Introduction to Programming through Video Game Creation

Brendan McGuinness (B00252443), [b00252443@studentmail.uws.ac.uk](mailto:b00252443@studentmail.uws.ac.uk)

Daniel Taylor (B00240424), [b00240424@studentmail.uws.ac.uk](mailto:b00240424@studentmail.uws.ac.uk)

Martin Wojtczak (B00253233), [b00253233@studentmail.uws.ac.uk](mailto:b00253233@studentmail.uws.ac.uk)

University of the West of Scotland, Paisley, Scotland

# Abstract

In this paper we review current literature on the topic of using serious games as a method to combat common issues faced with introducing people into programming. In the literature review we look at the type of games designed and developed to combat this issue, how feedback and assessment to the user was handled and the results gained from these studies. We also carried out a short survey to the games target audience in order to gain a better understanding of what users would like from a game to teach programming. Finally we propose a game design and assessment system we feel improves upon the current games described in the review literature and combines users’ needs and wants from the survey.

**Keywords:** Serious games, Programming, Language, Video Game, Introduction, Education.

# Introduction

Serious games is a new typology of digital games that is primarily designed for education and training with a pedagogical goal. Studies by Carly et al (2010) & Bradlyn et al (2008) in the USA and Europe attest to the valuable contributions of game-based approaches in education

According to Backlund et al (2007) Over the last 10 years serious game have become a common feature in a wide array of fields; healthcare, education, defence and security. This is because organisations such as Deloitte, IBM and Cisco are starting to embrace serious games. These companies are having game simulations or educational games made specifically to test and train their employees, claims Donovan (2012).

Sometimes serious games are confused with entertainment games. For example, the game *Trauma Center*: Under the Knife has a serious element to it where the player is in a surgery operating on patients. This has certain references to real life surgery but the game was not designed with the explicit purpose to train students how to learn surgical techniques Alvarez (2011).

However, Alvarez et al (2011) tells us about a serious game that teaches the student how to perform a total knee arthroplasty. This game is used in conjunction with other traditional learning material including books and videos. The end goal is for the user to successfully complete a total knee arthroplasty. This is what is expected of a more traditional serious game.

Serious games have become a genre of its own in the game industry, with mobile educational games in the USA making $227.9 million in 2013. Greer (2013) claims this statistic is expected to almost double in the next few years, with $410 million being the predicted annual revenue by 2018. The forecast predicts the worldwide market will also be growing. According to Greer (2014) Game-based learning will grow from $1.7billion in 2013 to $2.4 billion in 2017. The whole serious games market will also be growing, $3 billion was taken in 2013 and it is predicted that the number will grow to $7.1 billion by 2018.

Phantom Tech are creating a serious game that helps the user learn to code in C#. The games objective is to teach the user how to code in simple steps and in the end create a simple Shoot 'em' up game. The game will take the user on a slow learning curve and explain to the user what the exact purpose of what each function and variable is. The user will then slowly create a Shoot 'em' up game. The user will be given simple instructions on the side of the screen, the user will then follow the instructions and write simple C# code into the console. The code will compile and show the user the end result. If the user has made a mistake the game will show the user where they have went wrong and show them how to fix it. After following all the steps the user will then eventually have a Shoot 'em' up game that they can play. The purpose of this game is help the user understand the basic fundamentals of coding in C#. This game will be a great introduction to young students who have an interest in creating games or coding.

In this paper, the serious games that teach coding will be identified and analysed. The literature on game based learning will be examined and the paper will then discuss what is missing from the serious games that are in the same genre and theme.

The paper will explain in detail how the team propose to make the application. It will continue on to introduce how the assessment will be carried out. The paper will end with a detailed explanation into the reasoning behind the project.

The purpose of the paper is to show an understanding of the serious games field. Identify a problem that we have discovered in the field. Create a solution to the problem and convey an intelligent observation to support and prove our solution.

We hope this paper will lead to a clearer understanding of the serious game field by showing how serious games can introduce programming through video game creation.

The problem that is being addressed is that there is a lack of quality in terms of games that introduce programming through video game creation. We plan to show the reader the mistakes that have been made previously in the field and show how Phantom tech intend to correct these mistake in our project.

Some of the main problems are that the games that teach programming don’t actually assess the user. We also found that some games don’t evaluate how their game helps users learn to program and if users enjoy programming Shoot 'em' up games compared to programming simple functions.

# Previous Work

There are many other instance of developers using game creation as a basis to teach programming, on example of this is Ouabhi, Kaddari et al (2015, p.1480) *“We’ve use a new pedagogical teaching approach with the first sub-group based on the creation of games by students using the Scratch environment”*. Ouabhi, Kaddari et al, (2015) used three groups of students where group “A” would create a game using the scratch environment, with groups “B” and “C” being taught with a conventional approach to teaching programming. In their results Ouabhi, Kaddari et al (2015, p.1481) found that less students in group “A” found the course boring compared to groups “B” and “C”, *“Only15% of students from sub-groups “A” found that programming with the Scratch environment is boring but in the other sub groups “B” and “C”, 79.3 % of students found it monotonous and boring to program with a conventional environment.”*.

During their study Ouabhi, Kaddari et al made no mention of how they assessed their players or if learning to create a game instead of programming in a conventional style did actually improve students’ knowledge and or results. In our game we plan to keep track of the amount of attempts it takes a player to complete a task or section of the game in order to highlight to them or teach the needed improvement in this area, rather than just keeping track of their enjoyment and engagement in the course.

Coelho, Goncalves et al (2011) was to create an open ended game where teachers would be able to set the questions and programming language used in the game, (Coelho, Goncalves et al, 2011, p.63) *“Teachers should be able to customize the game design to adapt it to their courses.”.* Leaving the game open to customization could have its benefits but it could also have its draw backs, not all technology teachers especially at high school level have a background in programming which means they could set tasks that are overly complicated for the year they are teaching without knowing about it. They also intended to add in items to help students solve problems, *“Items can be used to aid the player, to complete other quests or educative information about the problems that are being solved”* (Coelho, Goncalves et al, 2011, p.64).

We feel by adding in items that can help students solve puzzles means that these items can be stored up early in the game and make later programming puzzles extremely simple. By allowing this it means instead of a student might learn about more complex programming concept they will be able to make the puzzle too simple to the point they do not advance their knowledge. We do agree students may need help but we feel by giving them a solid understanding of the basics of programming before moving on and clear instructions to their task, this method would work best. Similar to Ouabhi, Kaddari et al, Coelho, Goncalves et al results only note that the students that tested their game enjoyed it but not if it had an impact on their skills and abilities. *“The participants’ feedback on the game was very positive and at the end of the tests each of them filled a survey.”* (Coelho, Goncalves et al, 2011, p.69), we are starting to see that teaching programming through the use of serious games helps to increase student engagement, there is still no word on if it improves student knowledge.

To check if the players inputted code is correct Coelho, Goncalves et al (2011, p.68) use a web service which stores both the compiler and program output. Once the outputs are stored, the web service then checks a database and will either accept or reject the code. If the code is accepted the player will be able to continue with the game but if the code is rejected the player will be notified of their mistake.

We hope to implement a similar approach to notifying the player of errors in order to help them re-write their code as Watson et al (2011). Watson et al (2011, p17) approach was to provide explicit corrective feedback in order to give tailor made messages in order to help students solve errors while debugging their code. Our approach to creating programming problems for the player to solve will also use an approach much like Watson et al (2011). Watson et al (2011, p17) approach is to not give the player random or arbitrary programming tasks, but instead create tasks that players may actually use in actually creating a game.

Mitamura et al (2012 p.1813) design and implemented four games to help people learn programming through the use of serious games. Their first game was a typing game where players type out blocks on code shown on screen, once the player completed each task, they are then able to see what the code they just typed out actually does when compiled. Even though players can see the code they have just typed out running at the end of the task and are given an explanation to the code, in our opinion we feel that just copying out code does not help people learn and fully understand the code they are writing.

The second game Mitamura et al (2012 p.1814) developed was a multiple choice game were players can use a controller or keyboard to rotate three cubes, then they select the correct code on the face of the square to fill in blank spaces in order to complete the block of code. Compared to Mitamura et al first idea the team agree that a game with this style of challenge where the player has to think about and learn the code themselves is a lot more useful and beneficial to teaching people to code. However we feel having the player type out the code would have helped more beneficial to the player to learn about the code.

Game three from Mitamura et al (2012 p.1814) is a 3D game similar to game two, this time however it is a player vs player game where the players must collect pieces of code scattered around a level in order to complete a block of code missing certain words. The players can also collect the words the other player needs in order to halt their process. Again this idea is better to their first proposed idea as there is more emphasis in learning what the pieces of code do, however we feel that by having the player focus on different code at the same time in a fast paced environment runs the risk of them not fully understanding what the code does or is used for. We do agree that from the three games looked at so far this is the most game like and the player vs player challenge would help keep players more engaged down to the fun factor.

The fourth game from Mitamura et al (2012 p.1815) is a mobile based game where player can set a list of specific actions for the character to follow and make it to the end of the level. Similar to the third game this game takes more influence from entertainment video games, however again by not using a programming language and just arbitrary commands the players are not taught how to program in a more traditional approach. When comparing the four ideas we feel the third game is the best as it combines a traditional approach to teaching programming with the fun and excitement of a video game. To assess the players the four games check the player selected or input answers, once the answers have been checked the player will be able to move on to the next level or challenge. If the answer is wrong however the player will be notified and will need to redo the current level/challenge.

Kazimoglu et al (2012, p.1994) designed and created Program your robot, a serious game design to help users learn basic programming skills. The idea is for the user to give directions to the robot to allow it to move through the level. “*The goal of the game is to assist a robot and help him to escape from a series of platforms by constructing an escape plan called a solution algorithm*”. Kazimoglu et al, (2012, p.1995) mention how they are wanting the user to learn to code yet have the option within their game that allows the user to “ignore programming commands”. “*During runtime only the commands set inside the Main method are performed by default, in the initial sequence determined by the players”.* To allow the user to have an option of using programming commands or not may be a nice introduction for them, but this could take the users focus away from actually learning to code the commands themselves. Kazimoglu et al (2012) have an option where the user can drag commands into a main method section, as opposed to the user just writing it out themselves.

*“At any time in the game, players can use the debug mode to detect potential errors in their solutions. “*Kazimoglu et al (2012, p.1995) have included a debugging section in their game which will find an error in a certain area of the users algorithm. This feature is useful for finding an error in the users work, but with our game design we will not be allowing the user to move forward until the certain section is correct. We plan to keep track of their attempts to complete a certain section to inform them of how well they are doing or if they look like they need improvement in any specific area.

Arnez et al, (2014, p.88) explain their investigation of bringing students studying computer sciences together with members of staff and other students alike. “*Our initial efforts focused on simple game-like assignments that challenged students to put specific programming concepts into practice.”* Arnez et al (2014), wanted to have the users come up with original video game designs that would also teach a certain condition of programming. Arnez et al, (2014, p 89) mention that the game platforms are aimed at the students in introductory classes and say that they strive for simplicity. This is so the students can focus on working with game mechanics that they are familiar with. This idea is good in the sense it gives the students something to work on and create what they know, but it is only practise and not exactly teaching them something new. “*From the start, student team members — as developers, managers, and artist/designers—are involved in a full studio production environment, gaining invaluable hands-on experience”.* Arnez et al, (2014, p.90) focus on the students working together in project teams. This is an area that works really well for individuals who are learning. The connection that the teams have from working with each other to make deadlines and milestones is beneficial for giving them an insight into project team work.

The feedback from the students seem to be positive, mentioning how rewarding the experience was being involved in the development of building a game to teach programming. Within Arnez et al, (2014) all the screenshots of the four games created look rather good, fig 3, *SpaceSmasher,* which is designedto teach “if” statement and fig. 5, *GhostFinder* which was designed to teach arrays. It just seems to be a lot of work focused on to these games yet the games themselves only really cover one area of programming learning. Whereas our intended game will give the user more skills to learn within just our one game. Saving the user time essentially, by not needing to play multiple games to learn different skills.

Tillmann et al, (2012) have developed a well-designed web based puzzle serious game aimed to help teach programming material to users all the way through to graduate courses. The idea is for the user to edit code in a puzzle based browser game and have Pex4Fun to execute and analyse it in the cloud to find out if the engine can find interesting argument values Tillmann et al, (2012, p.90). This style of tutorial is well thought out, from making the user work straight away by having to enter in their own code to see if the puzzle will run correctly, however it may be viewed as daunting to start for a user that is learning computer science in high school and not had experience in writing code before. This application looks to be focusing on users that have coding knowledge to begin with and as Tillmann et al, (2012, p.90) mentioned “*from high school all the way through graduate courses”,* there is a large gap in knowledge between these study levels.

Our game design is aimed at giving a step by step guide to the user for the progression of their learning. This way they will be getting the correct information needed to proceed further regarding certain programming areas as they play their game to control their spaceship and actually see an end product working.

*“Create and publish Coding Duels.**Five steps are needed to create and publish Coding Duels.”* Tillmann et al, (2012, p.90) mention for the creation and publishing of coding duels (which is the interactive puzzle that the teachers can assess the users on) that it has five steps involved to set it up. This means that the teacher will have to:

1. Sign in to create an account
2. Write out a certain specification for the users
3. Create the Coding Duel
4. Edit visible program text
5. Publish after editing the visible puzzle method

Tillmann et al (2012, p.91) from these five steps it seems like a large amount of work needed to be carried out for the teacher before even giving the link to the students. It is respectively a good idea that allows the teacher to create assessments, however every teacher may not have the time to develop these out. Again referring back to our own game design of having a guide throughout their learning experience, we feel the users engagement will be more grater where they can see what they are learning rather than possibly writing the wrote code out and not knowing until afterwards.

Ibrahim et al, (2011) explore the topic of how student’s engagement with educational games for learning programming skills enrich their knowledge and understanding in a fun approach. “*Previous studies have found that using educational games as a learning approach can enhance student’s learning of various learning domain: namely cognitive, affective as well as psychomotor skills (Garris et al, 2002).”* Ibrahim et al, (2011, p.206) mentions how previous research have shown that by teaching through the use of educational games that the students learning skills are enhanced due to them understanding the subject content easier. The action of the student learning through games, motivate their learning as appose to structural class based learning.

Ibrahim et al, (2011, p.208) explain their methodology in their study that they created two mini online Educational Games (EG). The games where crossword and duck shooting. “*We used the common games genre with the goal that for students do not spend time learning how to play the games they start right away to play (and learn)*.” Ibrahim et al (2011, p.208) mention that they used common games genres to show the user something that they were familiar with and therefore not waste time on something complicated and new. We are taking this same approach in relation to our System Builder where the user will build a spaceship that they will control. Here the similarity of a known game genre (shoot em’ up) is beneficial in the engagement to bring the user in.

*“The noticeable finding from this construct is students found using games makes the programming subject more interesting (86%), and only 14% of them not sure about it.”* Ibrahim et al, (2011, p.208) carried out their findings with the use of questionnaires to the 21 students that took part in the project. Here we can see that feedback gathered about the students learning through EG is highly positive. This is an area that will motivate our team to work towards with the regards to the students’ engagement with learning through an enjoyable platform.

# Potential User Survey

As well as conducting a review of some previous research and proposed solutions into the area of introduction people to programming through the use of serious games, we also conducted a brief survey targeting people that either wanted to learn how to program, are currently learning or stopped learning due to no good teaching method available. By using the survey we are trying to understand why people who have tried to learn to program but stopped and what the reason for them stopping was, the survey also allows us to see why people who want to learn how to program have not tried to learn yet. As well as understand what makes other teaching solutions so appealing. The table below is an example of how qualitative research has been used for the findings in the user survey. Looking at research from Sandelowski (2000), the focus on Data Collection talks of how qualitative studies towards discovering out who, what and whereof findings including experiences or events. This relates to the relevant survey found below in table 1 that explores the users experiences with relation to previous programming educational learning through the questions provided for them.

From the findings gathered in the tables below, we can see that the results displayed focus more on actual explanations and words as appose to numeric and statistical findings. This is what qualitative researching mainly focuses upon when carried out as appose to quantitative research. Haber and Wood (2014)

Table 1 shows the questions required for participants to answer and the available answers to the survey participants.

|  |  |
| --- | --- |
| Question | Answer |
| Age Range | * 14-18 * 19-23 * 24-28 * 29+ |
| Have you tried to learn to program? | Yes or No |
| If you are learning or tried to learn to program did you find it enjoyable? | * Love it * I enjoy it * I don’t mind programming when I have to * I hate it |
| What did you do to learn to program and did you find it helpful or enjoyable? | User text response |
| Are you still learning? | Yes or No |
| If not what made you stop? | * Examples where too boring * I did not feel I was learning proper skills * Hard to start * Work felt too repetitive |
| If yes what approach are you using to learn to program? | * Follow traditional programming tutorials to make general software, e.g. word processing software * Follow game development tutorials * Use an online course service e.g. Codecademy * Learning at an educational institution * Reading programming books |
| If you stopped learning would a game that teaches you how to learn programming by making a game would you be interested? Could it make you start again | Yes or No |
| If yes what features would you like to see | * Start from the basics * Keep me entertained * Gives me relevant skills and or knowledge * Ability to start programming after completing * Step by step guide through various programming features e.g. variables, if statements, functions |

Table 1: Questions and available answers for potential user survey.

Table 2 shows the feedback from the potential users’ survey sent out to help with the proposed solution design.

|  |  |
| --- | --- |
| Question | User Feedback (10 Participants) |
| Age Range | 50% of participants’ were aged 19-23, 40% were aged 24-28 and 10% were aged 14-18. |
| Have you tried to learn to program? | 80% of participants’ we asked have tried learning how to program. |
| If you are learning or tried to learn to program did you find it enjoyable? | Only one of the 8 people surveyed only 1 loves programming and only 1 enjoys programming with the rest not minding it or hating it. |
| What did you do to learn to program and did you find it helpful or enjoyable? | Half of the participants’ learned though online tutorials and enjoyed it with the other half learning at an educational institution and did not like it. |
| Are you still learning? | Out of the 8 people that have programmed only 4 of them are still currently learning. |
| If not what made you stop? | Most people stopped learning to program as they felt they were not learning any relevant skills. |
| If yes what approach are you using to learn to program? | From the people that are still learning to program most are learning from online video game development tutorials. With following traditional programming tutorials and using online courses being the second most used methods. |
| If you stopped learning would a game that teaches you how to learn programming by making a game would you be interested? Could it make you start again | 3 out of the 4 people who have stopped learning how to program would be interested in starting to learn again with the help of a game. |
| If yes what features would you like to see | The most requested feature people want from the game is to be entertained, being taught relative skills and given the ability to then go on and create your own programs were the second most requested features. With a step by step guide to learning how to program begin third and starting from the basics of programming being the least requested feature. |

Table 2: Feedback from the potential user survey.

Looking at the results only half of the participants’ that tried to learn how to program have kept it up, with most stopping due to not learning any relevant skills. From the 4 people that are still currently learning to program their most used method of learning is following online tutorials to make a game. This means using a serious game to combine traditional learning with elements of video games could prove to be successful in learning how to program. With that being said 3 out of the 4 participants’ that stopped learning how to program said they would be interested in learning how to program again with the use of a video game. The results have shown that when it becomes time to design the game we will need to focus both on the teaching element as well as the gaming element to ensure users will have fun and stay entertained but also learn relevant skills they can use outside of the game. Phantom Tech will take into consideration the results in which we gathered from the survey and focus on these areas that have arisen in questioning and aim to excel in delivering this to the users. Exploring the section of Research as a Process in the paper Combining IS Research Methods: Towards a Pluralist Methodology by Mingers (2001), looking at the heading Assessment Mingers talks of how the importance of the results and inference of other situations gathered helps towards the creation of assessments. Our approach to taking the feedback into account works with this methodology under Mingers Assessment heading.

One fundamental reason behind the creation of System Builder is to get users to enjoy learning about programming a language in a fun and relative way. Breaking away from the generic learning teaching methods used in the past and incorporating the creation of relevant, fun assessments, i.e. creation of the game. Jessel, et al (2010) speak of how their questionnaire carried out on motivation of students was a key focus in relation to their serious game. The same approach can be seen through the research questions used in the questionnaire above.

# Proposed Solution

## Influence to Our Design

Two similar pieces of software that inspired the design of our proposed solution described in this paper are Alice and Scratch. As described by Calitz et al (2015) Alice and Scratch are visual based programming environments that allow users to create software such as games by dragging and dropping blocks of code with pre-programmed instructions, these blocks of code can also only be joined in pre-set ways so that only logical errors occur.

The simplistic nature have seen these programs used in educational institutions such as high schools and universities and seem to be more enjoyable with students than using conventional learning techniques to teach programming. A study by McDonald and Johnsgard (2008) used Alice to create a new class to introduce new students at Monmouth University to object orientated programming, by introducing this class before the universities C++ class it seen a higher number of students who took the new Alice class pass then those who did not take the class. The results were calculated that only 32 out of 69 students who did not take the Alice class passed, were as 26 out of 37 students who took part in the Alice class passed the C++ class.

However these pieces can still have issues with certain students. A study by Ferrer-Mico et al (2012) where they got students aged 12-13 years spilt into two groups of beginners and advanced users. Comments from journals filled out by the beginner students stated that they could not continue with their game because the block they wanted did not exist and they could not continue because they did not know what they wanted to do. With these comments Ferrer-Mico et al (2012, p.1221) concluded “*These type of comments show that the students were just focusing on how to use this new tool and not confident enough to use it to create new situations and knowledge.”* Due to this factor our solution will provide students will clear instructions on how these programming concepts work and have them code them in situation that they would use them at other times such as handling keyboard input. Our thought by doing this is that students will better grasp programming fundamentals and be able to put what they learned into practice with confidence they know what they are doing.

## Teaching Method

In identifying teaching methods that work well the team also looked at studies on this area to find out what traditional teaching methods students like and what they like about this method exactly. A study by Sajjad (2012) found that out of all the students questioned lectures were the favourite teaching method used, this is due to the fact that the teacher provides all knowledge related to the topic beforehand. To meet this similar method of teaching the team proposes to have an explanation and a small close but not exact example for each of the question asked so that users have adequate knowledge for the task.

Goel et al (2008) carried out a study to see if medical students that used computer assisted instructions versus the traditional method of lectures to teach students about dermatology morphology gained different results in a test. In the results in Goel et al (2008) they found that computer assisted teaching in this case an online tutorial students gained a lightly higher grad compared to the traditional approach. However a study by Diwakar et al (2007) were a group of medical students took part in an interactive lecture and a group of students played a game to learn about child development. The students’ performance was assessed with a multiple choice questionnaire. Unlike Goel et al (2008), Diwakar et al (2007) the year before found that students apart of the group that attended the interactive lecturer scored significantly higher than the group of students that played the game.

By analysing the results from all three papers it is clear to see that even though computer assisted learning can benefit students in certain situations it is always not granted. The results also show that a more traditional approach to learning such as lectures in this case can still be one of the more beneficial teaching methods. In conclusion the team aims to create our own teaching method that is a hybrid of traditional teaching methods and newer teaching methods by mixing the best areas available from both.

## The game design

We propose a game where the player will play a space pilot trapped in their ship due to a previous attack damaging the ships systems. The player will then have to re-code their systems in order to make it through the level and get themselves and their ship back home safely. The idea behind adding in a story and a goal for the player to experience and work towards will make the application more game like and enjoyable than just a piece of software like Alice or Scratch. The reason behind this thought is a study by Kelleher and Pausch (2006) were they used Alice to introduce programming to middle school girls animated stories, found that this method helped increased the students interest in learning how to program. We also agree that this same story telling effect could have an impact on a lot of user’s not just female users but also male users.

The player will start off by learning about variables then once this task(s) has been completed they will move on to if statements and so on until their ships systems have been rebuilt and they can use the ship to move and shoot in order to complete the level. We have decided to go for a 2D Shoot em up genre of game as we feel this genre has a good mix of simplicity and fun so they tasks will not get too complicated.

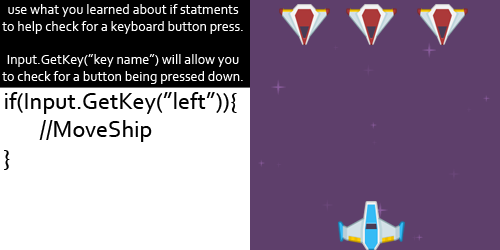


Fig. 1. Mock-up of the how the game will look.

The processes of flow is stated as follows:

* The player will be presented with a task above their text editor positioned next to the game (Fig. 1).
* The user will enter their answer in order to solve the task.
* The game will then check to ensure the users answer is correct, on screen feedback will be provided to inform the player on the status of their current task.
* The player will then be able to play the game to see what the code they just entered is used for.
* The player will then be taken to their next task.

The user will be expected to type all of their code into an in game text editor and will be able to play the game with the arrow keys and spacebar on their keyboard.

## How assessment will be carried out

In the game the user will be taught basic code. When writing this code in the console. The player will be made to lay out and format their code in an understandable and suitable fashion. This will create a good habit where the player will format their code when working on other projects in the future. Having well formatted and laid out code will help the user understand the code they are writing. Seeing structured code written down will be clearer to read and understand compared to untidy, unstructured code.

The user will not be able to move on from uncomplete tasks. This means if a user tries to skip a section of the game it will not let them. The game is designed to teach the user the very basics of coding. If the user start to skip parts of the game then they will end up not being taught some of the fundamental lessons of the game. This will also have an adverse effect on their game because it will have parts missing. Having the user fully complete the task will enforce the lesson that is being taught.

The game will be designed to keep track of the player mistakes at each stage of the game. This is to identify areas of improvement. This information will be passed on to teachers, tutors or the student to inform them of the progression and possible areas of improvement. The information that the game keeps track will be to ensure that the user is fully aware of what parts of coding they need to work on. According to Hummel (2008) this kind of feedback in programming based games is generally restricted even though it is beneficial to the end result of the user.

However, the feedback has to be relevant and on a continuous basis Lau et al (2010, p. 13) says “*Instant feedback offers several advantages to students. For example, it focuses the learner's attention on relevant information, ensuring that the important information is perceived consciously and processed in working memory”.* Feedback is a very important part of learning. This is why we will have a feedback when mistakes are made and at the end of each lesson. The game will also send feedback to the user at the end of the game to show them their strengths and weaknesses over the time it took them to complete the game.

Many novices struggle to act upon standard compiler messages because they are too complicated for them to understand, Claims Fisler (2011). This is because compilers are usually not designed for novices. Our feedback will show the user exactly what they have done wrong and give them guidance on what they need to change. Lau et al (2010) says that many errors make can be attributed to a poor understanding of what they are doing. To stop novices struggling we must have high quality feedback. This information will have to be in plain English and explain to the user what has went wrong without using confusing technical terms that might appear in other feedback.

## Reasoning

Our team are confident in the approach that we have taken to develop our Educational Game (EG)”System Builder”. The inclusion of tracking the player’s individual progress is a connection that will engage the student with the learning process of the game greatly. With the tracking system in System Builder, it will give a roundup of the students’ progress after each certain section, allowing the students to check on their work if they come back to the game in the future. This is beneficial for the institution that we aim to market this game at, a lecturer or teacher could access the students’ progress and allow the viewing of areas that need addressed or that should be praised. Eck (2006) mentions on how further educational institutions should hire instructional designers with experience to help assist with design and to support for the students. This is a good idea for Digital Game Based Learning as it will help out the students on a one to one basis but will lead to expensive cost issues for the institutions as they would need to pay to fill this position. Whereas the focus of System Builder is to cut out the use of an additional teacher to take them through the game as it provides the users with a step by step guide, making sure it is correct before moving on. After 5 attempts we will prompt the user with a mistake if they are not progressing. With this approach, it cuts out the need to have the additional experienced staff member. The outcome that the student will make in the completion of System Builder will be of a game that they’ve built in a genre that is very popular, Shoot em up.

Ibrahim et al, (2011) mentioned that when they created their two mini online EG, that they focused on creating common game genres. This is a valuable area we have picked up on as a team and follow this thinking with regards to our own creation. With regards to Ibrahim et al, (2011) the two mini online EG that they developed seem to be rather easy and not exactly challenging for development. The students play a Crossword where they have to enter the correct answer or select the correct answer in the second game, Duck Shooting. This seems to our team that the student is only getting the information that they might need by typing the answer correctly, but they’re not getting the correct practice with regards on how to actually write out the code correctly. By completing a Crossword relative to programming, is this going to allow them to create a game of their own? No, probably not as they will be unsure about carrying it out. System builders’ step by step guide, teaches the student the valuable information needed and allows them to put it into practice at the same time by creating their own game.

# References

Alvarez, J., Djaouti, D. and Jessel, J. (2011) **Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches,** Classifying Serious Games: the G/P/S model. **Waterford Institute of Technology, Ireland: IGI Global.**

Arnez, F., Pace, J., Sung, K., (April 2014) Learning while Building Games for Teaching. IEEE Computer Society- Entertainment Computing, Vol.47 (4), pp.88 – 91.

Backlund, P., Johannesson, M. and Susi, T. (2007) Serious Games – An Overview. Skövde: Department of Communication and Information, pp.1-28.

Backstein, D., Cowan, B., Dubrowskie, A., Kapralos, B., Porte, M. and Sabri, H.(2010) Serious games for knee replacement surgery procedure education and training. Procedia – Social and Behavioural Sciences, Vol. 2 (2) pp.3483-88.

Bradlyn, A., Cole, S., Kato, P and Pollock, B. (2008). A Video game improves behavioural outcomes in adolescents and young adults with cancer: A randomized trial. Paediatrics, Vol.122 (2), pp.305-317.

Caltiz, A., Cilliers, C. and Koorsse, M. (2015) Programming assistance tools to support the learning of IT programming in South African secondary schools. Computers & Education, Vol.82, pp162-178.

Carly, S., de Freitas, S., Dunwell, I., Jarvis, S., Knight, J., Mackway-Jones, K., Smithies, R and Tregunna, B. (2010). Serious gaming technology in major incident triage training: A pragmatic controlled trial. Resuscitation Journal, Vol.81 (9), pp.1174-1179.

Coelho, A., Goncalves, R., Kato, E. and Xavier, J. (2011) Serious Games for Introductory Programming. Serious Games Development and Applications, Vol.6944, pp.61-71.

Darhmaoui, H., Elachqar, A., Kaddari, F., Lahmine, S. and Ouahbi, I. (2015) Learning Basic Programming Concepts By Creating Games With Scratch Programming Environment. Procedia – Social and Behavioural Sciences, Vol.191, pp.1479-1482.

Diwakar, V., Selby, G. and Walker, V. (2007) A comparison of teaching methods: interactive lecture versus game playing. Medical Teacher, Vol.29 (9-10), pp.972-974.

Donovan, L. (2012) The Use of Serious Games in the Corporate Sector. Irish Symposium on Game Based Learning 2013. Learnovate Centre, Technology Centre for Learning Innovation. Retrieved from www. learnovatecentre.org/research/our-work/corporate-learning/serious-games/

Eck, V., R. (2006) Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless…. EDUCAUSE Review*,* Vol.41 (2) pp.1-16.

Ferrer-Mico, T., Prats-Fernandez, A.M. and Redo-Sanchez, A. (2012) Impact of Scratch programming on students’ understanding of their own learning process. Procedia – Social and Behavioral Sciences, vol.46, pp.1219-1223.

Fisler, K., Krishnamurthi, S. and Marceau, G. (2011) Measuring the Effectiveness of Error Messages Designed for Novice Programmers ACM Technical Symposium on Computer Science Education, pp.499-504.

Greer, T. (2013) 2013-2018 North America Mobile Edugame Market. Ambient Insight Research, pp.5-21.

Greer, T. (2014, July 22nd) The 2013-2018 Worldwide Game-based Learning and Simulation-based Markets. Paper Presented at the Serious Play Conference 2014, USC School of Cinematic Arts. pp. 1-33.

Goel, R., Jenkins, S. and Morrell, D.S. (2008) Computer-assisted instruction versus traditional lecture for medical student teaching of dermatology morphology: a randomized control trial. Journal of the American Academy of Dermatology, Vol.59 (2), pp.255-259.

Haber, J. and Wood, LB.G. (2014) Nursing Research: Methods and Critical Appraisal for Evidence - Based Practise. 3251 Riverport Lane, St. Louis, Missouri 63043: Elsevier.

Hummel, H., Nadolski, R., Westera W. and Wopereis, I. (2008) Serious games for higher education: a framework for reducing design complexity, Journal of Computer Assisted Learning, Vol.1 (5) pp. 420-32.

Ibrahim, R., Yusoff, M., C., R., Omar, M., H., Jaafar, A., (2011, January) Students Perceptions of Using Educational Games to Learn Introductory Programming. www.ccsenet.org/cis Computer and Information Science, Vol.4 (1) pp.205 – 216.

Jessel, J, P., Muratet, M., Torguet, P. and Viallet, F. (2010) Experimental feedback on Prog&Play, a serious game for programming practice. Computer Graphics Forum, Vol.30 (1), pp.61-73.

Kazimoglu, C., Kiernan, M., Bacon, L., Mackinnon, L., (2012) A serious game for developing computational thinking and learning introductory computer programming. Procedia – Social and Behavioural Sciences, Vol.47, pp.1991 – 1999.

Kelleher, C. and Pausch, R. (2006, September 4th-8th) Lessons Learned from Designing a Programming System to Support Middle School Girls Creating Animated Stories. Visual Languages and Human-Centric Computing, IEEE Symposium, Brighton.

Lau, R., Li, F. and Watson, C, (2010), Learning Programming Languages through Corrective Feedback and Concept Visualisation, ICWL'11 Proceedings of the 10th international conference on Advances in Web-Based Learning, pp.11 – 20.

Lau, W. H. R., Li, W. B. F. and Watson C. (2011) Learning Programming Languages through Corrective Feedback and Concept Visualisation. Advances in Web-Based Learning, Vol. 7048, pp.11-20.

McDonald, J. and Johnsgrad, K. (2008, April 14th-17th) Using Alice in Overview Courses to Improve Success Rates in Programming I. Paper presented at the Software Engineering Education and Training 21st IEEE conference, Charleston, SC.

Mingers, J. (2001) Combining IS Research Methods: Towards a Pluralist Methodology Information Systems Research, Vol.12 (3), pp.240-259.

Mitamura, T., Oohori, T. and Suzuki, Y. (2012, October 14-17) Serious Games for Learning Programming Languages. IEEE International Conference on Systems, Man and Cybernetics, pp.1812-1817.

Tillmann, N., de Halleux, J., Xie, T., Bishop, J., (2012) Pex4Fun: Teaching and Learning Computer Science via Social Gaming. 25th IEEE Conference on Software Engineering Education and Training, pp.90-91.

Sajjad, S. (2012) Effective Teaching Methods at Higher Education Level. Paper presented at the Second Biannual WFATE conference, Kenya.

Sandelowski M. (2000) Focus on Research Methods What Happened to Qualitative Description? Research in Nursing & Health, Vol.23, pp.334-340.