This text documents the UML diagram for our project, a 2-D endless runner game made with Unity, that has (but is not limited to) the following classes and relationships:

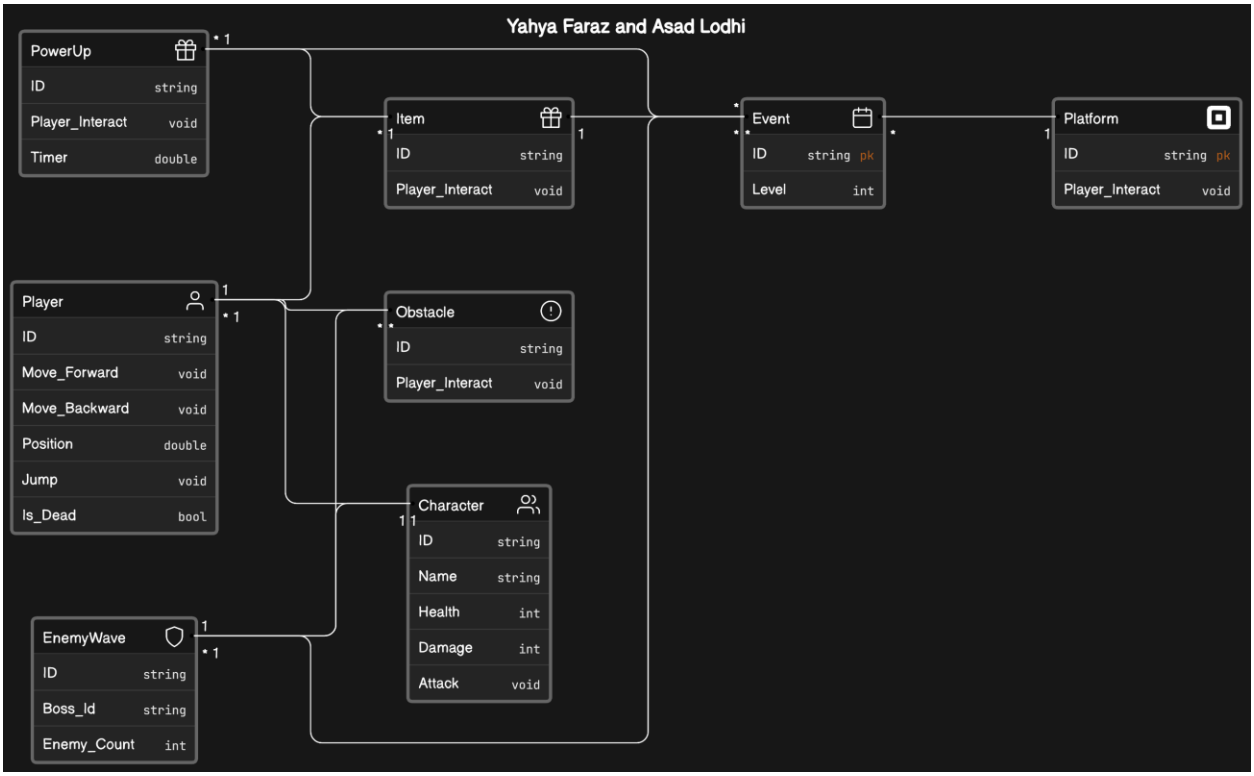
- Player: This class represents the player, our user. It has methods for moving forward and backward, jumping, and checking if our character is dead.
- Character: This class is a base class for all characters in the game, including the player character, enemies, and bosses. It has attributes for name, health, and damage. It also has a method for attacking and movement.
- Obstacle: This class represents obstacles that the player character can collide with. It has a method for checking if it has collided with a character.
- EnemyWave: This class represents a wave of enemies. It has attributes for the boss and a list of enemies.
- Item: This class represents items that the player character can collect. It has a method for checking if it has been collected by a character, what happens after it has been collected, and how long it will remain.
- Platform: This class represents platforms that the player character can stand on. It has a method for checking if a character is on it.
- Event: This class represents a game level. It has attributes for the level number and lists of platforms and enemy waves.

The arrows between the classes show the relationships between them and by clicking on the boxes, you can view the code snippets of all the classes.

```
**Association:**
Player <-- Item
Player <-- Obstacle
EnemyWave <-- Obstacle
Item <-- Event
PowerUp <-- Event
EnemyWave <-- Event

**Aggregation:**
Event --> Platform
Event --> EnemyWave

**Composition:**
Player --> Character
EnemyWave --> Character

**Inheritance:**
PowerUp --> Item
```

Hope this documentation helps!

UML DIAGRAM



GAMEPLAY SCREENSHOT