Serious games as innovative formative assessment tools for programming in Higher Education

Thomas Hainey, Gavin Baxter, Julie Black, Kenneth Yorke, Julius Bernikas, Natalia Chrzanowska, Fraser McAulay.

Abstract

Serious games are becoming increasingly popular as alternative supplementary learning approaches across all disciplines at all levels. Learners are more enabled to become more actively involved, not only in the learning process but also in the design and development of innovative formative assessment tools. This paper will present four serious games for programming education developed as part of a Serious Games honours year module in Higher Education at the University of the West of Scotland. Computer Games Development students were tasked as part of a research project to design and implement a serious game for students in their course or other Computer Science related courses in earlier years to provide their peers with innovative games for the purposes of formative assessment. It was hoped that the participants could enhance the engagement of courses that they had already passed but also utilise their creative Games Development skills in a new productive way to enhance the assessments in their own course having had the experience. Permitting students to design their own formative serious game assessments in this way enabled enhanced creativity with the students creatively tackling the overall problem in insightful and surprising ways. This paper will look at four games created for the purposes of teaching programming by analysing the overall learning outcomes, development process and providing a description of the games. The idea is to provide insight into the development process of serious games for programming education in terms of similarities, differences and to produce a basic framework for development in relation to content, assessment, and game mechanics. The paper will produce an overview of the excellent games created and present a case for increased active involvement of students to develop their own serious games – particularly on Computer Games degree or college courses.

1. Introduction

Programming is a very complex skill requiring years to master and is a contributory factor of the high levels of attrition experienced on Computer Science related courses. Moreover, Software Engineering graduates have a reputation for being under prepared for industry indicating that new novel teaching approaches should be investigated beyond the traditional lectures, tutorials, and labs extensively utilised. This is particularly true now as we enter a post-pandemic age of hybrid delivery. According to the Young Scot Report "*young people have a significant stake in how assessments need to change to reflect our future society and economy.*" In relation to inclusivity and accessibility, online resources need to change in relation to interactivity by providing games that can assess levels of knowledge and support progress (SQA, 2018). With this in mind Computer Games Development students in Higher Education are in a unique position to develop gamified, innovative formative assessments for courses that they have had high levels of experience in. Not only should learners have a say in the assessments that they are undertaking but also a direct role in developing these assessments. Computer Games students undertaking a final year Serious Games module were tasked with the objective of developing gamified innovative formative assessments for courses they had undertaken in earlier years of their course. This allowed application of their game development skills in a highly focused way and allowed the learners themselves to utilise their creativity, innovation, ingenuity in parallel with their course related knowledge. This paper will present a detailed description of four games for programming education that were developed and perform a basic comparison of learning outcomes, and game mechanics integrated in relation to pedagogy utilising the Learning Mechanics-Game Mechanics model, genre, and implementation/format.

2. Previous Work

Serious games have a rich history (Boyle, Hainey and Boyle, 2021) have been used for educational and training purposes in a number of diverse areas and have become an exceptionally popular research topic worldwide particularly in the last two decades (Schöbel, Saqr, and Janson, 2021). Zyda (2005) defines a serious game as "*a mental contest, played with a computer in accordance with specific rules, that uses entertainment to further government or corporate training, education, health, public policy,* *and strategic communication objectives.*" Developing a serious game is multidisciplinary challenge involving theory or cognition, pedagogy, flow, motivation, game design story, goals, rules, and content (Bellotti et al., 2013). Serious games have now become so ubiquitous that there are taxonomies for different types of serious game in different disciplines e.g., a taxonomy of games in Science Education (Magnussen, 2014) categorising games into:

- **Training games** – designed to train skills through repetition of individual exercises.
- **Inquiry Games** – focusing on scientific inquiry, gathering data, formulating hypotheses and building arguments.
- **Professional simulation/epistemic games** – gamers role play scientific or technical professions.
- **Embodies system games** – focussing on the manipulation of scientific phenomena.
- **Research collaboration games** – focussing on high-level scientific research and technical development.

Frameworks also exist for the integration of serious games into companies (Azadegan, Riedel and Baalsrud Hauge, 2012). There are also general classification taxonomies (e.g., Djaouti, Alvarez, and Jessel, 2011) where one in particular classifies serious games in terms of:

- **Application** – Education, Well-being, Training, Advertisement, Interpersonal Communication, Health Care and Others.
- **Activity** – Physical exertion, Physiological, Mental.
- **Modality** – Visual, Auditory, Haptic, Small, Others.
- **Interaction style** – Keyboard/mouse, Movement tracking, Tangible interfaces, Brain interface, Eye gaze, Joystick, Others.
- **Environment** – Social presence, Mixed reality, Virtual Environment, 2D/3D, Location awareness, Mobility, Other (Laamarti, Eid, and el Saddik,2014).

Effective game story and narratives are also considered to be highly important for serious games to promote learning with the following contributing constituents identified by Naul and Liu (2020):

- **Distribute narrative** – the narrative is placed throughout the entire game via environmental storytelling and is not isolated in one place.
- **Endogenous fantasy and intrinsic motivation** – inextricably interconnecting the fantasy with the learning outcomes is more conducive to intrinsic motivation meaning that if the player is motivated by the fantasy, then they will be motivated by the learning.
- **Empathetic characters and virtual agents** – relatable and powerful characters have the power to influence attitudes and decisions more effectively.
- **Adaptive and responsive storytelling** – This is where learners and gamers in general are more compelled by stories that are customised to them.

Research has been performed to produce a framework to map game mechanics to learning mechanics utilising Bloom's ORDERED thinking skills (Arnab *et al*., 2014). The existing research indicates that there are many attempts to classify serious games in relation to discipline, application, interactivity, format, game mechanic and educational content. This paper is primarily concerned with programming education and there have been a large number of serious games developed within Computer Science for that purpose. Miljanovic and Bradbury (2018) performed a review of 49 serious games for programming and categorised them in relation to educational content:

- **Algorithms and Design** – algorithmic comparison, problem solving.
- **Fundamental Programming Concepts** – non-specific programming concepts, syntax and semantics, variable and primitive data variables, expressions and assignments, input and output, conditionals and iteratives, functions and parameters, recursion.
- **Fundamental Data Structures** – arrays, heterogeneous aggregates, string processing and abstract data types.
- **Development methods** – program comprehension, debugging strategies and documentation and program style.
- **Software Design** – Object Oriented Design Paradigms.

One particular problem that is identified is that as a result of serious programming games being independently developed this does not take into account lessons learned from existing games. Suitable evaluation also has to be conducted in relation to appropriate validation. This paper will attempt to categorise and describe 4 programming games utilising a compiled serious games framework using:

- Learning outcomes (Hainey, 2010)
- Science education categorisation (Magnussen, 2014).

- Game mechanics integrated in relation to pedagogy utilising the Learning Mechanics-Game Mechanics model (LM-GM model) (Arnab *et al*., 2014).
- General classification (Laamarti, Eid, and el Saddik,2014).
- Story, Narrative (Naul and Liu, 2020)
- Educational content (Miljanovic and Bradbury, 2018).

The games will be overviewed methodically utilised the following compiled framework shown diagrammatically in Figure 1.

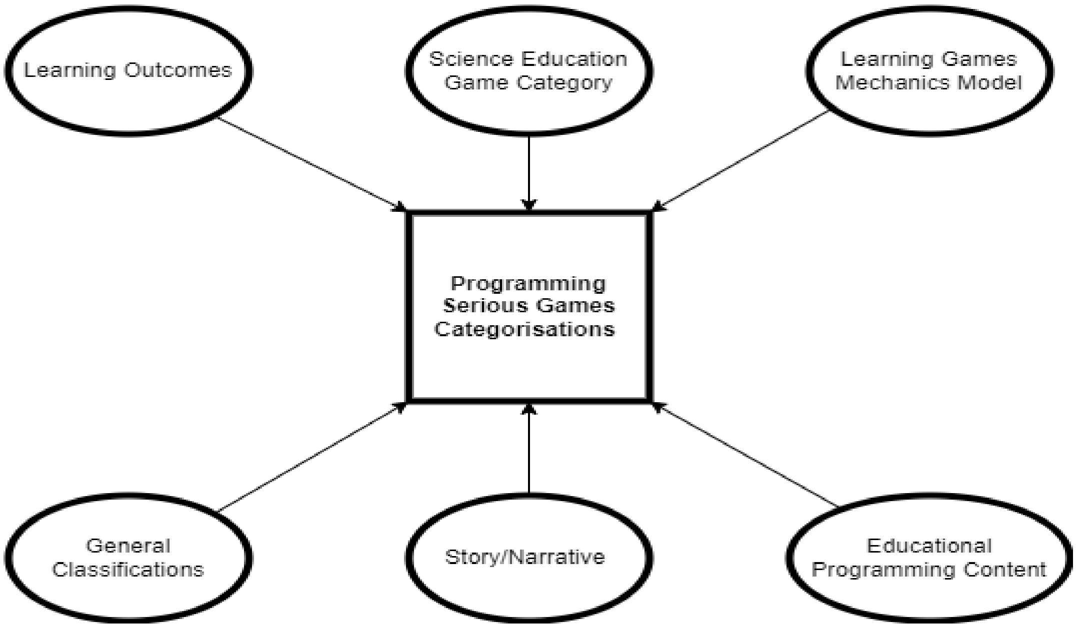

Figure 1: Compiled categorisation framework

3. Programming Games Developed

The games were developed as part of a final year Serious Games course at the University of the West of Scotland as part of a Vice Chancellors Innovation Research Grant. The idea was to utilise the most valuable and underutilised stakeholders in relation to producing innovative formative assessments for supplementary learning – the students themselves. It allowed them to reflect on their graduate capabilities and consider the types of games that they would like to have had during courses that they had already passed and to consider innovative ways to teach lower-level undergraduates in their course. The four programming games selected achieved the highest grades in the course and utilised endogenous fantasy coupled with creative mechanisms to fully integrate fun, immersion, and engagement beyond a formative/summative quiz. We will also use this study to stringently and thoroughly categorise these games to refine the categorisation/methodological framework can be used in the future and further refined.

3.1 Game 1: OOPs: Object-Oriented Problems

Description: The game allows you to be in charge of a Virtual Museum of OOP where some damage has been done and this is then rectified by the player answer all of the questions correctly i.e., restoring all of the missing information. Figure 2 shows screenshots of OOPs.

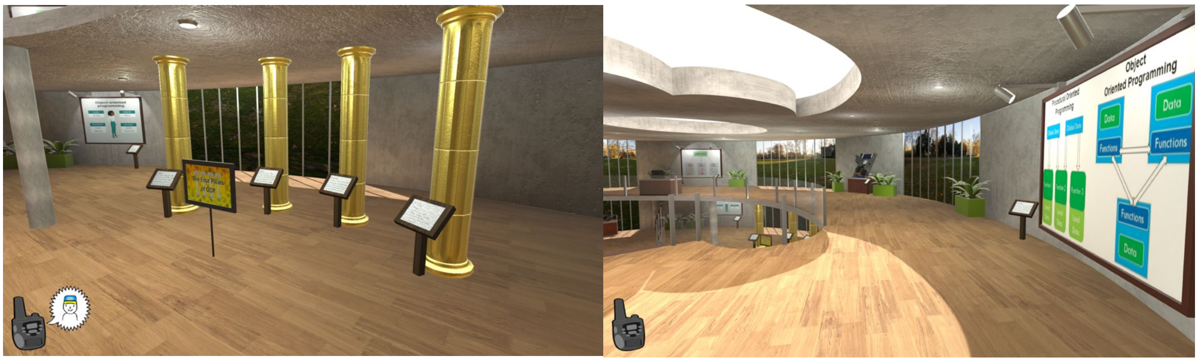

Figure 2: Screen shots of OOP

Learning outcomes

- To understand the core concepts of object-oriented programming
- Identify and describe the four pillars of object-oriented programming
- Teach users how to define various object-oriented programming concepts
- To develop a knowledge of useful object-oriented programming techniques
- To develop understanding of writing object-oriented programs that combine functions and data

Science Education Game Categorisation(s): Professional simulation/epistemic games

Features in the LM-GM model

Game Mechanics	Implementation	Thinking Skills	Learning Mechanics
Ownership. Protégé Effect	**Through story and the fact that the game is a Virtual Museum**	**CREATING**	Ownership, Responsibility
Action Points	**Various exhibits**	**EVALUATING**	Assessment, Motivation
Feedback, Realism	**Q and A with historical programming exhibits**	**ANALYSING**	Analyse, Identify, Feedback
Movement, Selecting/Collecting	**Free movement through the 3D environment**	**APPLYING**	Demonstration
Role-play, Tutorial	**Assuming the curator role and a tutorial example**	**UNDERSTANDING**	Participation/Questions and Answers, Tutorial
Cut scenes/Story	**Tutorial cut scene at the beginning**	**RETENTION**	Guidance, Explore, Discover

General classification

- **Application** – Education, **Activity** – Mental, **Modality** – Visual, Auditory, **Interaction style** – Keyboard/mouse, **Environment** – Virtual Environment, 3D, **Genre:** Virtual Museum. **Implementation:** Unity Engine

Story and Narrative – The story and narrative are utilised in this implementation to set the scene where a cut sequence is played at the beginning showing damage and destruction to the museum exhibits. It is then up to the player to restore the exhibits and the information/knowledge contained therein by arranging all the exhibits in response to the auditory scaffolding. In terms of the classifications of story and narrative the game utilises: **distribute narrative, endogenous fantasy and intrinsic motivation and empathetic characters and virtual agents.**

Educational content – The educational content includes main principles of Object-Oriented Programming (OOP) (encapsulation, inheritance, polymorphism, and abstraction). The origins of OOP, Object Scope, Features of OOP, Types of class, passing classes, defining classes, the meaning of polymorphism, classes using encapsulation, structuring arrays, the keyword used to describe static variables and the first purely OOP language developed. The classification is really software design

3.2 Game 2: Planet Edolia

Description

The game begins with the player in a compromising position, and it becomes apparent that they have to make their way through a Jurassic Park like enclosure by using their coding/dinosaur knowledge to open the gates with a T-Rex in pursuit. Figure 3 shows screenshots of Planet Edolia.

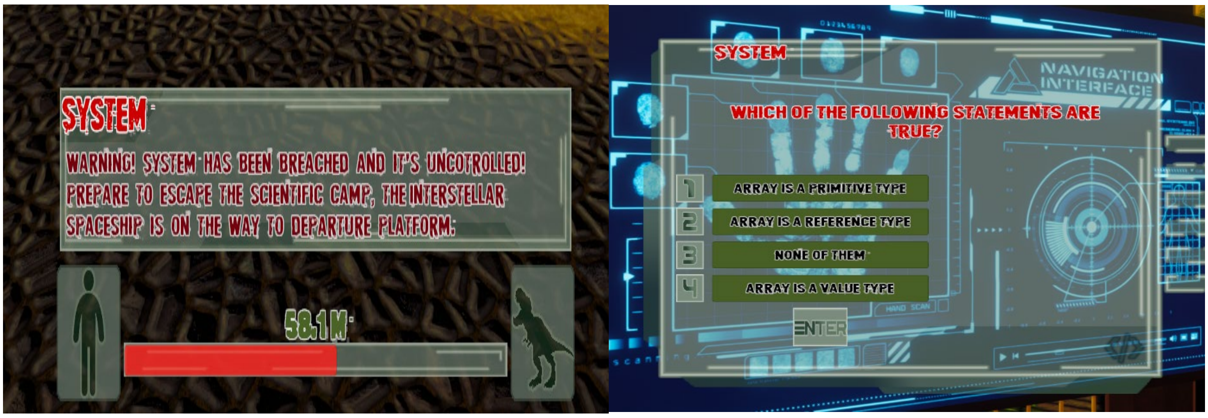

Figure 3: Screenshots of Planet Edolia

Learning outcomes

- Overall knowledge of C#
- Coding syntax
- Object Oriented Programming
- Problem and critical thinking skills.
- Informative information about the dinosaurs

Science Education Game Categorisation(s): Professional simulation/epistemic games

Features in the LM-GM model

Game Mechanics	Implementation	Thinking Skills	Learning Mechanics
Ownership, Status	**Through the situation presented in the game i.e., escape from a Dinosaur enclosure with a T-Rex in pursuit.**	**CREATING**	Ownership, Responsibility
Action Points, Game Turns	**Various gates have to be opened via quiz entering the correct answers**	**EVALUATING**	Assessment, Motivation, Incentive.
Feedback, Realism	**Questions and answers with consequences i.e., the imminent pursuit.**	**ANALYSING**	Analyse, Identify, Feedback
Movement, Selecting/Collecting, Progression, Capture Elimination, Time pressure.	**Movement through a particular route selecting particular answers to progress in a pressurized timeframe.**	**APPLYING**	Demonstration, Action/Task,
Role-play, questions, and answers	**Assuming the role of the curator and a tutorial example**	**UNDERSTANDING**	Participation/Questions and Answers, Tutorial

Cut scenes/Story, Pavlovian Interactions	**Tutorial cut scene at the beginning**	**RETENTION**	Guidance, Explore, Discover

General classification

- **Application** – Education, **Activity** – Mental, **Modality** – Visual, Auditory, **Interaction style** – Keyboard/mouse, **Environment** – Virtual Environment, 3D, **Genre:** 3D RPG/Puzzle, **Implementation:** Unity Engine

Story and Narrative - The story begins with the player in Dinosaur enclosure, and they are told that the T-Rex will be released in 1 minute, giving the opportunity for them to quickly explore and devise a solution. In terms of the classification the game utilises **distribute narrative, endogenous fantasy and intrinsic motivation and empathetic characters and virtual agents.**

Educational content - The educational content includes main principles of Object-Oriented Programming (OOP) (encapsulation, inheritance, polymorphism, and abstraction). The origins of OOP, Object Scope, Features of OOP, Types of class, passing classes, defining classes, the meaning of polymorphism, classes using encapsulation, structuring arrays, the keyword used to describe static variables and the first purely OOP language developed. The identified category is **software design**.

3.3 Game 3: Coder-Bot

Description – The game sets the scene by telling the player that they have to assist in producing an antivirus and allows progression of the Coder-Bot through the levels with a drag and drop sequence of programming constructs. After the code is executed, you then see the result of the coding commands. Figure 4 shows screen shots of Coder-Bot.

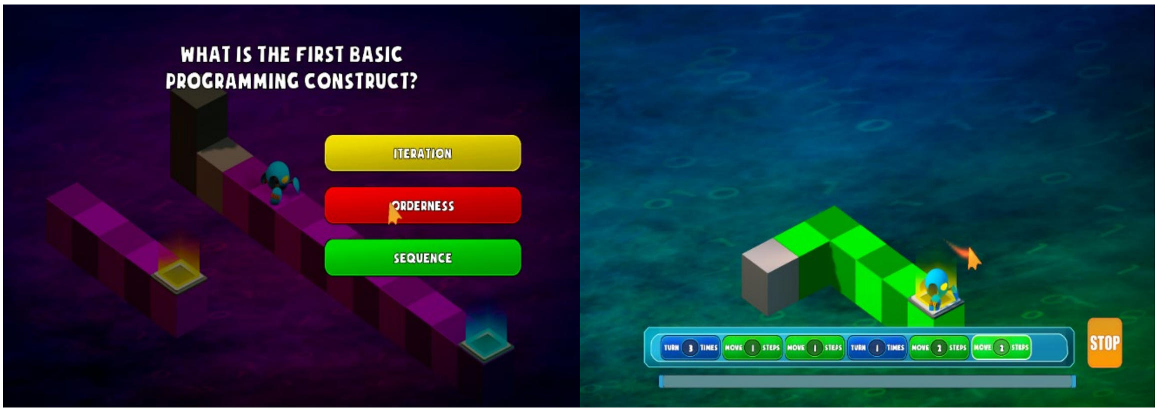

Figure 4: Screen shots of Coder-Bot

Learning outcomes

- Identify and describe the three basic programming constructs.
- Understand the importance of sequence and the consequences of an incorrect order.
- Explain the purpose of selection and when best to use it.
- Recognise the different types of iteration and prevent infinite loops.
- Apply the three basic programming constructs to solve puzzles.

Science Education Game Categorisation – Professional simulation/epistemic games, Embodies system games

Features in the LM-GM model

Game Mechanics	Implementation	Thinking Skills	Learning Mechanics
Ownership, Status, Design Editing, Protégé effect, strategy/planning, Tiles/Grids	Narrative sets the scene in terms of locating an anti-virus alongside a scaffolding companion to plan through a tile and grid environment using coding constructs	**CREATING**	Ownership, Planning Responsibility, Accountability
Action Points, Game Turns, Assessment, Rewards/penalties	Points have to be reached using programming directions or the player falls off the edge.	**EVALUATING**	Assessment, Motivation, Incentive.
Feedback, Realism	Creating a plan and programming script and then executing it. If incorrect the player does not reach the destination.	**ANALYSING**	Analyse, Identify, Feedback, experimentation, feedback, observation.
Movement, Selecting/Collecting, Progression, Capture Elimination.	Movement through a particular route selecting particular answers to progress in a pressurized timeframe.	**APPLYING**	Demonstration, Simulation.
Role-play, Cascading information.	Assuming the role of a coder to successfully navigate a bot using an elongated script.	**UNDERSTANDING**	Participation.
Behavioural momentum.	Once the script is executed you see the results of the simulation	**RETENTION**	Generalisation, repetition, discover, explore, guidance, instruction.

General classification

- **Application** – Education, **Activity** – Mental, **Modality** – Visual, Auditory, **Interaction style** – Keyboard/mouse, **Environment** – Virtual Environment, 3D, **Genre:** 3D isometric puzzle game, **Implementation:** Unity Engine

Story and Narrative - The story sets the scene for the problem by asking for your name and then enlisting your help to navigate a series of levels to get the computer antivirus and defeat Dr. Syntax. The game uses: **distribute narrative, endogenous fantasy and intrinsic motivation and empathetic characters and virtual agents and adaptive and responsive storytelling.**

Educational content - Sequence, Iteration and Selection - **fundamental programming concepts**.

3.4 Game 4: LearnToCode

Game description

The implementation contains six mini-games for various different aspects of programming including: Ball game where the player alters a boolean value to open a door to allow a ball to progress as well as editing the velocity to see how fast it needs to go to reach the appropriate teleporter utilising a degree of trial and error. 3rd person driver allowing the player to alter values associated with objects in the game to improve their gaming experience such as velocity of the car and the amount of petrol items that can be picked up to reach the end goal. Isometric Maze allowing the player to navigate through a maze with limited tries and a number of obstacles by identifying and removing coding structures such as collision boxes. Top-Down shooter allowing the player to actually play the game but also increase their health utilising the code and well as the moving speed and the projectiles fired in terms of number and semi-automatic/automatic. Tower Defence allowing the player to alter the aim of the towers against waves of enemies utilising code and becoming familiar with collision boxes. Endless runner allowing the player to alter the jump velocity to clear gaps and understand the basics of the deltatime functions in Unity. Figure 5(a) shows the Tower Defence mini-game and Figure 5 (b) shows 3rd person driver.

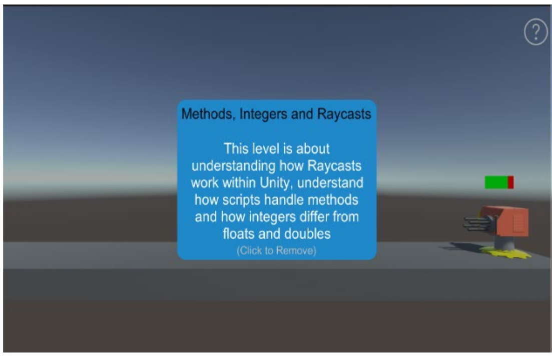

Figure 5(a): Tower Defence Game

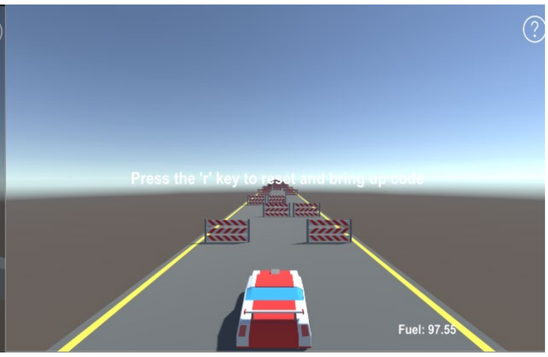

Figure 5(b): 3rd Person Driver

Learning Outcomes

- Knowledge of fundamental programming concepts (Variables, Objects, Properties, Methods, Control Statements (if/else statements and loops))
- Knowledge of C#.
- Problem solving skills.
- Critical thinking skills.
- Ability to code in C# for the Unity engine.

***Science Education Game Categorisation* –** Professional simulation/epistemic games, Embodies system games.

Features in the LM-GM model

Game Mechanics	Implementation	Thinking Skills	Learning Mechanics
Design, Editing, Ownership, Strategy, Planning	Simulation sets the scene with the puzzle/obstacle to be overcome.	**CREATING**	Accountability, Ownership, Planning, Responsibility.
Action Points, Rewards/penalties, assessment.	The game allows advancement to particular points through intervention and success and failure. The reward is progression.	**EVALUATING**	Assessment, Hypothesis, Incentive, Motivation.
Feedback	The results of the coding inputs and what happens on the screen.	**ANALYSING**	Analyse, Experimentation, Feedback. Identify, Observation.
Movement, Progression, Simulate/Response.	The code is entered, and the results are immediate on the screen.	**APPLYING**	Action/Task, Demonstration, Simulation
Role-play	All mini games are viewed in terms of you assuming the role of a puzzle solving programming.	**UNDERSTANDING**	Objectify, Participation, Tutorial.
Behavioural momentum	Cause and effect suitably demonstrated where alterations to the code are then materialised.	**RETENTION**	Discover, explore, generalise, repetition.

General classification

- **Application** – Education, **Activity** – Mental, **Modality** – Visual, Auditory, **Interaction style** – Keyboard/mouse, **Environment** – Virtual Environment, 3D, **Genre:** 3D Puzzle Game, Endless Runner, 3D FP Car game, Isometric Maze, Tower Defence., **Implementation:** Unity Engine

Story and Narrative - The story is based on **endogenous fantasy and intrinsic motivation** by allowing players to observe a scenario and take corrective action.

Educational content - The educational content of this game falls into the following categories: Algorithms and Design, Fundamental Programming Concepts and Software Design

4.Conclusions

This paper is a celebration and academic showcasing of some of the highly innovative, creative, exciting, and outstanding serious games developed by students in a SCQF L10 module in Higher Education at the University of West of Scotland. The entire idea was to involve the most valuable and underutilised stakeholders in education i.e., the learners themselves to develop serious games as innovative formative assessment tools for other learners earlier on in their course.

One extremely interesting point to note is the way that the learners all used their creativity in different ways and despite the fact that we only analysed a very small subset of the games produced, there was a remarkable degree of versatility.

- One was a time-pressured Dinosaur maze game with an integrated quiz.
- One was a Virtual Museum of Object-Oriented Programming with integrated quiz, exhibits and completion tasks.
- One was a drag and drop code modification level traversal game and
- One was manipulating code to achieve more desirable outcomes in a particular context of a game and game genre.

A very rudimentary and preliminary categorisation framework was formulated from a discursive sweep of the literature to provide as much insight into the games as possible and work towards a framework for the categorisation and development of serious programming games. This was a very useful exercise with some of the following revelations:

- It is highly likely that in terms of *Science Education Game Categorisation, the majority of programming games will fall into* Professional simulation/epistemic games and Embodies system games.
- The games adopt distribute narrative, endogenous fantasy and intrinsic motivation and empathetic characters and virtual agents with only one utilising adaptive and responsive storytelling which is far more difficult to implement. The four games in this study seem to integrate endogenous fantasy and intrinsic motivation as a minimum even if it simply altering variables to see the simulated effects.
- The framework used the LG-GM model which was exceptionally useful for deep analysis of the pedagogical effectiveness of the games and how this was going to be achieved. One particular aspect that Universities are concerned with is the idea of retention in their courses and only one of the games incorporated the majority of the learning mechanics for retention. Far closer and more specific consideration of pedagogical mechanics and content needs to be considered through recognised frameworks interconnecting game mechanics and pedagogy which is worthy of consideration in the design phase.
- In terms of general categorisations, the games were implemented in *Unity.* The **Application** was generally educational, **activity** was mental, **modality** was visual and auditory with the primary **interaction style** being keyboard/mouse and the **environment** being a Virtual Environment in 3D, with variable genres. One interesting part that this highlight is interaction and the possible requirement for adoption of new interactive approaches such as VR.
- Educational content was a mixture between fundamental programming concepts and software design which is also something to take into account when developing a programming serious game.

In terms of future directions and recommendations, the games have to be evaluated with learners to ascertain if they enhance, improve, and engage learners in the learning process. We also recommend that this has been a highly useful first study of its kind where Computer Game Development students have developed Serious Games for revision purposes on their own course and that this study should be repeated in other institutions as well as the University of the West of Scotland for the purposes of creating a Serious Games repository with a higher level of specialised focus. The specialised focus being that learners consider what particular revision tools they would have found useful at earlier levels in their course and implement them using their creativity and ingenuity. This ensures increased focus as one requirement for a successful serious game is the "*perfect knowledge of the learning domain to obtain the desired results*" (Barbosa et al., 2014). We will also develop the rudimentary framework into a workable set of guidelines for developing a serious programming game in terms of refinement. The main limitation of this mass innovative formative assessment production is that only really Computer Games Development and Technology students have the technical ability to implement games in general let alone games that require perfect knowledge of a learning domain with access to subject matter experts. In conclusion we do recommend that Computer Games Development courses running Serious Games or Games-Based Learning modules attempt this particular strategy of allowing their students to develop formative, innovative assessments for other learners at lower levels of the course. This makes the learners far more involved in the learning process in a holistic sense and is extremely worthwhile and illuminating process for educators.

References

Arnab, S., de Freitas, S., Bellotti, F., Lim, T., Louchart, S., Suttie, N., Berta, R. and De Gloria, A. (2014). Pedagogy-driven design of Serious Games: An overall view on learning and game mechanics mapping, and cognition-based models. . British Journal of Educational Technologies. 46, 391–411. DOI: 10.1111/bjet.12113

Bellotti, F., Berta, R., Carvalho, M., De Gloria, A., Wiedemann, A. (2013). Serious Games Design: A Tutorial. Retrieved May 2022 from: https://seriousgamessociety.org/wp-content/uploads/2016/09/sg_design_tutorial.pdf

Barbosa, A.F.S., Pereira, P.N.M., Dias, J.A.F.F. and Silva, F.G.M. (2014). A new methodology of design and development of serious games. Int. J. Comput. Games Technol. 2014, Article 8 https://doi.org/10.1155/2014/817167

Azadegan, A., Riedel, J. and Baalsrud Hauge, J. (2012). An Application Framework for Serious Games Integration in Companies. In: The IFIPWG 5.7 16th Workshop on Experimental Interactive Learning in Industrial Management, 3-5 Jun 2012.

Boyle, E.A., Hainey, T., and Boyle, J.M.E. (2021). Serious Games and Cyber Behaviour, *Cambridge Handbook of Cyber Behaviour.*

Djaouti, D., Alvarez, J. and Jessel, J. (2011). Classifying Serious Games: The G/P/S Model. Computer Science.

Laamarti, F., Eid, M., el Saddik, A. (2014). An Overview of Serious Games. International Journal of Computer Games Technology. 2014 (3), DOI: 10.1155/2014/358152

Hainey, T. (2010). Using games-based learning to teach requirements collection and analysis at tertiary education level. Unpublished Doctoral Thesis, Retrieved 20 May from: http://cis.uws.ac.uk/thomas.hainey/Final%20PhD%20Thesis%20Tom%20Hainey.pdf

Magnussen, R. (2014). Games in Science Education: Discussion of the Potential and Pitfalls of Games-Based Science Education. In C. Busch (Ed.), *Proceedings of the 8th European Conference on Games Based Learning - ECGBL 2014* (pp. 339-345).

Miljanovic, M. A. and Bradbury, J. S. (2018). A review of serious games for programming. In Joint International Conference on Serious Games (pp. 204-216). Springer, Cham.

Naul, E., & Liu, M. (2020). Why Story Matters: A Review of Narrative in Serious Games. *Journal of Educational Computing Research*, *58*(3), 687 707. https://doi.org/10.1177/0735633119859904

Schöbel, S., Saqr, M. and Janson, A. (2021). Two decades of game concepts in digital learning environments – A bibliometric study and research agenda. *Computers & Education*, Vol. 173, ISSN: 0360-1315, DOI: https://doi.org/10.1016/j.compedu.2021.104296.

SQA Assessment Futures (2018). Young Scot Report. Retrieved March 2020 from: https://www.sqa.org.uk/sqa/files_ccc/221018_SQAAssessmentFutures_YoungScotReport.pdf

Zyda, M., (2005) From Visual Simulation to Virtual Reality to Games, Computer 38, pp. 25-32.