# *Pygame Assignment Planning Document Template*

*You can use this as a guide to developing your template for your game plan/design. Remove any instructional/example text if you choose to use this document as a basis for your own. Alternatively you can create your own document from scratch, but it must contain the information specified on this document as a minimum requirement.*

*All instructional text is in red italics – you should not include any of the red text in your final submission. Example text should be replaced as appropriate.*

Title: The Legend of N++

Game Style: A Side-Scrolling, Platformer, Hack-and-Slash game

Developer: Caleb Killalea

# Overview

A hack-and-slash game where you must eliminate all the enemies in the game area and then use the platformer elements to make your way to the top of the map and proceed to the next area, different areas have different requirements to progress to the next stage.

## Win Condition:

The player wins by making their way through the levels while killing all the enemies in each area, once the enemies have been eliminated, find the required items to unlock the exit and make your way to the top.

## Lose Condition:

The player loses all their health by coming into contact with the enemies.

# Game Screens

*Note: Only new events need to be included on each screen – if something has already been observed previously, there is no need to include those details again.*

## Screen 1 – Start Screen and Menu

|  |  |
| --- | --- |
| https://i.gyazo.com/d9e597fc2ca7f62b9245d501eddeb7b1.png | Events, Transitions and Callbacks:   * click on Start   + Go to Difficulty selection * click on controls   + go to controls screen, blits the controls to the screen * click on Quit, Esc   + Exit the game   Objects and Variables:   * Uses a function to blit the buttons to the screen, the button function uses the variables (x, y, width, height, focusorder, selection(this is the screen it is on)   Notes:   * I made the screen and the art by hand, I also used custom locations and functions for the buttons and highlights. |

## Screen 2 – High Score Table

|  |  |
| --- | --- |
|  | Events, Transitions and Callbacks:   * M, click on Menu   + Go back to Menu Screen   Objects and Variables:   * High Score list – stores all previous high scores, loaded from text file.   Notes:   * Consider use of different colours for each high score – gold, silver, bronze then white. |

## Screen 3 – Difficulty Screen

|  |  |
| --- | --- |
| https://i.gyazo.com/6e82a4c27b1ea3b6f3a733ec2bb032d0.png | Events, Transitions and Callbacks:   * click on Easy   + Sets “diff” variable to 1, this sets the amount of enemies to 2 per level * click on Medium   + Sets “diff” variable to 2, this spawns 4 enemies per level * click on Hard   + Sets “diff” variable to 3, this sets the amount of enemies per level to 7.   Objects and Variables:   * Uses a function to blit the buttons to the screen, the button function uses the variables (x, y, width, height, focusorder, selection(this is the screen it is on)   Notes:   * I made the screen and the art by hand, I also used custom locations and functions for the buttons and highlights. |

## Screen 4 – Medium Level Example

|  |  |
| --- | --- |
|  | Events, Transitions and Callbacks:   * Collision with Wall of Death   + Player loses a life   Objects and Variables:   * Walls – list of coordinates required to deal with complex shapes; property to determine if Wall of Death vs. regular wall.   Notes:   * Death due to collision with Wall of Death does NOT reset time, only player’s position to start of level. |

## Screen 5 – Hard Level Example

|  |  |
| --- | --- |
|  | Events, Transitions and Callbacks:   * Collision between Enemy and Player   + Same behavior as collision with Wall of Death * Collision between Enemy and Wall   + Collision is ignored – enemy passes through wall.   Objects and Variables:   * Enemy Objects – List of tuples to determine path of movement and speed;   Notes:   * Enemies remain on path after collision and continue without change |

## Screen 6 – Win/Lose Screen

|  |  |
| --- | --- |
| << image goes here >> | Events, Transitions and Callbacks:   * Y, click on Yes   + Play Again – back to Level 1 with Score of 0 * N, click on No, Esc   + Exit the game   Objects and Variables:   * Win State – used to determine if the player won or lost, and to display an appropriate message. * High Score Flag – if player’s score was good enough, allow them to enter their name and save to the high score text file. |

# Supporting Information

## Development Environment

The game will be developed in Python using Pygame. All game assets (such as sprites and obstacles) will be created by the designer – no attribution will be required for assets owned by other people.

## Key Features

### Enemy AI

* Enemies will move on fixed paths – they will not divert from these paths in response to any player initiated event.
* As the levels increase in difficulty, enemies may move faster and/or slower to make it harder for the player to progress.

### *Something Else*

* *Add as many of these features as you might need to describe additional features that may not be captured in the screen examples above and will need to be considered in your implementation.*