**Software Now**

**HIT 137**

A close up of a logo

Description automatically generated**Assignment 3 Question 1**

**Group CAS076**

Gilbertofer Tanoto (S379097)

Phat Nguyen (S365522)

Umais Ahamed Anwar Ali (S373981)

Syed Sayed Hasan (S377467)

**Casuarina Campus Sem2 2024**

**Title: Side-Scrolling Game Development Report**

**1. Introduction**

The purpose of this project was to create a **side-scrolling game** using Python and the Pygame library. The game allows players to control a character, navigate through levels, defeat enemies, and ultimately face a boss. Throughout the game, players must maintain their health and lives while attempting to advance to higher levels. This report outlines the core components of the game, the mechanics implemented, and key features such as player movement, enemy behavior, and game progression.

**2. Game Design and Mechanics**

**2.1 Game Structure**

The game is structured into **three levels**, with increasing difficulty as the player progresses. Levels 1 and 2 consist of regular enemies, while Level 3 introduces a **boss** that requires multiple hits to defeat. The player can lose health upon contact with enemies and will lose a life if their health depletes. If all lives are lost, the game ends. Successfully defeating all enemies, including the boss, results in a win.

**2.2 Player Character**

The **Player class** defines the player's character, including attributes such as:

* **Movement**: The player can move left or right using the arrow keys and can jump using the spacebar. The jumping mechanics involve a gravity effect that simulates realistic jumps.
* **Health and Lives**: The player starts with 100 health points and 3 lives. Contact with an enemy reduces health, and losing all health causes the player to lose a life. The health resets after each life loss.
* **Projectiles**: Players can shoot projectiles by pressing the ‘Z’ key, allowing them to attack enemies from a distance.

**A screen shot of a computer code

Description automatically generated**

**2.3 Enemy Mechanics**

The **Enemy class** represents the regular enemies, and there is a specialized **Boss class** that inherits from the Enemy class. Enemies have attributes such as:

* **Movement**: Enemies move from right to left across the screen, and their speed increases as the levels progress.
* **Health and Hits**: Regular enemies can be defeated with one hit, while the boss requires multiple hits. The health of the boss is higher, reflecting its status as the final challenge.

**A screen shot of a computer program

Description automatically generated**

**3. Game Progression and Level Design**

**3.1 Levels and Enemy Spawning**

The game starts with **Level 1**, where 5 enemies spawn. As the player advances to **Level 2**, the number of enemies increases, making it more challenging. By **Level 3**, there are regular enemies as well as the **Boss**. This progression ensures the game remains engaging and offers increased difficulty.

The function spawn\_level() is responsible for setting up each level, ensuring that the correct number of enemies, including the boss, is spawned.

**A screen shot of a computer program

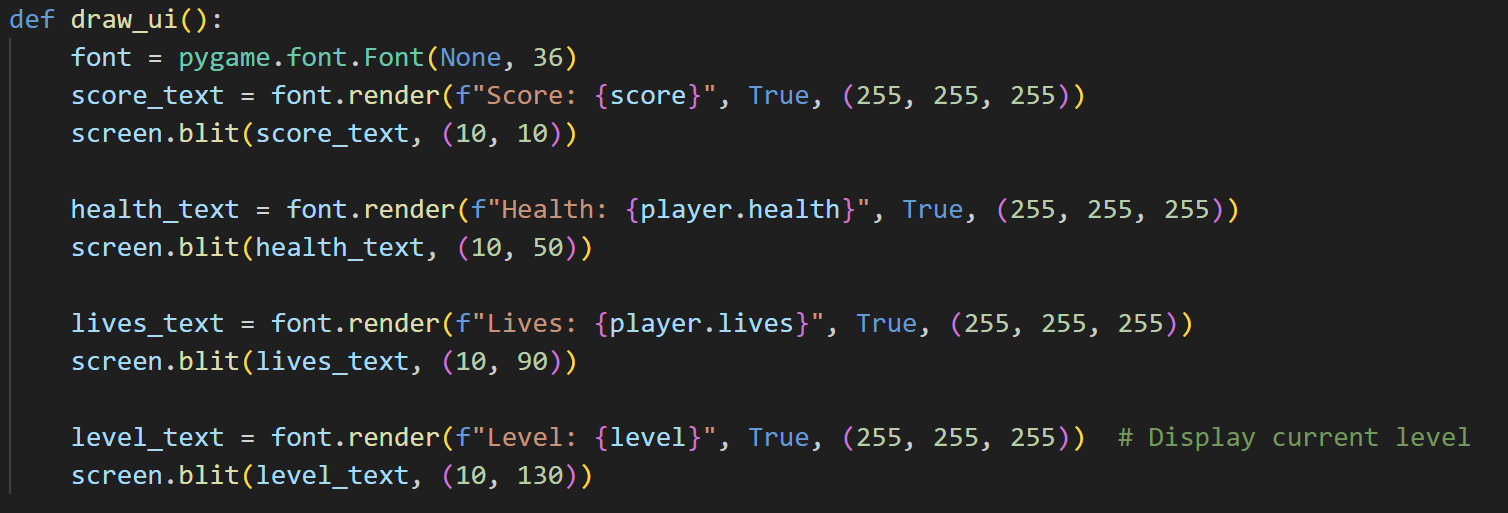
Description automatically generated**

**3.2 Scoring System and UI Elements**

The game features a **scoring system** where the player earns points for defeating enemies. Additionally, the user interface displays essential information, such as:

* **Score**
* **Health**
* **Lives**
* **Current Level**

This information helps players track their progress and understand their status within the game.



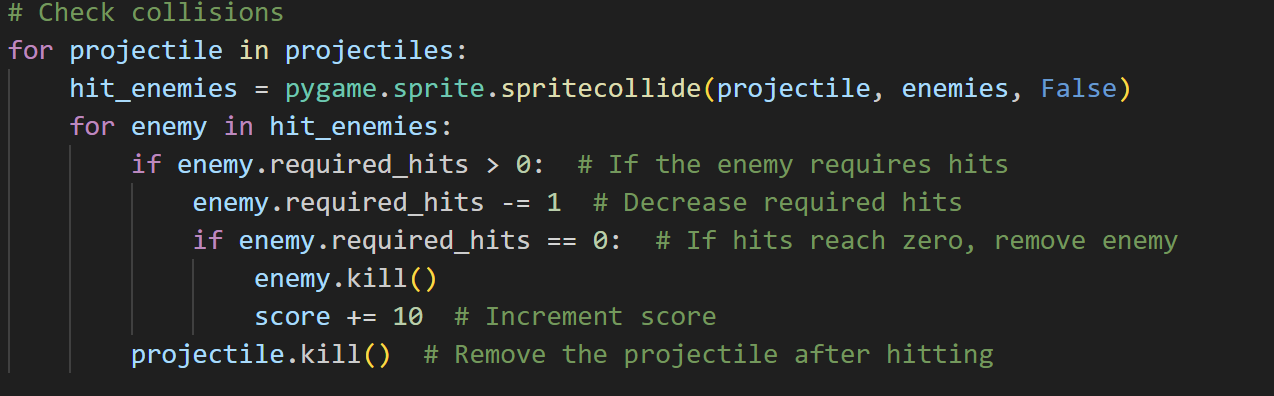
**4. Key Features and Visuals**

**4.1 Player Shooting Mechanism**

The player can shoot projectiles to defeat enemies. This adds a layer of complexity, as players need to time their shots while avoiding collisions. The Projectile class manages the shooting mechanics, ensuring projectiles move across the screen and disappear once they exit the screen or hit an enemy.

**4.2 Enemy Collision Detection**

Collision detection is used to manage interactions between **projectiles and enemies** as well as **player and enemy** collisions. If a projectile hits an enemy, the enemy's health is reduced, and if the health reaches zero, the enemy is defeated. Player collisions with enemies reduce the player’s health, increasing the challenge.

**5. Challenges and Solutions**

**5.1 Camera and Background Scrolling**

To create a **side-scrolling effect**, the game uses a camera system that follows the player. The update\_camera() function adjusts the camera position based on the player’s movement, ensuring the background moves smoothly, providing a consistent scrolling effect.

**A black screen with white text

Description automatically generated**

**5.2 Difficulty Balancing**

One of the key challenges was **balancing difficulty** across levels. Initially, enemies moved at a fixed speed, which made the game less dynamic. To solve this, enemy speed now increases with the level, and the boss requires multiple hits, adding to the game’s challenge.

**6. Win and Game Over Conditions**

The game features clear **win** and **game over** conditions:

* Players win by defeating all enemies, including the boss, and a congratulatory message is displayed.
* Players lose if they run out of lives, triggering a game-over screen.

The game allows players to restart from the beginning by resetting all variables, providing replayability.

**A black screen with text

Description automatically generated7. Conclusion**

This Pygame project successfully created an engaging and challenging side-scrolling game that features a **well-designed player character, dynamic enemies, and progressive levels**. The use of collision mechanics, projectile shooting, and a scoring system added depth to the gameplay, while visual elements like scrolling backgrounds and user interfaces enhanced the player experience. By integrating these mechanics, the game offers an enjoyable experience that requires players to strategize, react quickly, and navigate through increasing difficulty levels. Future improvements could include adding more enemy types, power-ups, and expanded levels to enrich gameplay further.