|  |  |  |
| --- | --- | --- |
| Ney Congjuico | 12432102 | [ncongjui@uci.edu](mailto:ncongjui@uci.edu) |
| Zachary Hart | 70953123 | [zhhart@uci.edu](mailto:zhhart@uci.edu) |
| Daniel Lara | 49651280 | [larad@uci.edu](mailto:larad@uci.edu) |
| Tyler Pearce | WHAT’S YOUR ID | [tpearce@uci.edu](mailto:tylerpearce@uci.edu) |
| Avelino Miranda | 16732033 | [acmirand@uci.edu](mailto:acmirand@uci.edu) |

**Single Player Game Architecture**

Prefabs Folder

* Contain various prefabs that deals with the in-game models such as the player model and walls. Aesthetically pleasing models will be added in the future, current tiles are just placeholders.

Resources Folder

* Input subfolder is used to hold the directional movement inputs and bomb deployment input, in this case spacebar or ‘jump’.
* Prefabs that contain the bulk of the written script components are found in the Managers subfolder.
  + GameManager: In charge of keeping track of the number of players there are currently in the game, round time, and checks if a ‘victory’ condition is met and end the game.
  + InputManager: Receives number of players from GameManager and then adds each player to a private list containing Players. Since it is a single player game as of right now, it only handles one user input.
  + MapGenerator: Takes a .txt file which contains a 2D grid of a map. Each number represents a different type of prefab. Map generation is done in a bottom to top order.
  + SoundManager: Handles the sounds found within a scene.
  + Spawner: Deals with the generation of bombs and explosions within the game scene. Created when the user deploys a bomb.WindowManager: The prefab oversees the number of window prefabs currently available and determines which canvas is active or not. High level explanation – each prefab of a window is toggled on and off.

Scenes Folder

* Holds all the scenes currently being used/tested.

Scripts Folder

* Controls subfolder contains the various scripts that involve user input. Used in InputManager.
* Effects Manager subfolder deals with the bomb/explosion spawning effect and sounds.
* Game subfolder contains the explosion logic which checks whether the explosion of the bomb is blocked by an indestructible wall or it destroyed a valid space.
* Map Generation subfolder contains the scripts (MapGenerator) in which the .txt file is read and generated into the game scene.
* Player subfolder contains the PlayerScript which determines the color a user will have. Since it is single player always defaults to the first color which is red.
  + Startup Scripts subfolder contains the StartManagers script. A script component found in each Main Camera game scene, simply dictates which Manager is enabled at the start of each scene.
  + Windows subfolder has all the different scripts for each different canvas. Each script is in charge of toggling off their current window and toggling on the next desired window.
  + Singleton script allows only one instance of a specified object derived from the Singleton class to be initialized. Allows any script to retrieve the instance of a game object and, if one is not already in the scene, automatically create one.
  + ButtonBehavior script is attached to the various menu buttons found within the main menu. This is to ensure the correct button will be selected when going from various menus and to animate a highlighted button.
* How it all works together?
  + Start Managers (attached to the Main Camera) is initially called and begins with the WindowManager. The WindowManager handles all of the different canvas within the main menu and game scene. The script’s purpose is to determine which canvas is toggled on or off dependent on user input. Once the game actually starts there are three Game Objects that are initialized. The Map Generator begins to populate the game scene with tile objects. The map created is determined by a .txt file found within the Resources folder. Secondly, the Input Manager is created which handles all the logic side of the player inputs such as moving and placing bombs. Lastly, the Game Manager is created, however it is currently not being used in the current build of the project. Our group found it unnecessary to implement an overseeing manager that kept track of only one player. The Game Manager will be used once the multiplayer feature is added. As of right now, the only win condition, since there is only one player in scene, is for the player to die by their own bomb.