**Assignment 02**

**Requirement Document**

Braeden Timlin

101005580

IMD 3006A

Revision History

|  |  |  |
| --- | --- | --- |
| Version | Description | Approval |
| 1 | First Draft |  |
| 2 | Revision 1 |  |

Scope of Document

This document describes the objectives and requirements of the visual based game I created.

Description of the Program

The game allows the player to move using the arrow keys. They must pick up all the power ups on the screen without getting hit by the enemies that are moving randomly on screen. If the user gets hit by an enemy they lose, but if they collect all of the power ups then they win.

Objectives of the Project

The objectives of this project is to demonstrate my abilities to use Object Oriented programming methodologies to create a simple graphics based game. The game will consist of a user controlled player object, 10 enemies that move in random directions, and 15 collectible items that the user needs to pick up to win.

Functional Requirements

1. User Input:
   * The user will use the keyboard arrow keys to control the player entity in the game
2. Game Entities
   * Player
     + Will be use a single sprite of Link from the original Legend of Zelda. Typical users would associate this character with the player character.
     + User keyboard input will move the character around the screen.
   * Enemies
     + Will use a single sprite of an Octorock from the original Legend of Zelda. This image is associated with enemies typically.
     + Getting hit by an enemy will cause the user to lose the game.
     + These entities will move in random directions, with a cooldown of 3 seconds before they can change their movement to another random direction.
   * Pick-Ups
     + The pick-ups will all use the sprite of a Rupee from the original Legend of Zelda. This is associated with currency and points to most users.
     + Collecting all of these will win the game.
3. Data Structures (classes & functions)
   * **Entity** class
     + Integer iX variable
       - X Position
     + Integer iY variable
       - Y Position
     + Image Sprite variable
       - The image that the game will render for the entity
   * Character class, inherits from Entity class
     + Integer iSpeed variable
       - The amount of pixels that the entity will translate each frame
     + Integer direction variable
       - 0-3 corresponds to the directions up, left, down, and right respectively
         * 0: up
         * 1: left
         * 2: down
         * 3: right
     + Boolean bHit
       - Dictates whether the entity is hit or not
   * **PickUp** class, inherits from Entity class
     + Boolean bDrawMe
       - Dictates whether the entity should be drawn and checked for collision
   * **Physics** class
     + Collision function
       - Checks the distance between 2 entities and returns true or false based on whether the distance between them is smaller or larger than 25 pixels
     + Move function
       - Increments or decrements the given entity’s x and y values based on their direction and speed
       - Checks X and Y position to ensure objects do not move out of the viewable frame
   * **Game** class
     + Integer iPts variable
       - The number of points that the user has, each pick up collected should increment this value
     + Setup function
       - Creates and initializes all of the entity objects in the game
         * Player
         * Array of 10 enemies
         * Array of 15 pick-ups
       - Creates Physics object
     + Update function
       - Calls the physics collision between the player and all enemies, and between the player and all pick-ups using for loops
       - Increments iPts if player collision with pic-ups returns True
       - Make player bHit variable to True if player collision with enemy is true
       - Calls the endCheck function
     + Render function
       - Updates drawing x and y value so each entity’s sprite is centered on their x and y coordinates
       - Draws the entities at their respective x and y values
     + endCheck function
       - Checks if the player bHit variable is true
         * If true, then the game ends with a “You Lose” message

Function returns True

* + - * + If false, then the game continues
      * Checks if the iPts value is equal to the total number of pick-ups
        + If true, the game ends with a “You Win” message

Function returns True

* + - * + If false, the game continues
  + **Main** Function
    - Creates Game object
    - Calls Game setup function
    - Boolean bGame variable
      * Boolean for whether game is running
    - While loop when bGame is equal to True
      * Calls game update function
      * Calls game Render function
      * Calls game endCheck function

1. Multimedia Components
   * The program will support .PNG type images
   * There will be no audio