Final Year Project Plan

Full Unit - Project Plan

Building a Game

Submitted By Sulaimaan Bajwa

Created in order to plan out the final project critical to the completion of the 3rd year of

Msci (Hons) Computer Science with Software Engineering

Supervised by Hugh Shanahan

EPMS School, Department of Computer Sciences

Royal Holloway, University of London

Contents

1 Abstract

Video games are an almost unavoidable applied form of computer science. There exist educational tools (Mojang, Microsoft, 2016) that also take the form of video games, created especially for their ability to captivate the mind of the young audience, and get them interested in the topics brought forward through creative fun and problem solving.

By definition, a video game is ‘an electronic game in which players control images on a video screen’ (Merriam-Webster, 2022). True to this definition, games are often much deeper than just sprites, often containing problem solving, a progression system, collections and completion rewards, and specific challenges for exceptional players. To master these individual devices, with the use of exceptional programming can yield incredibly enthralling games, in their responsive, accessible and rewarding nature.

There has been a recent surge in popularity within the rather specific genre of roguelike games. Commercial successes like Hades (Supergiant Games, 2020), Cult of the Lamb (Massive Monster & Devolver Digital, 2022), Dead Cells (Motion Twin, 2018) and Slay the Spire (Mega Crit Games, 2019) stand on the shoulders of game franchise The Binding of Isaac (McMillen, 2011).
All of these games share the common properties of exploring some generated or hand crafted space, collecting objects that alter the player - for better and/or for worse, and overcoming - or dying to - increasingly difficult foes/challenges to the player. Generally the game cycle is repetitive, and built on the motivation of a player wishing to get further than they managed in the previous attempt, or run (Dotson, 2020). Games like these are often fundamentally simple, with basic movement controls, collection of items, and some form of combat control. The challenge found within the game is entirely based on the combinations of items that the player chooses to collect, and the combat mechanics of enemies, both of which will largely need fine tuning during play testing.
In order to play the game for a potentially indefinite run, the game would require some form of indefinite scale to the values associated with both the player and enemies. This difficulty can be represented on a graph, with the magnitude of difference in the gradient of lines representing both the player and the foes representing the gradual increase in actual difficulty.

The final goal of the project will be to have formulated, built and deployed a game, making sure to utilise design patterns and complex technologies. In order to achieve this, I need an understanding of which design patterns exist in the industry, and how they are generally included. A good example of such a design pattern may include the use of controllers for handling specific objects, and factories to make objects.
I’ll also need to pay special attention to technologies and standards used within the industry. An early assessment of technologies has thrown Unity as a good way to work practically with games, having both a very detailed, interactive approach to the front end of the game development cycle, and an even more granular C# module for integrating the back end into the game building environment.

My own goals for this project, in addition to what is expected of me, includes learning new programming skills as I pick up C#, and start to understand Unity as an engine, as well as the software through which I interact with the engine. To create and publish a game will also help build my portfolio of presentable programming to potential employers, or future team members. These goals will end up forming the basis on which I am capable of defending my project decisions and software design choices in the interim evaluation.

2 Timeline

Academic Week	Milestones	Specific Notes
Term 1		
Week 1 - 3 19/09/2022-03/10/2022	Write project plan	Meeting Tuesday 27th,15:10 7th October Project Plan Deadline
	Research currently existing games of the targeted genre	
	Work on gaining familiarity with the targeted engine	
	Conduct research on breakdown of important game mechanics, like visuals, motion, items and difficulty	
Week 4 - 5 10/10/2022 - 17/10/2022	Create an early prototype for the game, featuring motion, damage, all major collision and simple items	
	Research on how to devise an algorithm for keyed procedural generation	
	Identify missing research areas	
Week 6 - 8 24/10/2022 - 07/11/2022	Work on Interim Early Deliverable, modifying the prototype	
	Start creating a method for breaking down the composite pieces of complex items, and applies the effects of them in a logical way to the character	
	Create an algorithm based on procedural generation research, represented simply	
Week 9 - 12 14/11/2022 - 05/12/2022	Work on interim report	December 2nd submit interim report, code and presentation materials December 5th presentation
	Work on interim presentation	
	Practice presentation	
	Start working on procedural generation algorithm, to create rooms of a set size, and of a set number of rooms	
Term 2		
Week 17 - 19 09/01/2023 - 23/01/2023	Create template rooms to potentially insert into randomised maps	
	Complete algorithm that randomly generates map	
	Attempt to integrate the template rooms into the randomisation algorithm, using both a set of required rooms, and optional rooms	
Week 20 - 22 30/01/2023 - 13/02/2023	Work on character visual modification algorithm	
	Work on enemy set templates	
	Attempt to integrate the template enemy sets into the randomisation algorithm, using both a set of regular enemy sets, and boss enemies	
	Draft up Final Project Report	
Week 23 - 25 20/02/2023 - 06/03/2023	Finalise programming side of project for submission	
Week 26 - 27 13/03/2023 - 20/03/2023	Finalise Project Report and submit all project documents	March 24th Final submission for all project documents

2.1 Gantt Chart (Goes here, 10 minutes job)

3 Risk Assessment

Risk	Impact	Mitigations
Data Loss	Potentially very high - potentially all of my project files	Backups, Storage in GitLabs
Stretch Resources/High Cost/Time Crunch (time, skill, money, hardware)	Potentially very high - behind on research, deliverables etc.	Drafting and regularly updating a timeline/gantt chart and budget if required
Scope Creep (Team Asana 2021)	Potentially very high - lack of focus on important deliverables/research	Keep an active trello board, add further developments to trello board
Low Performance (Team Asana, 2021)	Due to my current scope being rather small, performance issues with likely not happen, and if they do, they will be relatively non-impacting	Continuous testing/comparing several versions of the prototype to identify the cause
Operational Changes (Team Asana, 2021)	Medium to High, there are cases in which i may find a more efficient, or otherwise better API, module or framework, which may lead to an adaptation period of low output	Identify whether i have the time to learn new skills before deciding to incorporate them into the project
Lack of Clarity (Team Asana, 2021)	Low, I should know roughly when work is expected of me, and how much to produce	Have work done for a week before i may expect them to be due, in order to compensate for delays due to vague communication

4 References

Massive Monster & Devolver Digital. (2022, August 11). *Cult of the Lamb on Steam.* Steam. Retrieved September 26, 2022, from https://store.steampowered.com/app/1313140/Cult_of_the_Lamb/

McMillen, E. (2011, September 28). *The Binding of Isaac on Steam.* Steam. Retrieved September 26, 2022, from https://store.steampowered.com/app/113200/The_Binding_of_Isaac/

Mega Crit Games. (2019, January 23). *Slay the Spire on Steam.* Steam. Retrieved September 26, 2022, from https://store.steampowered.com/app/646570/Slay_the_Spire/

Merriam-Webster. (2022, September 16). *Video game Definition & Meaning.* Merriam-Webster. Retrieved September 26, 2022, from https://www.merriam-webster.com/dictionary/video%20game

Mojang, Microsoft. (2016, November 1). *Minecraft Education Edition.* Minecraft Education Edition. Retrieved September 26, 2022, from https://education.minecraft.net/en-us

Motion Twin. (2018, August 6). *Dead Cells on Steam.* Steam. Retrieved September 26, 2022, from https://store.steampowered.com/app/588650/Dead_Cells/

Supergiant Games. (2020, September 17). *Hades on Steam.* Steam. Retrieved September 26, 2022, from https://store.steampowered.com/app/1145360/Hades/

Team Asana. (2021, August 18). *7 Common Project Risks and How to Prevent Them • Asana.* Asana. Retrieved September 26, 2022, from https://asana.com/resources/project-risks

5 Mark Scheme

5.1 Abstract: 30 marks

Percentage	Description
< 40%	Copied from the project list
40%-69%	Describes and motivates the final project
70%-100%	Describes and motivates the final project. Gives well thought out individual project goals

5.2 Timeline: 40 marks

Percentage	Description
< 40%	Milestones missing, poorly thought through or just copied from project list
40%-69%	Milestones are adequate or good, each with dates but some have insufficient motivation
70%-100%	Good list of well explained milestones

5.3 Bibliography: 15 marks

Percentage	Description
< 40%	Bibliography missing, unusable or just includes web pages.
40%-69%	Clear use of more than one source. Not just web sites. Nice evidence of good research.
70%-100%	Bibliography items are very good. Each has a short relevant discussion of its value to the project

5.4 Risk Assessment: 15 marks

Percentage	Description
$<$ 40%	General Risks just given such as "may get behind" .
40%-69%	Risks associated with deliverables and show clear thought. Ideally each (or some) risks are provided with mitigations that seem reasonable.
70%-100%	A good case is made for understanding how the project might fail to proceed and what should be done about it. Some risks have likelihoods and importance.

6 Project List Points

Aims: To understand the game development ecosystem and to build and deploy a playable game.
Background:
Games development is one of the largest parts of the computing industry. There are many platforms for writing games and indeed, many architectures and frameworks.
In this project you will build and deploy a game.
In the first term you will research existing games and their underlying technology platforms. You will also look at game building tools such as Unity3D. You will consider game frameworks such as PyGame and also investigate APIs for game development such as Vulkan.
You will need to be familiar with the design patterns required for gaming: Flyweight, publish/subscribe (for message passing), Object Pool, Service, Dirty Flag, Factory, Visitor, Singleton, Game Loop, State, Observer, Command, Event Queue (See perhaps Game Programming Patterns by Robert Nystrom. Also consider reading some of the material on The Game Designs Patterns Wiki
Perhaps you will use Blender and other tools and do some maths and produce an excellent Unity game.

6.1 Early Deliverables

1. A simple program displaying an animation using sprites or similar animation technology.
2. A program displaying a simple, interactive user interface. You might want to explore the different means of displaying and writing user interfaces (XML, Object Orientated, Immediate Mode).
3. 3D demo using an API
4. A program that exhibits some 3D graphics using a relatively low level graphics API (Vulkan, OpenGL, WebGL, JavaFX etc. This might provide a starting point for choosing technology: https://en.wikipedia.org/wiki/List_o
5. A simple game using a game engine / library / framework. Think of it as an opportunity to learn about the technologies you'll use for your main game but keep it simple.
6. A report on design patterns in games, an enumeration of commons ones, the problems they solve and perhaps discuss some specific instances of their use.
7. A report on the specific technologies you've tried / intend to use for your game. You should do deep on you intended tech, show as deep an understanding as you can.
8. A report on animation techniques, enumerate them, perhaps tell a history of them, compare and contrast. The goal is to show a deep understanding of how animation in games is achieved.
9. A report on Threading an User Interfaces. What problems do games experience with user interfaces without threading, how does multi threading solve these problems and what different ways can this parallelism or concurrency be achieved? Perhaps review how it has actually been achieve in technologies or platform relevant to your game.
10. A report on designing games with graphical user interfaces for multiple platforms.
11. A report on the technologies used to write and draw graphical user interfaces.

6.2 Final Deliverables

1. Game Source Code: You should use a consistent architectural style. Our suggestion is object orientation but you could opt for another with sufficient justification.
2. Game Source Code: The game needs to have at least 3 'screens' or pages that are interactive using the GUI technology of choice.
3. Game Source Code: You should publish your game! (Android Store, itch.io)
4. Game Source Code: The game play should make use of animation.
5. The Final Report: An in depth review of background materials you used to learn the concepts and technologies used to write the game, perhaps presenting a historical timeline of those concepts / technologies.
6. The Final Report: A description of the more difficult concepts involved in writing graphical games e.g. Multi Threading and threads / processes, event based systems (handlers, events), special consideration for the environment your game with run in (memory foot print, battery usage).
7. The Final Report: A description of the software engineering processes involved in writing your game.
8. The Final Report: A description of interesting techniques, hacks or data structures you used (or could have used) in writing your game.