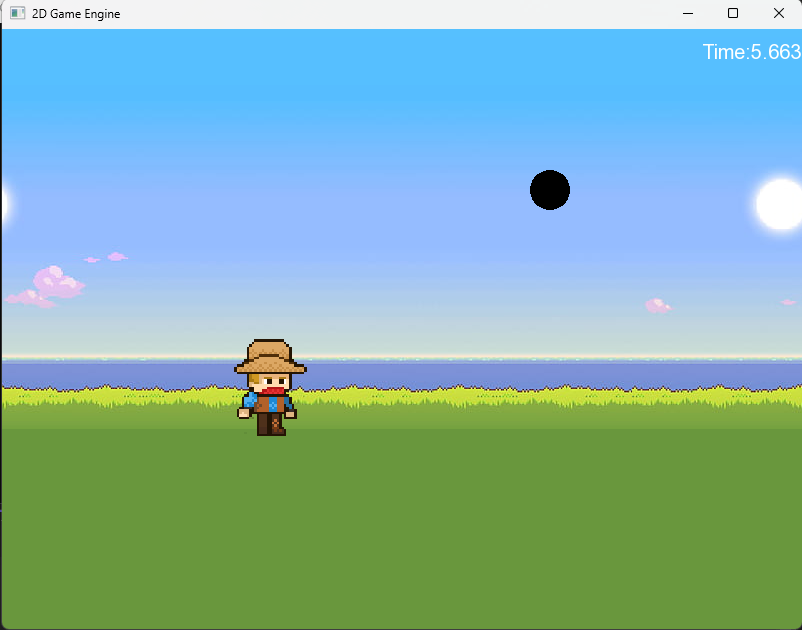
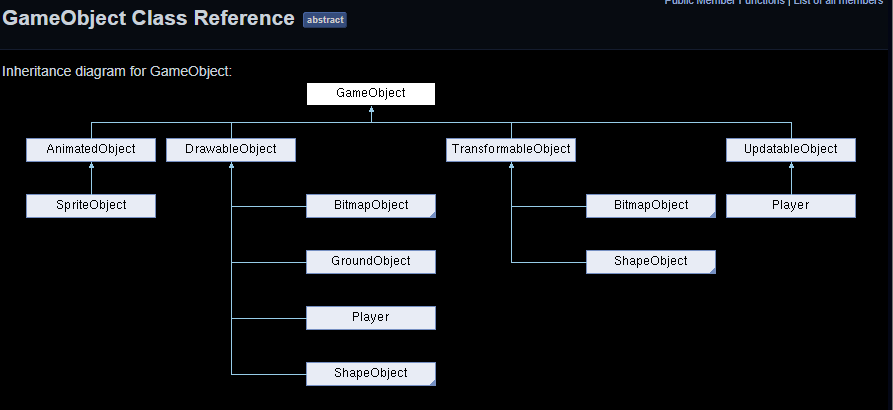
**SFML Game in C++**

**Overview**

This C++ program is designed to create a simple game using the Simple and Fast Multimedia Library (SFML). The game consists of various game objects, each represented by different classes with specific functionalities. The main components include a GameObject hierarchy, specific shape classes, additional object classes, game-specific classes, and collision handling.



**GameObject Hierarchy**

****

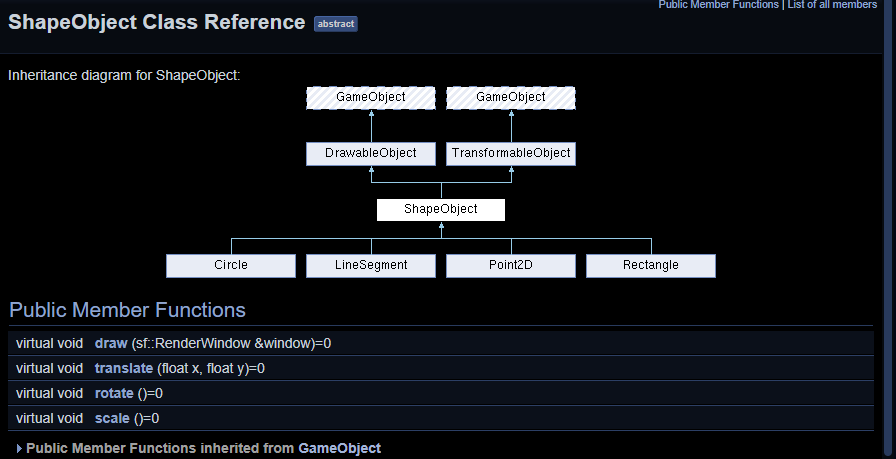
**1. GameObject**

* Base class for all game objects.
* Provides a virtual method **update()** for updating the object.

**2. UpdatableObject, DrawableObject, TransformableObject**

* Virtual base classes extending **GameObject**.
* **UpdatableObject** provides a pure virtual method for updating the object.
* **DrawableObject** provides a pure virtual method **draw(sf::RenderWindow& window)** for drawing the object.
* **TransformableObject** provides pure virtual methods for translation, rotation, and scaling.

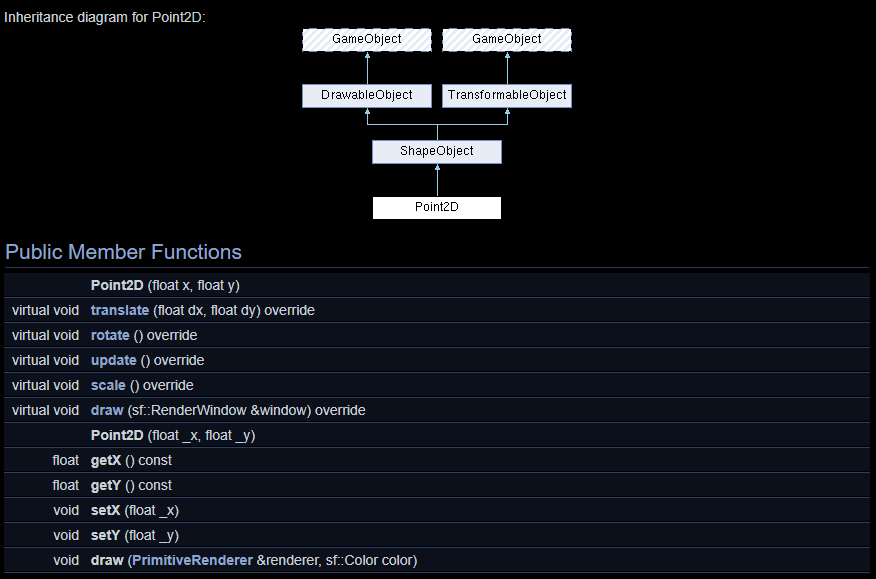
**3. ShapeObject**

****

* Extends both **DrawableObject** and **TransformableObject**.
* Provides pure virtual methods for drawing and transforming shapes.

**Specific Shape Classes**

**4. Point2D**

****

* Represents a point in 2D space with translation and drawing capabilities.
* Implements translation logic using **translate(float x, float y)**.
* Overrides virtual methods for rotation, scaling, updating, and drawing.

**5. LineSegment**

* Represents a line segment with translation and drawing capabilities.
* Implements translation logic using **translate(float x, float y)**.
* Overrides virtual methods for rotation, scaling, and drawing.

**6. Circle**

* Represents a circle with movement logic, collision detection, and reset functionality.
* Provides methods for getting position, radius, resetting, drawing, translating, and updating.
* Overrides virtual methods for rotation and scaling.

**7. Rectangle**

* Represents a rectangle with movement, rotation, and collision detection.
* Provides methods for drawing, translating, updating, rotating, and scaling.
* Overrides virtual method for scaling.

**Additional Object Classes**

**8. AnimatedObject**

* Virtual base class for animated objects.
* Provides a pure virtual method **animate()**.

**9. BitmapObject**

* Virtual base class for objects that can be drawn and transformed.
* Extends both **DrawableObject** and **TransformableObject**.

**10. SpriteObject**

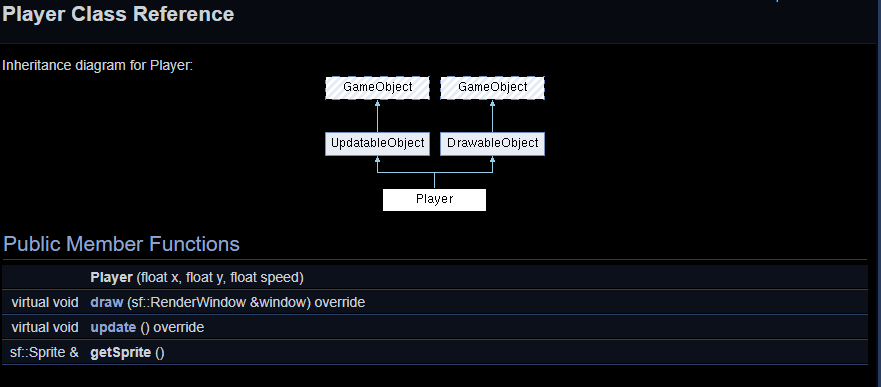
* Represents a sprite object that can be animated.
* Extends **BitmapObject** and implements the **animate()** method.

**11. GroundObject**

* Represents the ground in the game.
* Provides methods for drawing and updating.

**Game-Specific Classes**

**12. Player**

****

* Represents the player character in the game.
* Includes sprite animations, keyboard input handling, and collision detection.



* Provides methods for drawing, updating, and getting the sprite.

**Collision Handling**

**13. CollisionHandler**

****

****

* Static class with methods for checking collisions between rectangles.
* Handles collision events between game objects.

**Main Function**

**14. main()**

* The main function initializes the SFML window, sets up game elements, and implements the game loop.
* Handles keyboard input, updates game elements, checks for collisions, and continuously renders the game.

**Game Loop**

* The game loop updates and renders game elements.
* Background sprites move for a scrolling effect.
* Circles increase speed over time.
* Includes a "Game Over" screen with a delay before closing the window.

**Authors**

Berat Ferit OZGUL - 096859

Enes GOKDEN - 096815