Design Project Final Report Summary

CS 440 Group 12 - Roger Chiu, Hunter Olson, Seth Traman, Ryan Vincoy

The Underground Civilization Game (UCG) is a 2D idle game with multiplayer mechanics. There are three main aspects to UCG:

- The Main Game
- The Marketplace
- And the Dungeon

Main Game

In the Main Game of UCG, the player will be able to travel to different layers of this underground world. Here, the player can idly mine ores and cut down roots for wood to gain resources for items and gear. With the resources the player collects, the player can craft items and gear. Throughout the game, the player will earn experience points and once they reach a certain amount, they will increase in level which will allow them to reach deeper depths and also earn skill points. With these skill points, the player can unlock skills and abilities from the skill tree. These skills and abilities could increase the strength or health of the player. They could also allow the player to learn magic spells which could be used for combat in the dungeon.

Marketplace

In the Marketplace of UCG, the player can join a marketplace lobby where they will be joined with other players. Here, the player can talk to other players through an open-text chat and make trades with other players. A player can look at other player's inventories and if they see something they like, they can request an offer to trade with that player. If accepted, they will enter a bargaining sequence where each player will put in an item/gear/resource they want to trade. If both players agree, these items will be switched in their inventories.

Dungeon

In the Dungeon of UCG, the player can join a group up to 4 (or they could go solo) and venture into the dungeon section of the game. Here, the player will have to battle enemies either idly or manually with their group using skills they've unlocked from their skill tree and gear they built from resources they've earned or traded for. A dungeon level will have a certain amount of enemies and once the player's group clears that level, they can venture deeper into the dungeon. Enemies scale to the depth of the level so the deeper into the dungeon the player goes, the higher the difficulty rises of the enemies but with difficulty comes better loot drops. Each enemy has the chance to drop certain loot which can either be gear that can't be crafted or resources that can be used for crafting. Enemies also give experience points and similar to the loot drops, the harder the enemy, the more experience points they drop. The player's group can leave the dungeon if they'd like or they'll be forced to leave the dungeon when the whole group is defeated. Once defeated, the players will return back to the Main Game where they will continue to have all the loot they earned from the dungeon, however items they used, like potions, will disappear from their inventory.

As mentioned in previous sections, the players can do certain tasks idly. What that means is that the player will be able to be away from the keyboard (AFK) and still progress through the game. There are two main AFK algorithms that this game will employ:

- Resource Gathering
- And Dungeon Combat

Resource Gathering
With resource gathering, the player will be able to gain materials like wood or ores by idly standing next to ores or roots in the Main Game. There are certain sections in the Main Game where the player will be allowed to gather resources and that is in the ore cave or the root forest. If the player is in either, the player will automatically start gathering resources without the need of input from the player.

Dungeon Combat
With dungeon combat, the player has the option to either manually fight with player inputs or switch to an AFK mode where the game will use an algorithm that is optimized to keep the player alive and defeat enemies. This algorithm will prioritize keeping the player alive and choose the best attacks to defeat certain enemies. For example, if there is an enemy that is weak to magical spells, the algorithm will choose to attack the enemy with a spell rather than a physical attack. Another example is that if the player is low on health and the player has health potions in their inventory, the algorithm will make the player use those health potions to keep them alive.

Who are the Customers?
This game is for people who are busy with things like work and school. UCG allows the player to make progress in the game while the player is working on other things in the real world. For example, a student in college could be working on assignments for classes and on their laptop in the background, they could have their player mining or going through the dungeon in AFK mode.

Requirements
Functional requirements include implementing the AFK algorithms mentioned previously. Other fundamental requirements include implementing the multiplayer mechanics for the marketplace and the dungeon, the skill tree mechanic, and the crafting mechanic. Other important requirements include performance requirements, specifically for the dungeon and marketplace which have multiplayer mechanics so speed is important so the players can get practically real-time responses. Reliability requirements are also important for video games so the game needs to save consistently so if the player closes out suddenly, they can return back with all their progress from their previous session saved.

Design
For the architecture for some of the game mechanics mentioned above, the development team could implement a server-client architecture for the multiplayer to prevent any cheating or hacking. For the spawning of enemies in the dungeon, the development team could employ the abstract factory design pattern. For the inventory of the player, the development team could create an inventory space interface which is implemented by classes like Gear, Resource, Item, Potion, etc. which will allow all types of things to be stored in the player's inventory.

Reusable Components
For the creation of UCG, the development team could use game engines like Unity or Unreal Engine. Using a game engine will allow the team to focus on functionality rather than building a game engine from the ground up. Also with game engines like Unity, they support multiplayer services which will make employing the multiplayer mechanics easier.