Design Overview for <<Fantasy Fighter >>
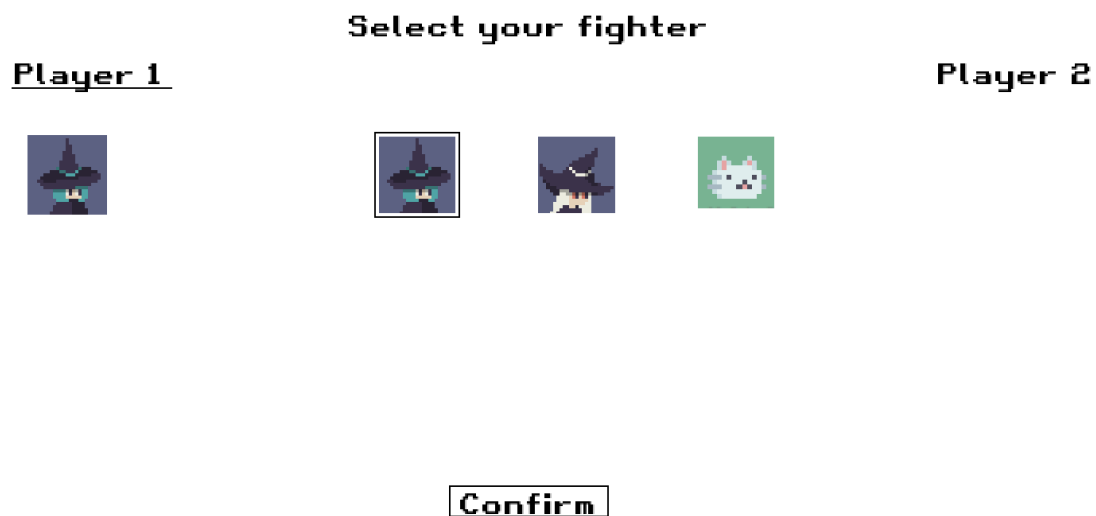
Name: Nadelin Nop
Student ID: 103487208

Summary of Program

This is a local player versus player game, where each player will have a variety of playable characters to pick from.

Each player will get their own turn to pick their fighter. After that their selected characters will spawn in a new place to combat. Player 1 will be using the WASD keys to move and attack, whereas player 2 will be using the arrow keys. The players will then have to battle to the death and the remaining player will win.
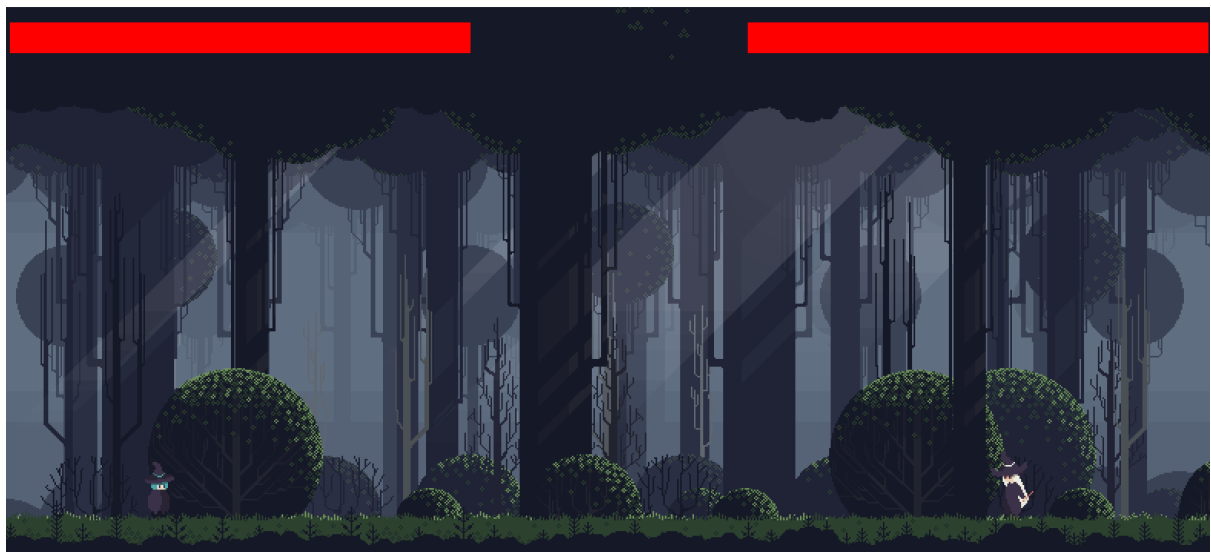
Required Roles

Describe each of the classes, interfaces, and any enumerations you will create. Use a different table to describe each role you will have, using the following table templates.

Table 1: <<Character>> details – duplicate

Responsibility	Type Details	Notes
Blueprint for making all the playable fighters	Bitmaps, Animation Script, Animation, Draw, Attacking and Taking Damage	This class will handle everything related to creating the fighters

Table 2: <<CharacterFactory>> details – duplicate

Responsibility	Type Details	Notes
Responsible for choosing which fighter class to instantiate based on conditions	Returning fighter objects	This class will essentially be the factory at creating characters

Table 3: <<GUI>> details – duplicate

Responsibility	Type Details	Notes
Blueprint for all the GUIs that will be required for this game	Handles all the drawings for the GUI	

Table 4: <<Input>> details – duplicate

Responsibility	Type Details	Notes
Handles all the user's input	Returning Point2D locations	

Table 5: <<UI>> details – duplicate

Responsibility	Type Details	Notes
Decides what to do with the user's inputs	Checks the inputs and does different actions based on it	It acts like a manager for the incoming inputs from users

Table 6: <<State>> details – duplicate

Responsibility	Type Details	Notes

Handles the different behaviours of the program at different sections of the game	Returning states	An implementation of simple pattern design

Table 6: <<Player>> details – duplicate

Responsibility	Type Details	Notes
Handles the different aspects of player 1 and player	Move methods, Draw methods, CheckingInputs from the UI, and Character Selection	

Table 2: <<StateBase >> details

Value	Notes
Select Move End	For dealing with different states of the game

Class Diagram

Sequence Diagram

This sequence diagram shows a snippet on how the gui interaction can lead to the creation of characters depending also on whether player 1 is selecting or player 2.