**OVERVIEW**

The project is a Pac-Man vs AI game built to run on multiple computer platforms. The user is first presented with a main menu where they can choose to play, see the rules, adjust volume, or quit.

The game requires two mazes: one for the player (user) and one for the computer (AI). The mazes are side by side. Each maze has a scoreboard over the maze with the score and the number of lives remaining. The game is timed using a single timer.

To play the game, the player controls their Pac-Man using the arrow keys. The AI Pac-Man is controlled utilizing a pathfinder to go to the nearest dot and avoid ghosts. The game is complete when either the Player or AI has collected all the dots, or one has lost three lives. Both the Player and AI have four ghosts that move around randomly trying to get the Pac-Man in their respective mazes. When a ghost captures Pac-Man, they (user or AI) lose a life. Upon losing a life, Pac-Man and the ghosts respawn to their starting positions.

When the game is complete, the user is shown a result screen with the final scores and time elapsed. The user can choose to play again or return to the main menu.

**HOW TO USE SOFTWARE**

1. Download Pac-Man zip file in the most recent release of the software on Git Hub
2. Extract the files from the zip file to a directory
3. Open the folder containing the extracted files
4. Run Pac-Man.exe

**QUALITY ATTRIBUTES**

*Usability* – A player should be able to understand how to play quickly in order to limit the time spent learning the game from trial and error.

*Reliability* – The game needs to respond correctly to each keystroke the player inputs to have the game be fair for the player.

*Integrity* – The game needs to run smoothly so a player can appropriately respond to the game and to feel the game is fair.

**SOFTWARE ARCHITECTURE**

Need two diagrams + 1-2 pages written

**DESIGN PATTERNS**

Observer

Abstract Factory

**FINAL STATE**

Known bugs/issues

* AI Pac-Man will sometimes “teleport” positions when a ghost gets close
* AI Pac-Man will sometimes not recognize the last dot in the maze and stop

Product Backlog

* None…

**FILE STRUCTURE**

The project source code and objects are in the Assets directory separated into directories containing related items:

* AstarPathFinding Project – Scripts for AI movement
* Audio – Audio files
* Maps – Mazes and scoreboards
* Prefabs – Dots
* Scenes – Main Menu, Game, and Results screens
* Scripts – Source Code for all items except AI movement
* spritesAndGameObjects – Images of game objects (not including maps and scoreboard)
* TextMesh Pro – Resources for Text

The files in the Packages directory are the external libraries that were utilized in the project

**HOW MEMBERS CONTRIBUTE**

New members of our team or other developers can contribute by downloading Unity 2018.2.18f1 and cloning the Git Hub repository that Abraham is the keeper of. A new user would need to request to be a contributor. Any new contributor should know the following key items:

* Source Code is mainly located in the Scripts directory
* Source Code for the AI is in the AstarPathFinding Project directory
* If they want to change any of the UI, most of those items are in the scpritesAndGameObjects directory