CRITICAL EVALUATION OF CULT OF THE MERGER

Will Bennett – b014262k

Table of Contents

Glossary

GDD – Games Design Document

GPC – Google Play Console

Key Words

Mobile Gaming, Games Industry, Games Development, Monetisation, Prototyping, Ethics

Introduction

This project is creating an idle mobile game that has the player merging minions to raise their cult's standing with another worldly entity. The plan was to create a game that has potential to be submitted to an app store. This meant that wide range of considerations had to be taken to ensure the project has the ability to launch smoothly and stay afloat through future proofing the game. This report will detail some of the key details related to the journey of the project, including any decisions that were made and their effect on the project. This includes decisions that did not make it into the final project.

Reflective Evaluation

Initial Design Decisions

The core gameplay for this game is for the player to spawn and merge crafting components together to form minions. These minions would then generate mana for the player to further expand their cult. These minions can then be sacrificed to increase the power of the cult. With expansion comes popularity, the player will also have to send their minions to battle against intruders in their lair. This introduces a core mechanic, resource management. With quick expansion the player will find themselves planning which minions to focus on merging together, and which to sacrifice to gain back limited space in their cult. This combined with offline progression which is explored in greater depth in the games design document (GDD) (Bennett, 2022) in appendix one, allows the player to plan out how long each offline session will be to reduce the potential loss of resources. Meanwhile this gives birth to the secondary gameplay genre, a puzzler. This mainly occurs when the player is actively playing the game. As mentioned, the player has limited space so they must decide which is the best use of their space.

This overview outlines key decisions made towards the game, mechanically, and artistically. Mechanically the game will need to be able to effectively create and destroy objects, while allowing the player to interact with the objects by dragging them around the scene and using them as crafting materials. This means pawn progression is needed, which will form the basis of a pseudo crafting system. As the game is planning to use offline progression, the game must be able to track the number of materials the player could have earned within the time away from the game. Artistically the game will need to create a cult like environment to immerse the player, this will help make the short burst of gameplay more enjoyable and feel less like a chore.

Design Patterns

Which design patterns can help with the creation of the mechanics previously mentioned. One of the key patterns is prototyping. Prototyping refers to being able to create duplicates of an object, while keeping performance in mind (tutorialspoint, N.D.). Due to the game's core mechanic being related to creating objects, it heavily relies on the premise discussed in the prototype design pattern. This project aims to save on memory through the use of scriptable objects. Scriptable objects are used as a way to cache information about an object and then this information can be shared to multiple objects. This could have been taken a step further and used prototyping in a traditional way, since this would have saved a larger performance of the game than just using scriptable objects alone. Especially as it is easier to implement than other patterns such as Object Pooling since predicting how many of a certain object a player would need can be hard to do so. This often leaves the creation of the maximum number of objects being created which can be a drain on performance and memory of the game, especially if not done properly. This is amplified on mobile devices due to the lack of resources available. Therefore, prototyping was used since it frees up memory as all the objects hold onto one shared reference.

Observer pattern (Refactoring.Guru, N.D.) was also implemented through the use of an event system. This builds upon Unity's event system, which allows for events to be subscribed to and sent the event to its subscribed listeners. This was implemented using ids, which each pawn gets their own unique id on creation. The main purpose of this is to allow pawns to know when they need to level up. This will allow all the other components attached to the pawn to update their stats simultaneously. The use of the observer pattern reduces the need to have a direct reference to other classes, which is beneficial since not all pawns will have the same number of components attached. The implementation of this pattern is somewhat limited with some direct refences having

the possibility to be replaced by an event call. These references were kept as references to reduce the number of events that are being fired to help performance.

As mentioned, the use of prototyping was a large influence on the project. The main implementation can be seen in the buildings script, where all the pawn-based spawning is done. For this when a building is tapped, the script will work out which cost it needs to take away from the players inventory, and which attributes it needs to set for the pawn to gain its functionality. All this data is pulled from the scriptable object. This allowed the pawns to be set up as the clones of each there. This was set up through using prefab variants of a pawn that has minimum functionality such as movement and combination possibilities. Each variant expands on this original variant by adding more components and setting then to complete their specific function. Being too focused on implementing a design pattern can cause harm. An example of this is where prototyping was of the highest use in this project, as mentioned is the buildings script. Initially when creating the scriptable object, it had an effect of trying to add all data to all the data related to the pawn's creation to it instead of creating variants to handle the differences. This approach was changed when the scale of the project got larger, and instead the amount of data the scriptable object holds is increased when needed.

Monetisation and Retention

To monetise the game, a system of gems and coins was implemented. The player can gain these naturally through the course of playing the game. But they can also purchase them through in-app purchases, this was only set up as dummy logic. The idea behind these currencies, is that the player would be able to buy items to speed up their progression, an example of this would be mana potions that would refill the player's mana stores. Another way to make progress in the game is through advertisements. Allowing the player to watch an advert to refill a storage allows players to skip the offline period of the game. This has not been implemented as of writing this document.

To give the player a reason to come back to the game at least once a day, is through a daily gift. This daily gift is free and is currently set up to be claimable every 2 minutes for testing purposes. Another reason for the player to come back to the app, is the potential loss of mana being generated when their storage is filled. This would be an opportunity to send out a notification to the player, to warn them that their storage is filled.

Ethical considerations

The premise of the gameplay for the game is very simple to grasp due to it being able to be boiled down to combining two of the same elements or "pawns" to upgrade its values. This means that any demographic can play the game. Due to this, some considerations were made about player health. Games can be addicting, which can cause a group of players to constantly play a game an unhealthy amount. To avoid this, most of the progression in the game is made when the player is not playing the game. This forces the player to stop playing. While this can force the player into interacting with the monetisation mechanics such as using gems to refill their mana, most players would rather wait for the refill to occur naturally.

The original plan was to have the player sacrifice their minions on an altar. This was to further push the idea that this is a cult. That being said, if a younger audience gets hold of the game this may be considered too violent. Therefore, the altar was swapped out visually for an otherworldly portal. This separates the idea that you are killing the pawns. This also helps reduce the classification rating of the game from boards such as PEGI (PEGI, 2023). With the removal of this, the game would likely score a PEGI 7 rating due to the mild forms of violence, since the minions are used to kill enemies,

this combat is not realistic. The design of some of the enemies such as zombies, is the reason the rating might get pushed up to a PEGI 7 rating, despite the depictions of them being cartoony.

Changes from initial plan

One change from the initial design created in the GDD (Bennett, 2022), was the SDKs and APIs that were going to be used within the project. Although the GDD mentions Google's APIs that can be used in creating a game, it only mentions Google's in-app purchasing system. This was expanded to use Google Play Console (GPC) (Google, 2023). This console allows the collection of analytical data, setting up achievements and leader boards through Google Play (Google, 2023). Through the use of GPC, the game is able to easily integrate into Google's ecosystem. This API, offers a wide range of functionality such as leader boards that your friends can see, or a method to set up testing the app. It can even go a step further and help the game's discoverability through the store presence and store performance sections. This can help maintain the game has good player acquisition and be a way to successfully implement and evaluate the success of the strategies initially discussed in the player acquisition strategy section of the GDD. Although some of the functionality that GPC offers is already offered in Unity, such as analytical data, the benefit of using the API instead is that all the data will be in a centralised location plus it speeds up development time. As mentioned, Google Play offers a way to easily add testers through Google Play, this can help test the game before and after launch before new features are implemented. This will help the game in the longevity as allowing the player base to easily test the game's features will allow insight into the game. GPC also helps with monetisation setup through their system. This can help set up an economy around the game and to speed up the integration of setting up actual logic behind it instead of using dummy logic. The API also gives access to setting up achievements which was not considered in the GDD, using GPCIDs, the player's actions can be linked up the Google play console which can allow the developer to easily set up points scored from the achievements. This allows players to compare their progress of achievements with their friends while also appealing to "completionists", who would aim to get every achievement in the game.

The GDD also plans out features that did not make it into the current build of the game. An example of this would be a tutorial. The GDD talk about setting up a first time play that would talk the player through learning the core mechanics. This did not manage to get implemented but would need to be implemented in the future as it makes the game's learning curve a lot smoother and the game accessible to its targeted audience. Some gameplay additives features did not get implemented, an example of this is demands which can be found under What is your cult section of the GDD. These can be implemented to the player as they progress and level up their cult. This would be a way to motivate the player into progressing through the game.

Future of the project

Due to the game structure, allows for future development of the project. This is heavily due to the scriptable object reliance. Theses scriptable objects are open-ended which allows future expansion easily, an example of this more minion types can be added easily through adding new types to pawn definition, or by increasing the number of entries within object data. This easy expansion also applies to how scripts were made. Each approach to a script was for it to be used as a component. These components can then be added to a pawn to give it additional functionality. An example of this is minions are the only pawn within the game that has the minion script attached, this script gives it the ability to deal damage to other objects. An additional application of the use of scriptable objects is for the game to reflect real life events such as Christmas or New Years. Since the visual representation of a pawn is saved to a list, variations of these progression can be added. These variations could then be swapped out to celebrate seasonal events or to equip skins that the player

has brought.
Additionally to increase the amount of progression and depth in the gameplay, the features that were not implemented in time could be added. This would be easy to do due to some of them having part of their functionality included. An example of this is consumables such as mana potions. Not only will this act as a good reward for the player, but it will also be able to be purchasable from the shop with coins.

Bibliography

Bennett, W., 2022. *Games Design Document for Cult of The Merger.* s.l.:s.n.

Google, 2023. *Google Play Console.* [Online]
Available at: https://play.google.com/console/about/
[Accessed May 2023].

Google, 2023. *GooglePlay.* [Online]
Available at: https://play.google.com/store/games?hl=en&gl=US
[Accessed May 2023].

PEGI, 2023. *PEGI.* [Online]
Available at: https://pegi.info/
[Accessed May 2023].

Refactoring.Guru, N.D.. *Observer.* [Online]
Available at: https://refactoring.guru/design-patterns/observer
[Accessed May 2023].

tutorialspoint, N.D.. *Design Patterns - Prototype Pattern.* [Online]
Available at:
https://www.tutorialspoint.com/design_pattern/prototype_pattern.htm#:~:text=Prototype%20pattern%20refers%20to%20creating,ways%20to%20create%20an%20object.
[Accessed May 2023].

Appendices

Appendix 1: Game Design Document

GAME DESIGN DOCUMENT

Will Bennett, Cult Of The Merger

Abstract

Summary of your game. The paragraph you'd probably put on the App Store/Play Store.

20014262
B014262K@student.staffs.ac.uk