



GAM405.01A – Scripting for Game Development B
Assignment: Project Brief and Project User Manual

Student Name: James Byron

Student Number: 1005425

Project Name: Robotic A-Mazement

High Concept:





“Robotic A-Mazement” is a simple, physics-based 2D Platforming game. The player assumes the role of a testing robot named G.O.O.D (Geometric Object Oriented Device) who has to navigate a maze that has been littered with junk, while having to tidy up for the next test that is due to begin in three minutes. G.O.O.D is able to pick up junk and toss objects around to ensure that the maze’s trials are working perfectly, collect batteries to enhance his movement and other stats, and avoid the malfunctioning B.A.D.D.I. Model Robots.

Aim of the game:

The player’s goal is to reach the end of the level within a 3 minute time limit. They can achieve this goal by collecting batteries to boost their mobility, and by defeating enemies by throwing objects at them.

How to Play and Rules

Play takes place in a 2D side-scrolling environment, with a defined starting point and finish line. At the top of the screen is a timer that starts at 3 minutes. The player has to make it to the end of the stage within this 3 minute timeframe or else they hit a game over. If the player makes it to the end of the stage, they win the game and are offered the chance to start again.

The player moves with the Keyboard keys (WASD) or Arrow keys () for movement and jumps with the spacebar key. As the player moves they will be able to collect batteries that power them up, which influence aspects of the player's controls. Colliding with these batteries will automatically level up the player.

- Red Batteries will level up the player's throwing distance
- Blue Batteries will level up the player's movement speed
- Green Batteries will increase the player's health
- Yellow Batteries will level up the player's jump height.

Throughout the maze there are three main non-player entities, grabbable objects that the player can pick up (For example, a cube or ball), a drone that will throw the player upon colliding them, and robots that will attempt to hurt the player by firing a laser at them. The player has to avoid or defeat these threats by throwing objects at them to stop them, otherwise the player will lose their own precious time that they need to complete the maze. Finally is the stopwatch item. If the player touches a stopwatch, the timer will go up by 15 seconds, allowing the player to have more time to complete the level.

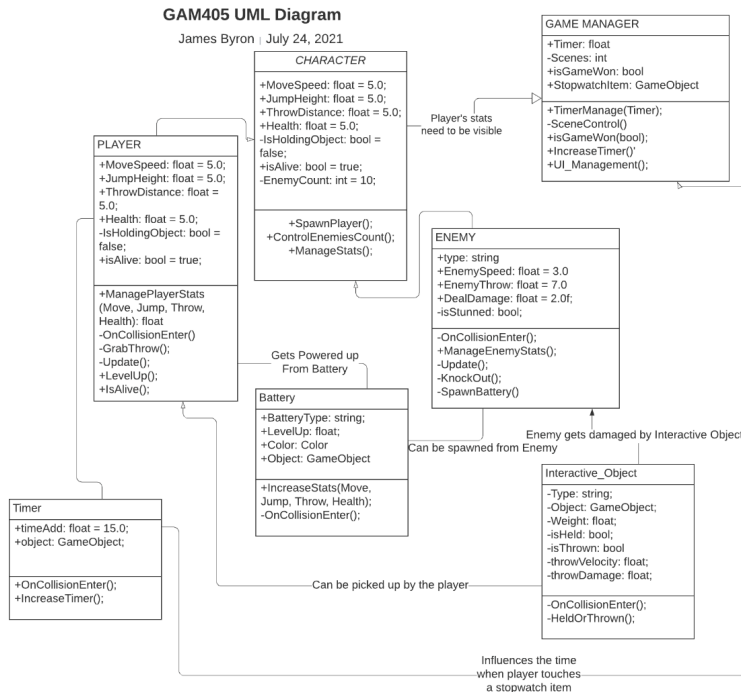
Mechanics:

The game will have two main mechanics, grabbing and throwing objects, and the ability to perform an eight-way dash when jumping. The player will pick up interactable objects by pressing the R key. When the player has picked up an object, the player can still move and hold the item. Pressing the T key will make the player throw the object in one of eight directions, depending on the direction the player is moving.

By pressing the Q key, the player will be Launched, making them fly in a given direction based on the player's movement. This can be used as a double jump when in the air or to help the player's mobility to speed through the maze.

The player will automatically collect beneficial items by colliding with them. This includes batteries that influence the player's mobility stats, such as increasing movement speed or jump height. Colliding with a stopwatch item will increase the amount of time needed to complete the level by 15 seconds.

UML Planning Documentation:



Basic flow of game/Pseudocode

Game begins on Main Menu Screen. Player chooses between Start, Controls or Close Game options. Pressing Start begins the game, Controls to check how to play the game.

In Game:

Play begins with the player spawning on the stage. The Game Manager oversees the Timer, starting at 3:00 and decreasing in real time. The player has to reach the goal within 3 minutes or the game is over. Alternatively, the game is over if the player's health reaches 0.

Player controls:

The player moves with the WASD keys or Arrow Keys. Space to Jump. R to Pick up objects, T to throw, Q to launch in a moving direction.

Player Collisions:

Check for when the player collides with entities and objects.

```
if (Collision battery) {
    CheckBattery
    Case: Battery = Red:
        Increase Throwing Distance
    Case: Battery = Blue:
        Increase Movement Speed
    Case: Battery = Yellow:
        Increase Jump Height;
    Case: Battery = Green:
        Increase Player Health;
}
if (Collision timer) {
    Tells GameManager to add 15 seconds to the timer
}
if (Collision enemy) {
    Case: Collides with Blue enemy:
        Player gets thrown away into air.
    Case: Collides with Red Enemy:
        Player loses health and time
}
if (Collision Goal) {
    Player has won the game and goes to victory screen
}
if (Enemy collides with thrown object) {
    Set enemy to stunned, and unable to move for 5 seconds
    Drop a power-up like a battery
}
For interactive objects:
if (heldByPlayer == true) {
    ThrowingDamage = PlayerMove + this.velocity;
}
```

Pseudocode and Systems:

>Begin with the Main Menu screen, which provides the options to start the game, check the controls, or exit the game.

>Pressing start will take the player to the game scene, and the timer becomes active, starting from 3:00 minutes. The player has to make it to the goal within this 3-minute window, or else it's game over.

>Player moves left and right with the WASD or Arrow Keys, and jumps with the Spacebar key. Player can pick up interactive objects with the R key, and throw them with the T key. The player can launch themselves in a moving direction with the Q key.

>The player can collide with objects and entities, including batteries, time-limit increasers (Will add 15 more seconds to the timer). There are four types of batteries with positive effects:

>Colliding with an enemy will have adverse effects, with blue enemies launching the player away into the air, while red enemies will stun the player, which detracts both the player's health and time. If either the time or player health reach 0, it's game over.

>Enemies are defeated by throwing an object at them. Defeating enemies will drop batteries or stopwatches to increase stats or time respectively.

>When the player collides with the end goal, then the game is won and the player is taken to a victory screen.

GameManager

- Controls the scenes of the game, and how much time is left
- Controls Audio
- Timer objects increase the remaining time needed to complete the stage.

Time:

- Begin at 3:00 Minutes. Decreases in real time.
- If (Time runs out)
 - Bring up game over screen.
- If (Player collides with stopwatch item)
 - Increase the timer by 15 seconds.

Player

- Start with Default Stats: (Movement = 5, Jump = 5, Throw = 5, Health = 5)
- OnCollisionEnter() Function detects player's collision with entities.
- If (player collides with Battery)
 - Red: Increase player's throwing distance
 - Blue: Increase player's movement speed
 - Yellow: Increase player's jump height
 - Green: Increase player's health
- If (Player collides with RED Enemy)
 - Decrease Health and Time
- If (Player's Health = 0)
 - Bring up Game Over screen
- If (Player collides with Goal)
 - The player has won the game
- Player Moves in response to player keyboard input. (WASD/Arrows for Movement, Space = jump, Q = Launch, R = Grab, T = Throw)
- Has a state for when they are holding an object.
- Sounds play when collecting items and power ups.

Battery

- Upon colliding with player, level up their stats
- Spawn in four types to represent these stats.

- Red = Throwing, Blue = Movement, Yellow = Jumping Height, Green = Health
- Increases the value of the player's stats by 1 for each battery type collected.

Enemies:

- Two types of B.A.D.D.I Enemies, Red and Blue (A string value determines enemy type).
- If (Enemy collides with a thrown object)
 - Disable the enemy
- If (Player collides with Blue B.A.D.D.I)
 - Enemy will throw the player away into the air.
- If (Player collides with Red B.A.D.D.I)
 - Attack the player and stun them, depleting health and time.
 - Sounds play upon colliding with enemies.

Interactive Objects:

- If (Being held by player)
 - ThrowDamage = Player's Movespeed + this.ObjectVelocity
- If (Collides with Enemy)
 - Disables the enemy
- Track whether the item is being held by the player or not.

Scenes and Screens:

- Main Menu:
 - Start: Takes the player to the main game scene to begin play.
 - Controls/How to Play: Gives the player the controls of the game
 - Quit: Exits the game
- Game Scene:
 - 2D Platformer level. Player spawns here and has three minutes to reach the goal post at the end of the level.
- Game Over:
 - Screen displayed if the player loses all health or time runs out
 - Pressing the retry resets the timer and stats to 0, resets everything back to normal and play resumes from this point.
 - Pressing the exit key takes the player back to the main menu.
- Game Won:
 - Screen displays when the player has successfully completed the level
 - Same functions as the game over screen - Retry resets the timer and stats to 0, pressing Exit takes the player to the main menu.