

# One Piece Game

## 1. Project Description & Scope Specification

This game is called One Piece. It is a simple adventure game made with UWP (Universal Windows Platform). The player controls Luffy, a character from the anime One Piece. The goal is to collect crew members like Sanji, Zoro, Nami, and Robin while avoiding obstacles like rocks and waves.

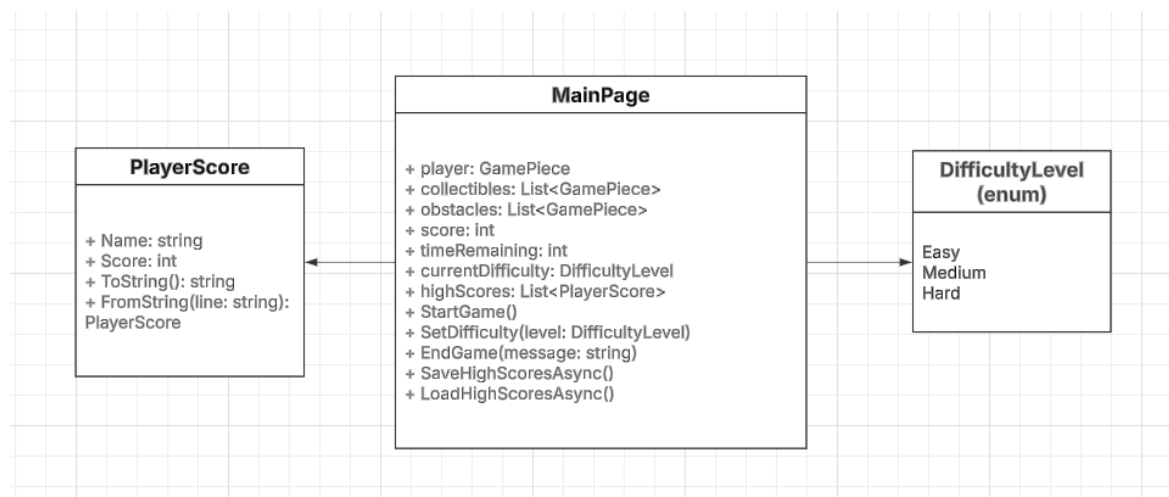
The game has three difficulty levels: Easy, Medium, and Hard. Each level changes the number of collectibles, obstacles, and the time limit. The game also includes a score system and saves the top 5 high scores in a file. If the player gets a new high score, they can enter their name.

## 2. Class Descriptions and Diagrams

**MainPage:** This is the main class of the game. It controls the game logic, player movement, score, timers, and game events.

**PlayerScore:** This class saves the player's name and score. It helps to save and read scores from a file.

**DifficultyLevel:** This is an enum with three levels: Easy, Medium, and Hard. Each level changes the game settings.



### 3. Agile Project Specifications

**Epic:**

*As a player, I want to play a fun and challenging One Piece-themed game so that I can enjoy collecting characters and improving my score.*

**User Stories:**

1. *As a player, I want the game to automatically adjust the difficulty level as I progress, so that the challenge increases gradually.*  
Acceptance Criteria:
  - The game starts at Easy difficulty.
  - After completing a level, the game increases to Medium, then Hard.
  - Each difficulty level changes the number of collectibles, obstacles, time limit, and score target.
2. *As a player, I want to see a timer so that I know how much time I have left.*  
Acceptance Criteria:
  - A countdown timer is visible during the game.
  - The game ends when the timer reaches zero.
3. *As a player, I want to save my high score with my name so that I can see my achievements.*  
Acceptance Criteria:
  - If I get a high score, I am asked to enter my name.
  - My name and score are saved in a file.
  - The top 5 scores are shown after the game ends.
4. *As a player, I want to choose to play again or quit after the game ends.*  
Acceptance Criteria:
  - A dialog appears with "Play Again" and "Quit" options.
  - Choosing "Play Again" restarts the game.
  - Choosing "Quit" closes the app.

### 4. User Manual / Guide

How to Play:

- Use the arrow keys to move Luffy.
- Collect crew members to get points.
- Avoid obstacles. They take away points.
- Finish the level by collecting all crew or reaching the target score.
- If your score is high enough, you will go to the next level.
- If your score is 0 or time ends, the game is over.
- If you get a high score, you can enter your name.
- After the game ends, choose to play again or quit.

## 5. Change Log (from Lab 1A)

- Difficulty levels (Easy, Medium, Hard) were added.
- A timer was added to limit game time, and the method from Lab 1A was improved.
- A high score system was created using file saving.
- A new class called PlayerScore was created to store player names and scores.
- A dialog box was added to enter the player's name when a high score is achieved.
- Options to play again or quit were added at the end of the game.
- The PlayerScore class was created to manage scores.
- The code was organized using #region tags to make it easier to read and update.
- The collision detection system was improved.
- Collectibles were made random, with different values and positions depending on the difficulty level.
- The code was updated to play background music from the MainPage class.

## 6. Bug Report

**Bug:** HighScores.txt saved only zeros.

**Cause:** Wrong format used when saving scores.

**Fix:** Used PlayerScore.ToString() to save name and score correctly.

**Bug:** Name input dialog did not appear.

**Cause:** Dialog only showed for new high scores.

**Fix:** Checked score logic and added option to always ask for name.