Game Engine

This is a simple game engine project. Currently, it supports creating game objects, rendering them, changing their positions, and adding box colliders. A collision manager is also in development.

Features

- **Game Objects**: Create and manage game objects.
- **Rendering**: Render game objects on the screen.
- **Transformations**: Change the position, scale, and rotation of game objects.
- **Box Colliders**: Add box colliders to game objects for collision detection.
- **Collision Manager**: (Coming soon) Manage and detect collisions between game objects.

Getting Started

Prerequisites

- C++ compiler
- CMake
- SFML
- ImGui
- ImGui-SFML
- ImGuiFileDialog

Building the Project

1. Clone the repository:

   ```
   git clone https://github.com/yourusername/game_engine.git
   cd game_engine
   ```

PROF

2. Create a build directory and navigate to it:

   ```
   mkdir build
   cd build
   ```

3. Run CMake to configure the project:

   ```
   cmake ..
   ```

4. Build the project:

```
make
```

Running the Project

After building the project, you can run the executable:

```
./MyCppApp
```

to run tests

```
./tests
```

Usage

Creating game objects

You can create game objects and add components such as transforms and box colliders. Here's an example of how to create a game object and add components:

```
GameObject obj("Player");
obj.addComponent<Transform>(100.0f, 200.0f);
obj.addComponent<SpriteRenderer>();
obj.addComponent<BoxCollider>(100.0f, 200.0f, 50.0f, 50.0f);
```

Rendering Game Objects

PROF

The game engine automatically renders all game objects added to the current scene.

Changing position

```
Transform* transform = obj.findComponent<Transform>();
transform->setPosition(sf::Vector2f(150.0f, 250.0f));
```

Adding Box Collider

```
obj.addComponent<BoxCollider>(100.0f, 200.0f, 50.0f, 50.0f);
```