|  |
| --- |
| **Feature Name:** Player Move |
| **Feature Description:**  Player can move left right up and down, only send data to server when input has changed to minimise data  **Conditions of Satisfaction:**   1. Player Moves when press left and right 2. Player moves when press up and down 3. Player cant go through walls |
| Feature Name: Player Eats Pebble |
| Feature Description:  Player can collect pebbles big and small, when a player collects 10 small pebbles increase their speed  Conditions of Satisfaction:   1. Player collides with pebbles 2. Pebbles disappear when touched 3. Collsion increments a pebble counter for player 4. Big Pebbles respawn after a set amount of time |

|  |
| --- |
| **Feature Name:** Player Connects to Server |
| **Feature Description:**  Player can join server at anytime from the game when the player joins the game is notified of the number of players  **Conditions of Satisfaction:**   1. Player can be host or join a game 2. Game Starts when two players on the screen |

|  |
| --- |
| **Feature Name:** Player Win Lose |
| **Feature Description:**  Player can win by eating the other player. When player wins a win message displays similar for a loss too  **Conditions of Satisfaction:**   1. Show win message on end 2. Show loss message on end 3. Restart the game |

|  |
| --- |
| **Feature Name:** Player Power Up |
| **Feature Description:**  Player gains ability to eat other player when they eat a large pebble. Sprites should change from pac man to ghost.  **Conditions of Satisfaction:**   1. Change opposing player to ghost 2. Player can eat other ghost |

|  |
| --- |
| **Feature Name:** Add Real Time + predictive movement |
| **Feature Description:**  Players are playing in real time so should both be as close as possible to playing on the same computer. Latency must be accounted for so we must predict player movement/position at specific intervals of synchronisation.  **Conditions of Satisfaction:**   1. If sync data is slightly behind player movement on an axis don’t update with a synchronisation on the slow axis, instead fix to axis that isn’t changing. 2. Must look seamless between two computers 3. If movement is behind where it should be push the synchronisation data to align it with the other pc(server/host). |