Before

Game
- enemy : GameObject
- player : player class
- collectable : GameObject
- platforms : GameObject
- score : int
- player lives : int

+ update() : void
+ draw() : void
+ restart() : void

GameObject
- position : Vector2
- size : Vector2
- sprite : Sprite

+ update() : void
+ position() : void

Player
- movement : vector2
- speed : float
- jump : bool

+ move(): void
+ jump() : void
+ update() : void
+ collision() : void

Enemy
- movement : vector2
- speed : float
- jump : bool

+ move(): void
+ update() : void
+ checkCollision() : void

Platform
- position : vector2
- size : vector2
- movement : vector2

+ checkCollision() : bool

Collectable
- is_collected : bool
- total_amount : int

+ collision() : void

After

Game
- enemy : GameObject
- orbs_collected : int
- total_orbs : int
- platforms : GameObject
- score : int
- player lives : int
- player_direction : vector2
- player_velocity : vector2
- gravity : float
- is_running : bool
- is_dead : bool
- game_over : bool
- has_won : bool

+ update() : void
+ draw() : void
+ restart() : void
+ initialisePlatforms() : void
+ initialiseOrbs() : void
+ initialiseEnemies() : void

GameObject
- position : Vector2
- size : Vector2
- sprite : Sprite

+ update() : void
+ position() : void

Enemy
- movement : vector2
- speed : float
- is_vertical : bool
- bounds : float

+ move(): void
+ update() : void
+ checkCollision() : void

Platform
- position : vector2
- size : vecotr2

+ checkCollision() : bool

Player
- movement : vector2
- speed : float
- jump : bool

+ move(): void
+ jump() : void
+ update() : void
+ collision() : void

Orb
- orb_collected : bool

+ collision() : void
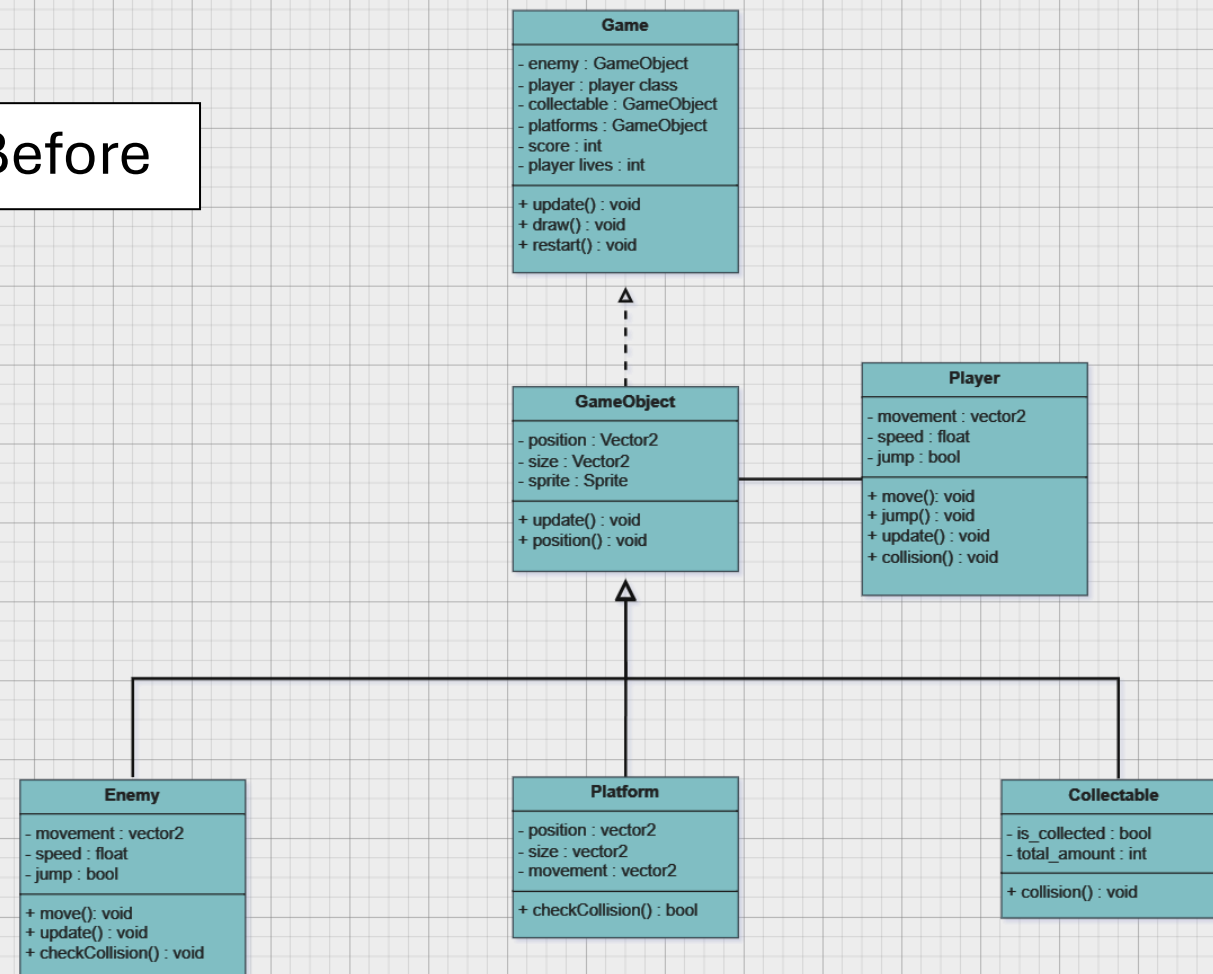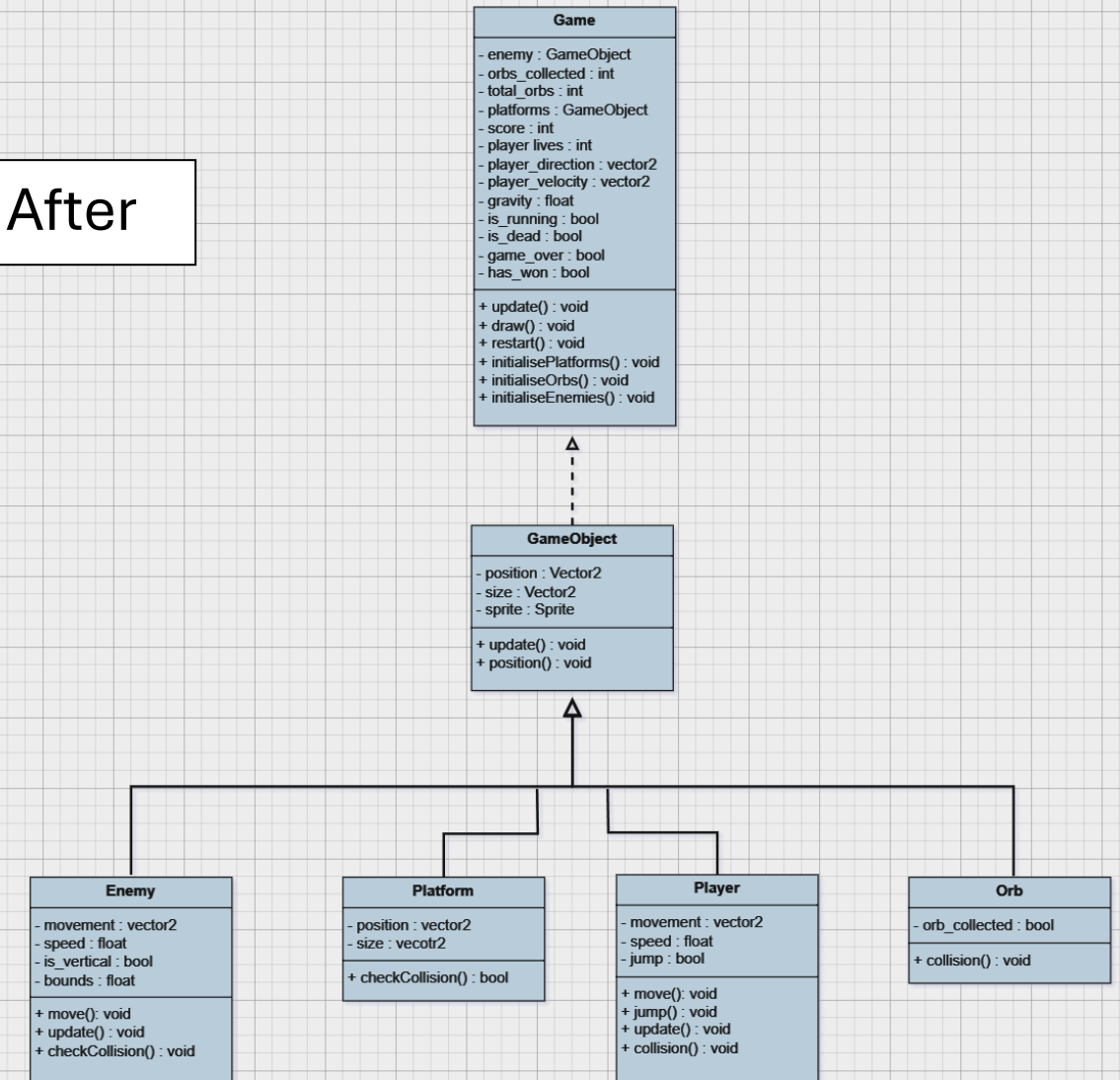
The main difference between the two UML class diagrams is that I have added the player class in line with the other nodes which all inherit from GameObject. I have also added more methods to the game node, including all the initialising. I added more private variables inside the game node to help with clarity.

Pseudocode for character movement
// horizontal movement
if (A key pressed)
    Move character using vector2 struct (velocity.x) -----move_speed
else (D key is pressed)
    Move character using vector2 struct (velocity.x) -----move_speed
// jumping
If (space bar pressed)
    velocity.y ---- jump_strength
    Bool can_jump = true
// gravity
    *Player velocity.y + gravity (float) * dt*
    *Player Move = player.x, player.y * dt*