Gamification Unity Documentation:

This document will serve as an overview for the frontend side of the Gamification project. It will not dive too deep into technical details of the project, but should give a beginner an idea of how the current system works as well as the Seeker, Sudoku and Conqueror games currently function. I implore the next team who works on this project to read through the documentation as well as all of the scripts on the server and Unity side of things to get a big picture before making changes, as sometimes making a change can have unintended consequences. If anything is unclear, feel free to email me at nigel.haney27@gmail.com and I can help track down the source of the error as I feel like I have, at this current moment, a good understanding of the codebase of this project.

Overall System Explanation:

There are various components to the system present in the Gamification project, in this section I will introduce them individually.

Active/Passive Mode: Active and passive mode are put in place to limit effective playtime of the students. One goal we had in mind was not to allow students to play all day and be rewarded for it, the substantial amount of rewards should be gained by participating in class. In active mode, rewards are increased and the more valuable “active coin” is gained. The playtime clock at the top of the screen indicates how much “active time” the user has left in their session. This clock can be restored from in-class rewards. Also the progress bar will not reward the player in active mode unless the user makes an action, passive mode will fill no matter what if the player is at the keyboard or offline.

Progress Bar: Every time this bar is filled, you get experience and passive/active coins depending on what mode you are playing in. The amount you get is displayed to the right of the bar.

Upgrade Shop: Spend active or passive coins to increase the rate at which you progress. Also from here you may ascend after reaching level 30.

Boosters (potions): These will increase the exp or coins gained while leveling or your overall progress rate. Potions are gained by leveling up, the player may choose what type of potion they want at each level.

Stamina: Stamina is consumed by initiating the gameplay loop in the minigames.

Ascension: After reaching level the 30, the user may choose to ascend. Ascending increases the overall rate of progress gained as well as position on the leaderboard. When you ascend, you also gain a perk point in a game of your choice, which is meant to increase your power in that game, without affecting the incremental as a whole.

Leaderboard: Used as a method of motivating competitive students, the leaderboard displays the top 10 players based on Ascension and level. This could be expanded to a bigger list in the future, so you can see your player on the leaderboard if you are not within the top 10 players.

Server Interaction:

The server interaction in Unity involves exchanging the following classes which get serialized and deserialized by the game and server on both ends:

```
IncrementalData - found in Incremental/Scripts/update.cs

sudokuData - found in Mastermind/Scripts/Test/SudokuControl.cs

ConquerorSave - found in Conqueror/Scripts/PlayerShip.cs

seeker - found in Seeker/Scripts/SaveLoad/GlobalControl.cs

DaredevilSave - found in Daredevil/Scripts/ddPlayer.cs

SokobanSave - found in NewMasterMind/Scripts/GameManager.cs
```

To make a change to how saving works, additional fields can be added to the above classes if more information needs to be saved and the saving should work on the server. If a new class needs to be added, additional code must be added both on the Unity frontend and on the server. The way to do this is specified in the following files in Unity:

```
update.cs
playerInfo.cs
```

And these files on the server:

```
Database.cs
MongoDatabase.cs
Client.cs
```

Seeker Documentation

Disclaimer:

This game would be very hard to explain completely in a document. The best advice I can give for understanding it is to play through it and look at all of the scripts that are active in Unity. Most of the code is well commented, but it is structured very poorly. A rework/refactor of this game might be a valuable thing to take on for future groups.

The game starts out with a scripted tutorial segment to introduce players to aspects of the game. After this is played out, the main game loop begins.

Game loop description

The goal of the game is to go into dungeons of varying difficulty, get loot, and escape safely in order to sell the loot back in town. There is a main quest with two seperate objectives for now, more could be added in later. Various power-ups and active coins can be bought in town with the in-game currency of "Reputation". The harder the dungeon, the more stamina that is required to enter and the loot obtained in the dungeon is more valuable.

Future tasks:

Extra quests can be added to the game to make it more playable as well as varying dungeon types. Combat needs to be reworked to be more engaging to the player. A complete overhaul of the game is possible, but an expansion might make more use of the existing content. I was thinking to possibly zoom out the game for story purposes to make the current game a game within a game within the Incremental game. You could add a more futuristic setting where you can choose to go back into the world that has been built up for now or maybe even others as you learn more about the VR company mentioned in the intro. Overall, the game is playable but is just lacking replayability.

Sudoku

This game is fairly basic and straightforward with it’s implementation. Again, like the Seeker game, the best way to understand it is to play it and look in the comments in the code.

Game loop description:

The player can choose to fixed mode or random mode. This will either start a randomly generated board or a fixed board of varying difficulty. When completing a board the player will get rewarded with active/passive coins. Stamina is depleted every time a puzzle is started.

Additional Information:

Throughout the past year this game has been touched the least by our team, as it always has been functional unlike the rest. It is however, the most forgettable game and probably the most boring to the user. Additional features will need to be added in order to make the game fit in better with the Incremental model.

Perk Point Problem:

A perk point addition to this game has not been decided upon yet. Future teams will need to think of a way that players can meaningfully ascend to make the Sudoku game more fun to play.

Conqueror:

This game is about progressing through levels and shooting up the boss in each level. The game takes place in one scene and I will describe below how the game functions.

Gameplay Loop:

The player consumes a stamina point by starting this game. The player can choose which level they want to start out with, based on what level they have reached before. From the main menu the player can also choose which gun they would like as well as which powerup they would like to use. A powerup is on a cooldown and can be used by hitting the space bar. Every time the level is loaded up a boss with new stats is also loaded in. The bosses stats scale with the level, which can be found in Boss.cs. Every time the player defeats a boss, it drops an item. The item contains a different type of gun the player may use and also will reward the player with passive and active coins when picked up. There are currently three types of guns, which can be viewed in GunScript.cs, making a new gun archetype would be an easy addition to add more content to the game. When you die, you are brought back to the main menu where you can change your weapon, powerup, and level to start on.

Future Tasks:

Right now the game is fairly bland, and not very replayable until you get a new perk. Improvements could be made with balancing as well as replayability. A different kind of upgrade system would work very well in this game, as well as a wider variety of guns and enemies. This game even has the potential for online multiplayer, but that would be quite a large undertaking.

DAREDEVIL AND SOKOBAN COVERED IN OTHER DOCUMENTS

Future Tasks to Complete Overall:

In it's current state the game is fairly playable, but could still use some improvements to make it more fun to play for the whole semester. The main point that needs to be made by the next team taking on this project is to implement some of the features that allow it to tie into class, as that is the underlying purpose of the entire study. In my opinion this should be ironed out first, and then the individual games could be looked at for improvement. Polishing of the tutorial of the game as well as each individual game should be done before deployment in a classroom.