants To Be A Millionaire?” and “Kahoot!” more engaged in learning Python programming when compared to peers who engage in traditional class activities?
- Do students who use Codecademy and play “Who Wants To Be A Millionaire?” and “Kahoot!” develop deeper understandings of Python programming when compared to peers engaged in more traditional instruction?

2 Related works

The idea of using gamification for learning is not entirely new. In the 1980s Malone (1980; 1981; 1982) did research on what makes video games attractive to players and how these aspects can be applied to education as a means to promote student engagement and motivation. Carroll (1982) analysed the design of the seminal text adventure “Adventure”, which in turn led him to propose redressing routine work activities in varying “metaphoric cover stories” in order to turn them into something more intrinsically interesting, and to “urge for a research program on fun and its relation to ease of use” (Deterding et al., 2011; Carroll and Thomas, 1982).

The new millennium saw the introduction of the terms “ludic engagement”, “ludic design”, and “ludic activities” to describe “activities motivated by curiosity, exploration, and reflection” (Gaver et al., 2004), as well as the emergence of a new field called “’funology’ – the science of enjoyable technology” (Blythe et al., 2004) which was inspired by game design and studied “hedonic attributes” (Hassenzahl, 2003) or “motivational affordances” (Zhang, 2008) of “pleasurable products” (Jordan, 2002). Related research focused on using game interfaces and controllers in other contexts (Chao, 2001), creating “games with a purpose” in which game play is employed to solve human information tasks (e.g., tagging images) (Ahn and Dabbish, 2008), and exploring “playfulness” as a desirable user experience or mode of interaction.

The use of video games for educational purposes was also emphasized by the works of Prensky (2001) and Gee (2003). Although these studies were related to game-based learning rather than gamification, their findings form the core of gamification in education: they described the influence of game play on cognitive development, identified 36 learning principles found in video games, and recognised potential advantages of video games in learning such as the value of immediate feedback, self-regulated learning, information on demand, team collaboration, and motivating cycles of expertise (Borys and Laskowski, 2013).

More recently, major corporations and organisations including Adobe (LevelUp, Jigsaw - Dong et al., 2012), Microsoft (Ribbon Hero), IBM (SimArchitect - IBM Global Business Services, 2012), and Autodesk (GamiCAD - Li et al., 2012) consulted with game experts to develop gamified systems that focus on keeping users engaged while learning new software and techniques. Other successful cases of gamification in education include Khan Academy, Treehouse, Udemy, and Duolingo, organisations that provide access to a rich library of content (including interactive challenges, assessments and videos on several subjects) and use badges and points to keep track of student progress. Codeacademy is an e-learning platform specialised for computer programming, designed with gamification in mind, while Kahoot is an example of a popular game-based Classroom Response System (CRS, also commonly known as “clicker”) (Fies and Marshall, 2006) that can be played on any device with a browser, both in online and traditional learning environments.

In the context of higher and secondary education, gamification can be applied at vastly different scales to any discipline. At one end is gamification at the micro-scale: individual teachers who gamify their own class structures (Lee and Hammer, 2011) such as Lee Sheldon (2011), a professor at Rensselaer Polytechnic Institute who turned a conventional learning experience into a game without resorting to technology by discarding traditional grading and replacing it with earning “experience points”, while also converting homework assignments into quests (Laster, 2010). At the other end of the scale, a charter school in New York City called “Quest to Learn” uses game design as its organising framework for teaching and learning. Teachers collaborate with game designers to develop playful curricula and base the entire school day around game elements (Corbett, 2010).

To summarise, although the amount of literature on gamification in education is constantly increasing, the wide range of course types, learning preferences, student backgrounds, and socio-economical environments requires more systematic studies of the influence of different gamification techniques in order to assess their efficiency (Barata et al., 2013).

3 Methodology

3.1 Study design and sample

Teaching and assessment of computer programming is considered to be difficult and frequently ineffective, especially to weaker students, as computer programs and algorithms are abstract and complex entities that involve concepts and processes which are often found hard to teach and learn (Olsson et al., 2015; Lahtinen et al., 2005). This sometimes results in undesirable outcomes such as disengagement, cheating, learned helplessness, and dropping out (Robins et al., 2003; Winslow, 1996). Furthermore, most students would not describe classroom-based activities in school as playful experiences. However, research on multimodal teaching has shown that adding more channels for the knowledge transfer can facilitate learning in general (Olsson et al., 2015). Based on this fact and the concepts of the increasingly popular gamification, game-based learning, and serious games movements, the present paper evaluates how gamification affected students of a 12-week university course named “Fundamentals of Software Development” (FSD) via the use of the “Kahoot!” CRS, a modified classroom version of the TV game show “Who Wants To Be A Millionaire?” (WWTBAM), and Codecademy’s online interactive platform.

To reach this objective faculty staff composed of three lecturers conducted a quasi-experimental study over two consecutive academic years at the School of Computing and Technology, University of West London. The sample included a control class (CC) of N_{con} = 54 students (43 males, 11 females) who attended the FSD course in the first year of the study, and an experimental class (EC) of N_{exp} = 52 students (44 males, 8 females) who attended FSD in the second year. The participants ranged from 19 to 25 years of age. Additionally, 16 students of the experimental group were regular gamers (31%), 28 played games occasionally (54%), and 8 did not play video games at all (15%).

During the first year FSD followed a non-gamified approach that was similar to the one used in previous years. The syllabus included 12 regular one-hour lectures, 12 two-hour laboratory classes, and 12 one-hour seminars. The theoretical lectures covered Python programming concepts ranging from loops, functions, and object-oriented programming, to GUI applications and videogame development. In laboratory classes students were presented with a series of programming tasks that they had to complete individually during the session, with the tutors offering occasional help. Finally, seminars were used for revision purposes and were delivered via a combination of Q+A and typical lectures. All course materials were uploaded to the institutional Virtual Learning Environment (Blackboard) on a weekly basis. The course evaluation consisted of 6 theoretical quizzes (30% of total grade) and 2 mandatory assessments: a final exam (35%) and a programming project (35%).

An analysis of the student performance data at the end of the first year showed low attendance rates, numerous late arrivals to classes, and lack of interest in the reference material (low number of downloads that increased only before the exams period). In order to address these issues and to make FSD more fun and engaging, teaching methods changed in the second year to incorporate gamification. Literature indicates that educational gameplay fosters engagement in critical thinking, creative problem solving, and teamwork (NMC Horizon Report, 2014). When students are actively engaged in the content that they are learning, there is increased motivation, transfer of new information, and retention (Premkumar and Coupal, 2008). Additionally, the attention span of students diminishes after the first 15-20 minutes into a lecture (Middendorf & and Kalish, 1996). Based on these facts, while the course evaluation remained the same, the delivery of the course was gamified as follows.

3.2 Gamification of the course

3.2.1 Formative assessment using Kahoot!

The initial one-hour theoretical lectures were replaced by three 20-minute cycles of a micro-lecture, a formative assessment in the form of a Kahoot! game, and a brief discussion. As mentioned earlier, Kahoot! is a web-based CRS (Hwang et al., 2015) that uses colourful graphics and audio to temporarily transform a classroom into a game show, with the instructor acting as the show host and the students being the competitors. Every week the instructor created three Kahoot! games based on the topics that were going to be covered in the three micro-lectures of the upcoming class. After a micro-lecture was completed, the instructor launched its related Kahoot! game, which in turn generated a unique game pin for each session. Students then used their own digital devices (tablets, smartphones, laptops) or the class desktops to log-in to the game, enter the game pin, and create a username that would be displayed as the game progressed. Once everyone had joined the game, the instructor's computer, which was connected to a large screen, displayed a set of 5 MCQs for students to answer on their devices. Each answer was transmitted to Kahoot!'s online processing unit (server) which analysed it and rewarded students with points according to their accuracy and response time (Figure 1). Between each question Kahoot! showed a distribution chart of the students' answers, thus allowing the instructor to receive immediate feedback on whether concepts had been understood by the whole class or required further elaboration; in the latter case, he paused the game and offered any required explanations. Consequently, a scoreboard revealed the nicknames and scores of the top five responders, and at the end of the game a winner was announced and received some candy as a reward.

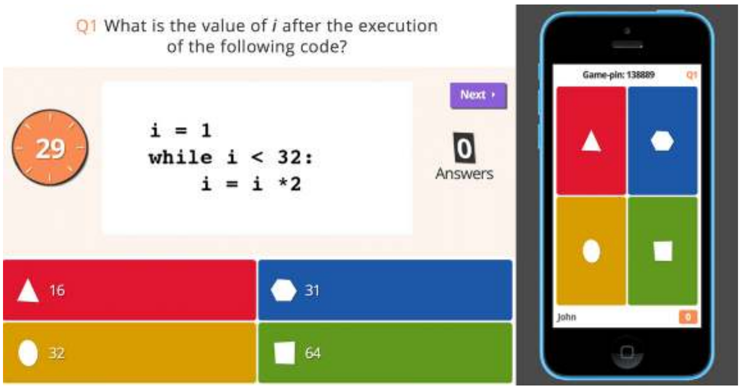

Figure 1: "Kahoot!" in-game screenshot

Following the game's completion, the instructor discussed briefly all answers to each question and downloaded a spreadsheet of the results in order to get an overview of the individual student and overall class performance. Each student's score was updated every week and was entered to a leaderboard webpage, which was publically accessible through Blackboard and displayed enrolled students in descending order according to their total points. This visual display of progress and ranking provided students with direct feedback on their performance against both their own goals and the performance of their peers, while also serving as instant gratification. The thinking behind this decision was that rankings tap into people's natural competitiveness and encourage them to do better, which might motivate students to study more by the desire to improve their position (Natvig et al., 2004).

3.2.2 Collaborative problem solving with "Who Wants To Be A Millionaire?"

The one-hour revision seminar was also changed; the combination of Q+A and lectures that took place during the first year was replaced by an open-source implementation of WWTBAM, a television quiz show that offers a top prize of $1 million for answering correctly successive MC questions of increasing difficulty (Figure 2).

Figure 2: "Who Wants To Be A Millionaire?" in-game screenshot

```
question - week2.txt
Which statement will check if a is equal to b?|1|if a == b:|if a = b:|if a === b:|if a equals b:|
Which statement tests if the user entered the letter Y?|3|if letter = "y":|if letter = "Y":|if letter == "Y":|if letter == "y":|
What is the error in this statement? if~score~=~21: print("You win")|4|Missing else statement|Missing elif statement|There is no error|Equality should be evaluated using
The boolean (bool) data type has exactly two values:|2|Yes / No|True / False| true / false|-1 / 1|
Rewrite the following algebraic expression to an equivalent Python expression: 32~<=~temp~<=~100|3|if temp <= 32 and temp <= 100|if temp <= 32 or temp <= 100|if temp >=
Which values make the following expression True? not~x~==~y~and~z~>~x|2|x = 10, y = 10, z = 15|x = 10, y = 20, z = 15|x = 10, y = 2, z = 5|x = 10, y = 20, z = 10|
What will be the output of the following code: token~=~False; while~token:~print("Hello")|3|"Hello" will continued to be displayed forever|No output because of compilati
What does a sentinel value do?|4|Creates a bridge between a data set and unrelated input|It's part of the data to be processed by the program|It terminates a program|It
What will be printed by this statement? for~num~in~range(10,-5,-3):~print(num, end=" ")|1|10 7 4 1 -2|10 7 4 1 -2 -5|10 5 0|7 4 1 -2|
How many times does this loop execute? for~i~in~range(50,~0,~-10):~print(i)|1|5|6|4|50|
Which of the following for loops will run the loop body 5 times?|2|for i in range(4, 0, -1):|for i in range(5, 0, -1):|for i in range(5, 1, -1):|for i in range(6, 0, -1)
Which code will generate a random integer from 1 up to and including 10 and store it in x?|3|x = randint(0, 10)|x = randint(0, 11)|x = randint(1, 10)|x = randint(1, 11)|
What range of numbers are generated by the random() function?|1|0 <= number < 1|0 < number < 1|0 < number <= 1|0 <= number <= 1|
What values does i assume when this loop executes? for~i~in~range(20, 2, -6):print(i,~end=",")|3|20,14,8,2|20,14,8,2,-4|20,14,8|14,8,2|
What is the value of i at the end of~the~code? i~=~1; while i < 32: i = i *2|3|16|31|32|64|

What is the opposite of~the~following~condition? count~>~10?|3|count >= 10|count < 9|count <= 10|count <= 9|
What statement is used to implement a decision?|2|while|if|for|import|
Which condition will cause the body of the if statement to execute when count is 0?|4|if count = 0:|if count < 0:|if count =< 0:|if count == 0:|
Which operators are considered boolean operators?|2|< >|and or|== !=|<= >=|
Given two variables x and y, how do you test whether exactly one of them is zero?|3|if x == 0 or y == 0|if x = 0 or y = 0|if x == 0 and y != 0 or y == 0 and x != 0|if x
Which statement is used to stop the execution of a looping statement?|1|break|stop|pause|exit|
What will be the output of the following~code: while~True:~print("Hello")|1|"Hello" will continued to be displayed forever|No output because of compilation error|No outpu
How many times does this code print "hello"? for~i~in~range(0,~10):~print("hello")|1|10|9|11|infinite loop|
Which of the following options checks that city is neither London nor Paris?|2|if not city == "London" or not city == "Paris"|if not (city == "London" or city == "Paris"
Which option refers to the technique of simulating program execution on a sheet of paper?|4|Compiling|Prototyping|Debugging|Tracing|
What will be printed by this statement? for~num~in~range(10,~5,~-1):~print(num, end=" ")|2|10 9 8 7 6 5|10 9 8 7 6|5 6 7 8 9 10|6 7 8 9 10|
What does this code print? for~x~in~range(1,~11,~2):~print(x + 1)|2|The odd numbers 3 to 11|The even numbers 2 to 10|The odd numbers 3 to 13|The even numbers 2 to 12|
What are the smallest and largest values my_float can take? my_float~=~random()~*~2~+~3|1|3.0, 5.0 (excluding 5.0)|0.0, 6.0 (excluding 6.0)|-3.0, 3.0 inclusive|0.0, 3.0
What is the value of j at the end of the code? j~=~0; for~i~in~range(0,~4):~j~+=~i|3|2|4|6|8|
Which statement will generate a random integer in the range -20 to 20 inclusive?|4|randrange(-20, 20)|randint(-20,21)|randrange(20) - 20|randrange(41) - 20|

The operator >= stands for:|2|greater than|greater than or equal to|not equal to|invalid operator|
What are the two parts of an if statement?|1|A condition and a body|A check and an increment|An increment and a body|An increment and a return value|
If num1 is 3 and num2 is 4, which condition is true?|4|num1 + 1 < b|num1 + 1 > b|num1 + num2 != 7|num1 - num2 <=0|
Which of the following variables is used to store a condition than can be either True or False?|1|Boolean|Logical|Algebraic|Conditional|
Suppose that b is False and x is 0. Which of the following expressions evaluates to True?|3|b or x == 1|b and x == 0|not b and x == 0|not b and x == 1|
```

Figure 3: A sample question file

The version of the game used in the classroom featured 540 Python-related MC questions (3 sets of 15 questions per week), which were created by the instructors through a straightforward process that required the editing of a simple text file (Figure 3).

For logistic purposes the class was randomly divided into four groups of 13 students (11 male, 2 female) who attended a separate seminar every week for a total of 12 weeks. During the first seminar each group was randomly split into three teams of 4-5 contestants that remained the same for the duration of the course, and then the gaming activity started as outlined below.

Each team was seated in front of the class facing the screen with their backs to the audience so that they could not receive any unsolicited assistance. Students were then asked 15 increasingly difficult questions on Python programming which covered a different topic every week. Since some of these questions were also scheduled to appear in the 6 theoretical quizzes, in fairness to the team of student contestants, all other students in the class were instructed to put away their note-taking materials for the duration of the game. This also enhanced the perception that the class was taking a break.

Although there was no official time limit to answer a question, each game's duration was limited to 20 minutes in order to give all teams the opportunity to play once during the seminar. Questions were multiple-choice: 4 possible answers were given and the team had to collaborate, reach a consensus, and give a single response. Additionally, at the beginning of each game contestants were presented with an aid of three lifelines:

- Poll The Class: All students provided their answers for a particular question by raising their hands and the percentage of each specific option as chosen by the class was displayed to the contestants.
- 50/50: The game eliminated two incorrect answers, thus leaving contestants with one incorrect and the correct answer to choose from.
- Ask A Friend: Contestants had 30 seconds to read the question and answer choices to a non-team classmate, who in turn had the remaining time to offer input.

After viewing a question, the team could respond in one of three ways:

- Refuse to answer the question, quit the game, and retain all points earned up to that point.
- Answer the question and, if their answer was correct, earn points and continue to play, or lose all points earned to that point and end the game if incorrect. However, the $5,000 and $100,000 prizes were guaranteed: if a team got a question wrong above these levels, then the prize dropped to the previous guaranteed prize.
- Use a lifeline (Ask A Friend, Poll The Class, or 50/50).

The game ended when the contestants answered a question incorrectly, decided not to answer a question, or answered all questions correctly (Figure 4). All answers to each question were conscientiously reviewed for the entire class as the game proceeded. This discussion of the relative merits of the various provided answers was an integral part of the learning process that took place during the execution of the game.

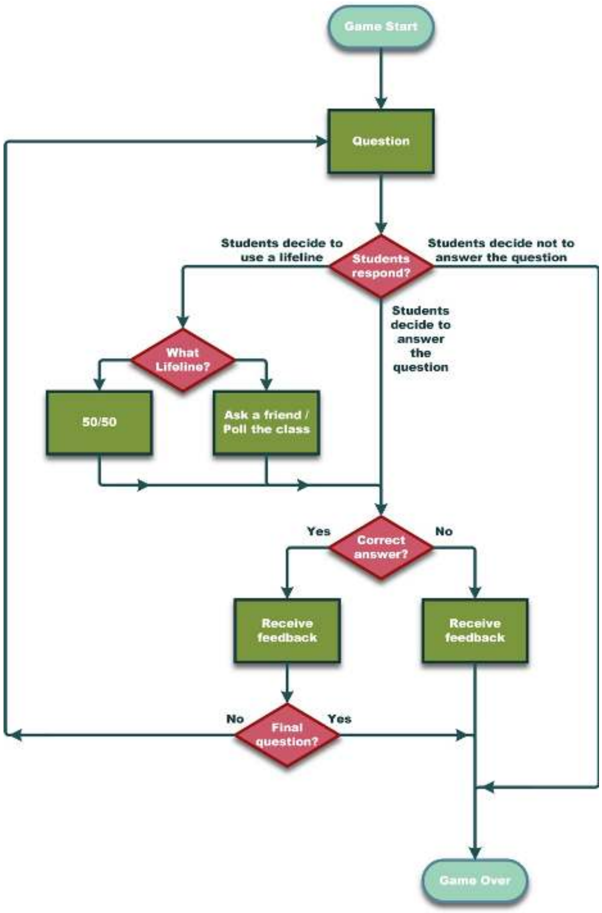

Figure 4: "Who Wants To Be A Millionaire?" game procedure

At the end of every seminar, newly earned points were added to the points carried from previous weeks. The whole scoring process was done manually, with points being collected by faculty and then added to a leaderboard webpage on Blackboard, which showed the team rankings for every group and provided an entry point to the gamified experience. After the twelve seminars were completed, the leading team won the title of "Pythonista of the year" and received chocolate bars as an award. Finally, in order to promote self-assessment and allow students who missed the seminar sessions to experience this alternative form of learning, the game and its latest set of questions became available for download at the end of every week.

3.2.3 Practising programming skills with Codecademy

Founded in 2011, Codecademy offers free coding courses tailored for the new computing syllabus in the UK in a number of programming languages, including Python, JavaScript, HTML/CSS, jQuery, Ruby, and PHP. Additionally, it serves as a competitive virtual classroom that allows students to track their peers' achievements and work to match or outdo them. The programming courses are organised into sections containing a series of interconnected exercises which in turn include an educational text introducing the related topic, instructions that tell students what to do, and the actual interactive exercise to be completed. Students earn points for completing each exercise and every completion of a lesson is registered as an achievement. Other achievements include the maximum number of points earned in one day, the maximum number of days a student logs-in in a row etc. Badges are also awarded for attaining specific number of points, exceeding a streak length, or completing certain lessons or courses (Swacha and Baszuro, 2013). These gamification features have been crucial to making Codecademy one of the most popular online education providers with over 24 million users to date (Richard Ruth, 2015) and were the main reason behind selecting it as the delivery platform for the programming exercises.

In the first laboratory session instructors created an “FSD Class” containing 36 lessons of Codecademy’s Python track that were mapped to the syllabus of FSD. Students were then asked to sign up and create a pupil account, which was used to enrol them to the FSD class. From that point lab sessions proceeded as follows: every weekly session began with a five-minute introduction to the exercises for the day, and then students were required to complete a certain number of Codecademy lessons based on the topics that had been covered until then. Each lesson was broken down into bite-sized chunks and comprised practical exercises accompanied by notes that explained the programming techniques and terms used. After reading the exercise instructions, students would type in their Python code to the code window, submit their code for execution, and see its output in a separate window (Figure 5). If the code were erroneous, they would receive an error message and would have to try again. Once they managed to solve the exercise, they would earn points and proceed to the next lesson. Students who were not able to finish on time could continue the lessons independently and at their own pace at home, while students who finished early and wished to further their programming skills were provided with additional exercises.

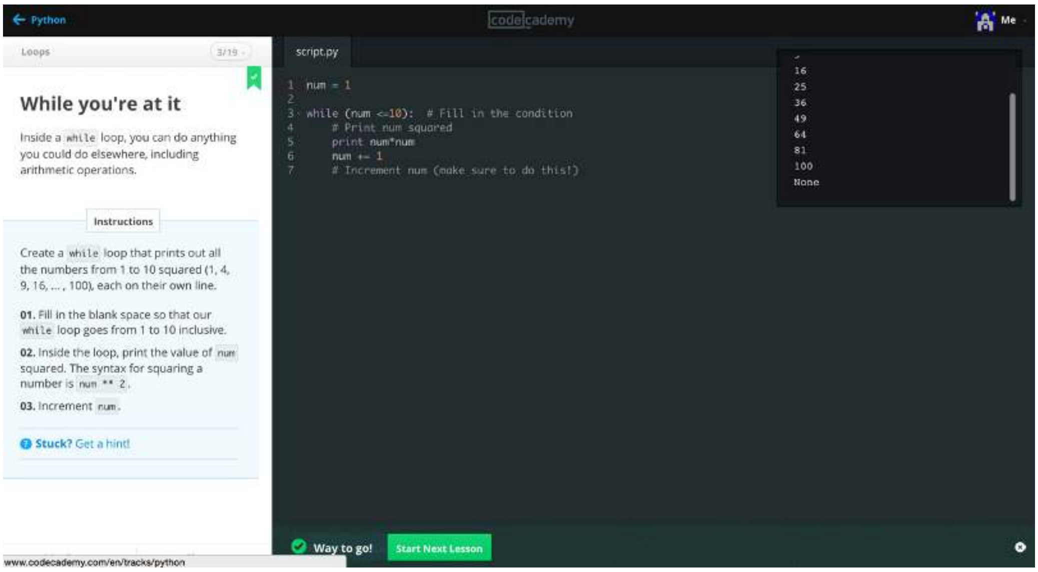

Figure 5: Codecademy’s lesson screen

The Codecademy platform provided students with direct feedback on their progression via graphical representations such as completion indicators for each lesson and for the overall course, badges and points for various achievements etc. (Figure 6). This served as instant gratification and offered an added dimension to learning, as students could track their peers’ scores and try to surpass them. Additionally, Codecademy’s “Pupil Tracker” feature allowed instructors to track student progress, including percentage completion, badges, and last log-in dates, as well as to measure students’ courses and tracks in comparison to one another (Figure 7).

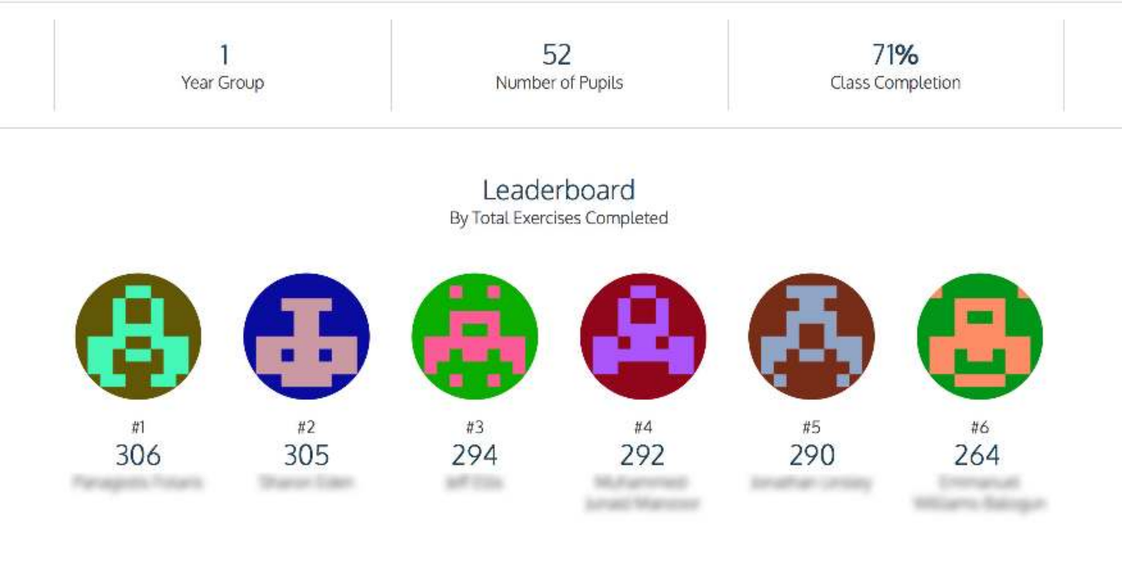

Figure 6: Codecademy's leaderboard

Figure 7: Codecademy's Pupil Tracker

In an effort to motivate students to complete the exercises as quickly as possible, the lecturers set a number of different challenges, e.g., highest score achieved in 1 and in 4 weeks, fastest student to reach 50, 100, and 200 points etc. However, no actual physical rewards were given to the winners. The rationale for this decision was to allow faculty staff to evaluate whether the aim of winning a challenge was in itself enough as intrinsic motivation for students to complete their tasks. Each challenge had its own leaderboard, which was made accessible to the students through Blackboard. At the end of each week, staff used the Pupil Tracker to download the spreadsheet with the students' progression and updated the leaderboards accordingly. The devised challenges motivated the majority of the students to perform on a weekly basis, thus engaging them with programming activities throughout the semester.

4 Results

To ensure that the gamified approach encouraged students' active participation in the educational process, formative and summative assessments of student engagement were performed using the following methods (Jennings and Angelo, 2006):

- Observation of student behaviour;
- Online survey exploring the effects of gamification in the classroom;
- Students' self-report of activity through focus groups and semi-structured interviews;
- Collection of administrative data such as student attendance, late arrivals to class, number of reference material downloads, completion rate of lab exercises, and academic performance.

4.1 Observation of student behaviour

In regards to classroom observation of student behaviour, the majority of the EC students demonstrated the following characteristics during all seminars, lectures, and laboratory sessions, which are considered immediate indicators of engagement (Franklin, 2005; Mandernach et al., 2011):

- Actively listened, focused attention and made eye contact;
- Responded to the instructors' prompts;
- Actively participated in the WWTBAM and Kahoot! games, and in the Codecademy challenges;
- Questioned, explored, brainstormed or discussed the WWTBAM and Kahoot! question topics with their peers and instructors;
- Utilised decision-making or problem solving skills in questioning and responding;
- Demonstrated body language that was open and relaxed with appropriate smiles or laughter.

4.2 Online survey exploring the effects of gamification in the classroom

To gather quantitative feedback about the effectiveness of the gamified experience, all EC students ($N_{exp} = 52$) completed a 15-question online survey at the end of the semester. Every question had 5 possible answers measured on a Likert scale of 1 (Strongly Disagree) to 5 (Strongly Agree).

#	Question	Disagree 1	2	... - ... 3	4	Agree 5	Aver.	Var.	Dev.	Med.
1	The games made the learning environment a fun and engaging one.	0.0%	0.0%	9.6%	13.5%	76.9%	4.7	0.4	0.6	5
2	The games motivated me to attend classes.	0.0%	0.0%	48.1%	25.0%	26.9%	3.8	0.7	0.8	4
3	The games motivated me to arrive to class on time.	0.0%	9.6%	36.5%	26.9%	26.9%	3.7	1.0	1.0	4
4	I was more motivated to study the course material every week in order to do well in the leaderboard for the games.	0.0%	0.0%	34.6%	50.0%	15.4%	3.8	0.5	0.7	4
5	I communicated with other players while playing.	0.0%	0.0%	9.6%	15.4%	75.0%	4.7	0.4	0.7	5
6	The total duration of the games was satisfactory.	0.0%	0.0%	11.5%	36.5%	51.9%	4.4	0.5	0.7	5
7	I was comfortable with adding the Top-5 leaderboard to the module's Blackboard page.	0.0%	0.0%	34.6%	25.0%	40.4%	4.1	0.8	0.9	4
8	The discussions about the correct and incorrect answers after every question (i.e., why wrong answers were wrong, and right answers were right) were satisfying.	0.0%	0.0%	11.5%	34.6%	53.8%	4.4	0.5	0.7	5
9	I believe that the games have improved my understanding of the covered topics.	0.0%	0.0%	23.1%	48.1%	28.8%	4.1	0.5	0.7	4
10	Performing well in the games increased my self-confidence.	0.0%	0.0%	9.6%	50.0%	40.4%	4.3	0.4	0.6	4
11	I would have prepared and engaged better if the game results were translated to actual marks for the module assessment.	0.0%	23.1%	36.5%	25.0%	15.4%	3.3	1.0	1.0	3
12	I believe that the games have improved my analytical and problem-solving skills in terms of developing solutions for Python challenges.	0.0%	9.6%	34.6%	26.9%	28.8%	3.8	1.0	1.0	4
13	I wish Kahoot! and "Who Wants To Be A Millionaire" were used in other modules.	0.0%	0.0%	32.7%	26.9%	40.4%	4.1	0.7	0.9	4
14	I believe that gaming is a valuable use of instructional time	0.0%	0.0%	30.8%	32.7%	36.5%	4.1	0.7	0.8	4
15	I found the use of the leaderboard intimidating.	53.8%	25.0%	21.2%	0.0%	0.0%	1.7	0.7	0.8	1

Figure 8: Online survey results

According to the weighted Likert scale average shown in Figure 8, students mostly agreed that the classroom games made learning fun and would like to see them introduced to other modules as well. Students were also generally motivated to attend classes and arrive on time, a finding that was also supported by the administrative data collected at the end of the course. Most students communicated with their peers while playing and believed that performing well in the games increased their self-confidence. Additionally, they were not intimidated by the use of leaderboards and some of them even studied the course material on a weekly basis in order to appear high in the leaderboard rankings. The discussions about the correct and incorrect answers after every Kahoot! and WWTBAM question were deemed satisfying and improved the students' understanding of the cover topics. Surprisingly enough, there were mixed opinions about getting some tangible rewards, such as translation of game points into actual marks for module assessments. Finally, most students considered gaming a valuable use of instructional time as they felt it helped them improve their analytical and problem-solving skills.

4.3 Semi-structured interviews for in-depth student feedback

To get extra insight into the survey results, qualitative research was conducted in the form of focus groups and semi-structured interviews with a small number of students, featuring questions on collaborative learning, cognitive development, and development of personal skills. As demonstrated by the following sample of responses, the overall reaction by interviewees was extremely positive:

- "I know that I have learned from watching other people play WWTBAM, as well as through playing myself."
- "I feel great when I know all the answers. Bragging rights are a plus, too."
- "It makes you feel like you've learnt something when you complete a lesson in Codecademy."
- "Seeing my name at the top of the leaderboard made me feel smart and proud."
- "Although I am rather shy and quiet as a person, playing WWTBAM boosted my confidence and made it easier for me to collaborate with my classmates."
- "At last I was allowed to use my iPhone in the class, even if it was for educational purposes."
- "I enjoy Kahoot! because it's always fun to beat your classmates."
- "Lectures don't feel boring anymore."

4.4 Analysis of the administrative data

As a means of gauging student persistence, interest, and effort in the gamified classes, there was a comparison of the attendance and the late arrivals (students arriving to class with at least a 10-minute delay) among the control and the experimental classes (Figure 9).

Average class attendance for CC was 65% (≈35 students), while EC had an average class attendance of 78% (≈42 students). Additionally, an average of 4 to 5 CC students and 1 to 2 EC students arrived to class late every week, respectively. Both findings suggest that gamification motivated EC students to be more punctual and attend classes more often than their CC peers.

In regards to the reference material, every week the instructors uploaded two compressed files: the first one contained the lecture notes and handouts, and the second one contained further reading material (book chapters, journal papers, selected articles, blog posts, and other optional readings). As documented in the number of the reference material's weekly downloads, it can be argued that CC students demonstrated a relative lack of interest with an average of 1.2 weekly file downloads per student (an average of 65 total downloads per week), which spiked only during the two weeks preceding the final exams; in comparison, every EC student downloaded 1.7 files every week (an average of 89 total downloads per week) without showing any significant deviations (Figure 10). When combined with the survey's results, this could suggest that EC students were motivated to download and study the course and further reading material every week in order to perform well in the classroom games.

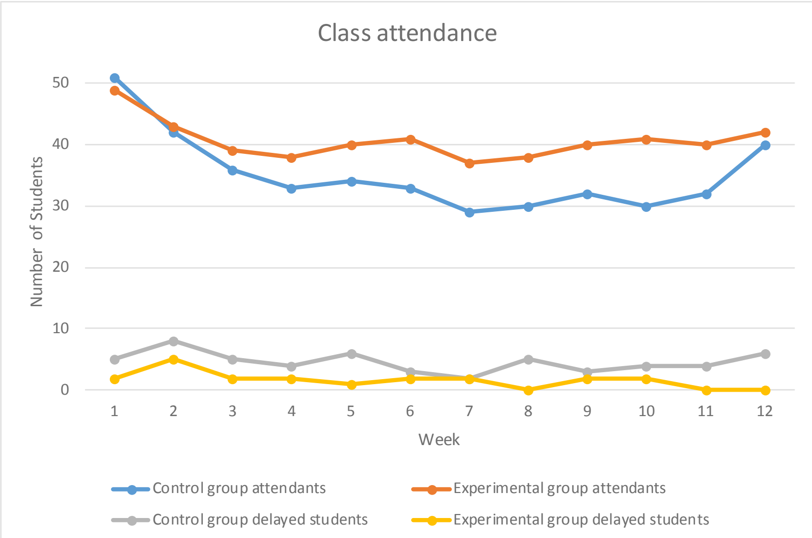

Figure 9: Weekly class attendance

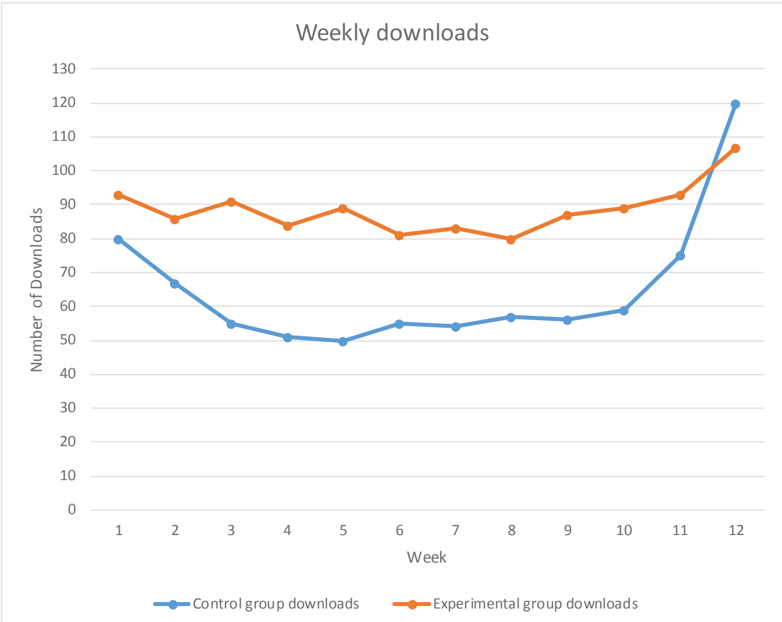

Figure 10: Weekly downloads of reference material

While the CC completion rate of the practical exercises remained roughly around the 50% mark for every laboratory class, EC students showed a small but steady weekly increase in their completion rate, which might indicate that the weekly challenges motivated them to try harder so as to complete their exercises and improve their programming skills. Finally, EC had the best overall academic performance with an average final grade of 61% compared to CG's 53%. However, due to the relatively low number of participants, additional studies are needed to identify possible correlations between gamification and academic performance.

5 Conclusion and future work

The present study explored how the application of gamification in a computer-programming course could affect the learning experience and the students' motivation, recall ability, and performance. The aforementioned findings suggest that using a multi-dimensional gamified learning approach has successfully achieved the pedagogical goals outlined in the introduction. Based on the concepts of the increasingly popular gamification, game-based learning and serious games movements, it gives teachers and students the opportunity to experience first-hand how game mechanics can be used to make learning fun and addictive. Coupled with effective pedagogy, games can offer a more effective and less intrusive measurement of learning than traditional assessments.

Both Kahoot! and WWTBAM serve as an opportunity for instant application of knowledge and reinforcement of learning outcomes. They allow common programming language misconceptions to be revealed and explored, while also using similar game mechanisms to make students feel good about their accomplishments and overcome their personal records.

More specifically, Kahoot! provides students with the opportunity for self-assessment through a fun and engaging atmosphere, which allows them to master new programming concepts relatively quickly. It is a great tool for learning terminology and can be also used to introduce a topic, as it can help instructors discover what the students already know and where they should focus their instruction on. Additionally, the findings are comparable to those from other studies which show that the use of CRS increases students' attendance, attentiveness, enthusiasm, confidence, and in-class participation (Duncan, 2005; Suchman et al, 2006; Bullock et al. 2003; Roschelle et al., 2004; Wit, 2003). As for WWTBAM, it requires students to compare and discuss their answers with their teammates in order to come to a consensus regarding the answer, thus improving communication efficiency and honing important employability skills such as problem solving, critical thinking, and collaboration. In both games students not only reported more enjoyment in their class, but also stated that confidence in their own learning had grown, while instructors noticed an increase in their own ability to respond to students' misconceptions.

This mix of individual and group competition in the classroom catered to the needs of diverse students, some of which preferred to initially develop their coding skills alone while others performed better in groups. As the semester progressed though, it was noticed that the students' engagement decreased slowly in the Kahoot! sessions; on the other hand, the engagement for WWTBAM remained unchanged. This could be attributed to the fact that students competing at individual level in Kahoot! began to lose interest once they trailed behind in the leaderboard. Another concern from the teaching staff's point of view was the limited length of the multiple-choice questions and answers in both games, which made their authoring quite challenging.

The use of Codecademy's points and badges as the sole motivator for completing the practical exercises also provided some interesting insights. Although students were intrinsically motivated to complete their exercises and generally performed better than their CC peers, they expressed some concerns about the lesson contents, saying that some lessons were not always a good fit to the FSD syllabus, lacked clear instructions, and had ambiguous explanations and vague error messages. As a result, students who struggled on a particular aspect of programming due to the poor quality of that particular set of lessons tended to associate that aspect with being difficult to grasp and master, when it was not necessarily so. A possible yet rather demanding solution to this problem would be to provide students with a more personalised experience by developing lessons specifically for the FSD syllabus. Additionally, data analytics could be used to identify which programming concepts are more challenging for students, so as to give the latter opportunities for more practice.

Whilst the results are encouraging, the authors acknowledge that the limited nature of this study does not preclude the possibility that the improvements in student engagement are simply the result the short-term "novelty" factors generally associated with the introduction of new technology and learning techniques. Further study is needed to assess whether the increased student engagement suggested by these methods is sustainable and applicable to other subjects.

References

Ahn, L. von and Dabbish, L. (2008) "Designing games with a purpose", *Communications of the ACM*, vol. 51, no. 8, pp. 58-67.

Annetta, L.A., Minogue, J., Holmes, S.Y. and Cheng, M.T. (2009) "Investigating the impact of video games on high school students' engagement and learning about genetics", *Computers & Education*, vol. 53, pp. 74-85.

Barata, G., Gama, S., Jorge, J. and Goncalves, D. (2013) "Engaging Engineering Students with Gamification", in *5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, Bournemouth, UK, Sep 11-13 2013, pp. 1-8.

Blythe, M., Hassenzahl, M. and Wright, P. (2004) "Introduction: Beyond fun", *Interactions*, vol. 11, no. 5, pp. 36–37.

Borys, M. and Laskowski, M. (2013) "Implementing game elements into didactic process: A case study", in *Management, Knowledge and Learning International Conference*, Croatia, 19-21 Jun 2013, pp. 819–824.

Britain, S. and Liber, O. (2004) *A framework for pedagogical evaluation of virtual learning environments. Research Report*, [Online], Available: https://hal.archives-ouvertes.fr/hal-00696234/document [10 Dec 2015].

Bullock, D., LaBella, V., Clingan, T., Ding, Z., Stewart, G. and Thibado, P. (2003) "Enhancing the student-instructor interaction frequency", *The Physics Teacher*, vol. 40, no. 9, pp. 535-541.

Carroll, J. M. (1982) "The Adventure of Getting to Know a Computer", *Computer*, vol. 15, no. 11, pp. 49-58.

Carroll, J. M. and Thomas, J. M. (1988) "FUN", *ACM SIGCHI Bulletin*, vol. 19, no. 3, pp. 21-24.

Chao, D. (2001) "Doom as an interface for process management", in *Proceedings CHI '01*, New York: ACM Press, pp. 152-157.

Chen, J. (2007) "Flow in games (and everything else)", *Commun. ACM*, vol. 50, pp. 31-34.

Coller B. and Shernoff, D. (2009) "Video game-based education in mechanical engineering: A look at student engagement", *International Journal of Engineering Education*, vol. 25, no. 2, pp. 308-317.

Corbett, S. (2010) *Learning by playing: Video games in the classroom*, [Online], Available: http://www.nytimes.com/2010/09/19/magazine/19video-t.html [10 Dec 2015].

Csikszentmihalyi, M. (1990) *Flow: the psychology of optimal experience*, New York: Harper and Row.

de Aguilera, M. and Mendiz, A. (2003) "Video games and education: education in the face of a "parallel school"", *Computers in Entertainment*, vol. 1, no. 1, pp. 1:1-1:10.

Deterding, S., Dixon, D., Khaled, R. and Nacke, L. (2011) "From game design elements to gamefulness: defining "gamification"", in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (MindTrek '11),* Tampere, Finland, Sep 28-30 2011, ACM Press, pp. 9-15.

Duncan, D. (2005) *Clickers in the classroom*, Addison: San Francisco, CA: Addison.

Fies, C. and Marshall, J. (2006) "Classroom Report Systems: A Review of the Literature", *Journal of Science Education and Technology*, vol. 15, no. 1, pp. 101-109.

Franklin, E.E. (2005) "Assessing teaching artists through classroom observation", *Teaching Artist Journal*, vol. 3, pp. 148-157.

Gee, J.P. (2003) *What Video Games Have to Teach Us About Learning and Literacy*, New York: Palgrave Macmillan.

Gee, J.P. (2008) "Learning and games", in Katie Salen (ed.). *The ecology of games: Connecting youth, games, and learning* (John D. and Catherine T. MacArthur Foundation series on digital media and learning), Cambridge, MA: The MIT Press.

Green, C.S. and Bavelier, D. (2006) "Effect of action video games on the spatial distribution of visuospatial attention", *Journal of Experimental Psychology: Human Perception and Performance*, vol. 32, no. 6, pp. 1465-1478.

Griffin, A. (2014) "Technology Distraction in the Learning Environment", in *Proceedings of the Southern Association for Information Systems Conference*, Macon, GA, USA Mar 21-22, 2014, [Online], Available: http://saisconferencemgmt.org/proceedings/2014/Griffin.pdf [10 Dec 2015].

Hartley, J. and Davies, K. (1978) "Note Taking: A Critical Review", *Programmed Learning and Educational Technology*, vol. 15, pp. 207-224.

Hassenzahl, M. (2003) "The Thing and I: Understanding the Relationship Between User and Product", in M.A. Blythe, K. Overbeeke, A.F. Monk and P.C. Wright (eds.), *Funology: From Usability to Enjoyment*, New York: Kluwer, pp. 31-41.

Huang, W.H-Y. and Soman, D. (2013) *A Practitioner's Guide To Gamification Of Education* [Online], Available: http://inside.rotman.utoronto.ca/behaviouraleconomicsinaction/files/2013/09/GuideGamificationEducationDec2013.pdf [10 Dec 2015].

Hwang, I., Wong, K., Lam, S. L. and Lam, P. (2015) "Student Response (clicker) Systems: Preferences of Biomedical Physiology Students in Asian classes", *The Electronic Journal of e-Learning*, vol. 13, no. 5, pp. 319-330. [Online], Available: http://www.ejel.org/volume13/issue5 [23 Jan 2016]

IBM Global Business Services (2012) *Online simulation gaming comes of age for employee skill development at IBM* [Online], Available: http://public.dhe.ibm.com/common/ssi/ecm/en/gbc03106usen/GBC03106USEN.PDF [10 Dec 2015].

Jennings, J.M. and Angelo, T. (eds.) (2006) "*Student engagement: Measuring and enhancing engagement with learning"*, Proceedings of a symposium, Universities Academic Audit Unit, New Zealand.

Jordan, P.W. (2002) *Designing Pleasurable Products. An introduction to the new human factors*, London: Taylor & Francis.

Kebritchi, M., Hirumi, A. and Bai, H. (2008) "The effects of modern math computer games on learners' math achievement and math course motivation in a public high school setting", *British Journal of Educational Technology*, vol. 38, no. 2, pp. 49-259.

Koster, R. (2004) *A theory of fun*, New York: Paraglyph Press.

Lahtinen, E., Ala-Mutka, K. and Jarvinen, H. (2005) "A Study of Difficulties of Novice Programmers", *Innovation and Technology in Computer Science Education 2005*, pp. 14–18.

Laster, J. (2010) *At Indiana U., a class on game design has students playing to win*, [Online], Available: http://chronicle.com/blogs/wiredcampus/at-indiana-u-a-class-on-game-design-has-students-playing-towin/21981 [10 Dec 2015].

Lee, J.J. and Hammer, J. (2011) "Gamification in Education: What, How, Why Bother?", *Academic Exchange Quarterly*, vol. 15, no. 2, pp. 1-5, [Online], Available: http://www.gamifyingeducation.org/files/Lee-Hammer-AEQ-2011.pdf [10 Dec 2015].

Lee, J.J. Luchini, K., Michael, B., Norris, C. and Soloway, E. (2004) "More than just fun and games: assessing the value of educational video games in the classroom", in *CHI '04 Extended Abstracts on Human Factors in Computing Systems, ser. CHI EA '04*, New York: ACM Press, pp. 1375–1378.

Li, W., Grossman, T. and Fitzmaurice, G. (2012) "GamiCAD: A Gamified Tutorial System For First Time AutoCAD Users", in *Proceedings of the 25th annual ACM symposium on User interface software and technology UIST '12*, Cambridge, MA, USA, Oct 7-10 2012, pp. 103-112, [Online], Available: http://www.autodeskresearch.com/pdf/GamiCAD_CameraReady.pdf [10 Dec 2015].

Malone, T.W. (1980) "What makes things fun to learn? Heuristics for designing instructional computer games", in *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems – SIGSMALL '80*, New York: ACM Press, pp. 162–169.

Malone, T.W. (1981) "Toward a theory of intrinsically motivating instruction", *Cognitive Science*, vol. 4, pp. 333-370.

Malone, T.W. (1982) "Heuristics for designing enjoyable user interfaces", in *CHI '82 Proceedings of the 1982 Conference on Human Factors in Computing Systems*, New York: ACM Press, pp. 63-68.

Mandernach, B. J., Donnelli-Sallee, E. and Dailey-Hebert, A. (2011) "Assessing Course Student Engagement", in Miller, R.L., Amsel, E., Kowalewski, B., Beins, B., Keith, K. and Peden, B. (eds.), *Promoting student engagement, volume 1: Programs, techniques and opportunities*, Syracuse, NY: Society for the Teaching of Psychology, [Online], Available: http://www.teachpsych.org/teachpsych/pnpp/ [10 Dec 2015].

McClean, P., Saini-eidukat, B., Schwert, D., Slator, B. and White, A. (2001) "Virtual worlds in large enrollment science classes significantly improve authentic learning", in *Proceedings of the 12th International Conference on College Teaching and Learning*, Center for the Advancement of Teaching and Learning, pp. 111–118.

McGonical, J. (2010) *Gaming can make a better world*, [Online], Available: http://www.youtube.com/watch?v=dE1DuBesGYM [10 Dec 2015].

McKeachie, W.J. (1999) *McKeachie's Teaching Tips: Strategies, Research, And Theory For College And University Teachers*, Boston, MA: Houghton-Mifflin.

Middendorf, J. and Kalish, A. (1996) "The ''change-up'' in lectures", *The National Teaching and Learning Forum*, vol. 5, no. 2, pp. 1-7.

Muntean, C.I. (2011) "Raising engagement in e-learning through Gamification", in *Proceedings of the 6th International Conference on Virtual Learning ICVL 2012,* pp. 323–329.

Natvig, L., Line, S. and Djupdal, A. (2004) "'Age of computers'; an innovative combination of history and computer game elements for teaching computer fundamentals", in *Proceedings of the 34th Annual Frontiers in Education conference*, ser. FIE 2004, vol. 3, pp. S2F–1–6.

NMC Horizon Report > Higher Education Edition (2013) [Online], Available: http://www.nmc.org/pdf/2013-horizon-report-HE.pdf [10 Dec 2015].

NMC Horizon Report > Higher Education Edition (2014) [Online], Available: http://www.nmc.org/pdf/2014-nmc-horizon-report-he-EN.pdf [10 Dec 2015].

Olsson, M., Mozelius, P. and Collin, J. (2015) "Visualisation and Gamification of e-Learning and Programming Education", *The Electronic Journal of e-Learning*, vol. 13, no. 6, pp. 441-454. [Online], Available: http://www.ejel.org/volume13/issue6 [23 Jan 2016]

Prensky, M. (2001) *Digital Game-Based Learning*, New York: McGraw-Hill.

Premkumar, K. and Coupal, C. (2008) "Rules of engagement-12 tips for successful use of 'clickers' in the classroom", *Medical Teacher*, vol. 30, no. 2, pp. 146-149.

Robins, A., Rountree, J. and Rountree, N. (2003) "Learning and Teaching Programming: A Review and Discussion", *Computer Science Education*, vol. 13, no. 2, pp. 137-172.

Roschelle, J., Penuel, W. R. and Abrahamson, L. (2004) "Classroom response and communication systems: Research review and theory", in *Annual Meeting of the American Educational Research Association*, San Diego, CA, pp. 1–8.

Ruth, R. (2015) "Codecademy is teaching the world programming, for free", Startup Hook [Online], Available: http://startuphook.com/employment-2/codecademy-teaching-world-programming-free/498/ [10 Dec 2015].

Ryan, R. and Deci, E.L. (2000) Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions, *Contemporary Educational Psychology*, vol. 25, no. 1, pp. 54-67.

Sheldon, L. (2011) *The Multiplayer Classroom: Designing Coursework as a Game*, Boston, MA: Course Technology PTR.

Squire, K.D. (2003) "Video games in education", *International Journal of Intelligent Games & Simulation*, vol. 2, no. 1, pp. 49-62.

Squire, K.D., Barnett, M., Grant, J.M. and Higginbotham, T. (2004) "Electromagnetism supercharged!: learning physics with digital simulation games", in *Proceedings of the 6th international conference on Learning sciences, ser. ICLS '04*, International Society of the Learning Sciences, pp. 513–520.

Suchman, E., Uchiyama, K., Smith, R. and Bender, K. (2006) "Evaluating the impact of a classroom response system in a microbiology course", *Microbiology Education*, vol. 7, p. 3.

Swacha, J. and Baszuro, P. (2013) Gamification-based e-learning Platform for Computer Programming Education. Proceedings of the 10th IFIP World Conference on Computers in Education, July 1-7 2013, Torun, Poland, pp.122-130. [Online], Available: http://wcce2013.umk.pl/publications/v1/V1.14_125-Swacha-fullR-FPR.pdf [1 Feb 2016].

Wigfield, A., Guthrie, J.T., Tonks, S. and Perencevich, K.C. (2004) "Children's motivation for reading: Domain specificity and instructional influences", *Journal of Educational Research*, vol. 97, pp. 299-309.

Wimmer, G.E., Braun, E.K., Daw, N.D. and Shohamy, D. (2014) "Episodic Memory Encoding Interferes with Reward Learning and Decreases Striatal Prediction Errors", *The Journal of Neuroscience*, vol. 34, no. 45, pp. 14901-14912.

Winslow, L.E. (1996) "Programming pedagogy – a psychological overview", *SIGCSE Bulletin*, vol. 28, pp. 17-22.

Wit, E. (2003) "Who wants to be. . . the use of a personal response system in statistics teaching", *MSOR Connections*, vol. 3, no. 2, pp. 14–20.

Zaha, A., Nichols, J. and Deichman, A. (2014) "Gaming and Gamification in Education (K-12)", *Emerging Technologies for Education Change (EDPE640) – Gaming Newsletter*, [Online], Available: https://edpe640.wikispaces.com/file/view/EDPE640%20-%20Gaming%20Newsletter.pdf/530585356/EDPE640%20-%20Gaming%20Newsletter.pdf [10 Dec 2015].

Zhang, P. (2008) "Motivational Affordances: Reasons for ICT Design and Use", *Communications of the ACM*, vol. 51, no. 11, pp. 145-147.