Objectives:

* Flask server with list of users and their corresponding account information.
* World manager that generates a world full of pixels
* QT menu with user login or user selection
* PyGame game shell that displays the pixels and player(s).
* Ability for players to ‘mine’ pixels.
* Controller support

Stretch Goals:

* Place mined pixels
* Craft new pixels from mined pixels
* World expands as you reach the edge, player is center of focus.

|  |
| --- |
| **Server** |
| - users : List<Player>  - world : World Manager |
| + login() : void  + worldState() : World Manager  + playerStates() : List<Player>  + update() : void  + controlInput() : void |

|  |
| --- |
| **World Manager** |
| - pixels : List<Pixel>[][] |
| + generate(w : double, h : double) : void  + force(x : double, y : double, dir : double) : void  + getView(x : int, y : int) : List<Pixel> |

|  |
| --- |
| **Pixel** |
| - isMoving : bool |
| + ForceApplied(x : double, y : double) : void  + tick() : void  + staticFriction : double  + kineticFriction : double  + colour : colour  + forces : double[2]  + velocity : double[2]  + mass : double  + position : double[2]  + motionBlur : pixelChainNode |

|  |
| --- |
| **Login Window** |
| - isConnected : bool |
| + connect() : bool  + verifyConnection() : void  + serverID : string  + userID : string  + password : string |

|  |
| --- |
| **Player** |
| - isMining : bool  - timeUntilLogout : int |
| + save() : string  + tick() : void  + reverify() : void  + move(x : double, y : double) : void  + setCursorPosition(xMod : double, yMod : double) : void  + breakPixels(enabled : bool) : void  + position : double[2]  + cursorPosition : double[2]  + userID : string  + userPW : string |

|  |
| --- |
| **Player Client** |
| + tick() : void  + getWorldState() : List<x : double, y : double, c : colour>  + getPlayerLocations() : List<x : double, y : double, d : metadata> |