Overview of the Game Engine:

I have implemented the Entity Component System. Where Entities are mainly used to provide a unique Identifier, make the environment aware of the existence of a single Individual. Components are nothing more than the container. Each type of Component can be attached to an entity to provide some sort of property.

* **Graphics**:

Graphics is the class mainly used for initialization of SDL and to create a window. If SDL is initialized properly and window is successfully created then it notifies the Game Manager to create all other necessary instances and run the game loop.

* **Game Manager:**

Game Manager is the singleton class. This class hold the instances of all other managers in the Game. GameManager runs the game loop. It hold the instances of InputManager and creates the Game Objects as per the need.

* **Game Object:**

Game Object is the base class for any object created in the game. Example: Player, Enemy etc. Game Object will provide the basic properties required for GameObject. Each Game Object may or may not have the components. Each Game Object maintains a list of their owned Components.

* **ManageGameObjects:**

Manage Game Objects have the list of Game Objects. This class is responsible for calling Update method of all the objects.

* **Component:**

Component is the base class for all other components in the Game. Example: Sprite Component, Physics Component. Each component has its unique ID as well as type. Each Component will have the Parent. Parent could be any Game Object. Each game object can have one or more components.

* **Component Types:**

This class just maintains the list of Components in the Game.

* **Sprite Component:**

Sprite Component is responsible for creating Sprites. Play the sprite animation, Load the texture. If necessary it will communicate with other components of the same parent.

* **Texture Manager:**

Texture Manager is the Singleton class. He is responsible for creating a new texture whenever requested by the Sprite Component. If the texture is already exist then it will simply increment the reference of the texture and return the texture. He is also responsible for Unloading the Texture, when requested by Sprite Component.

* **Animation:**

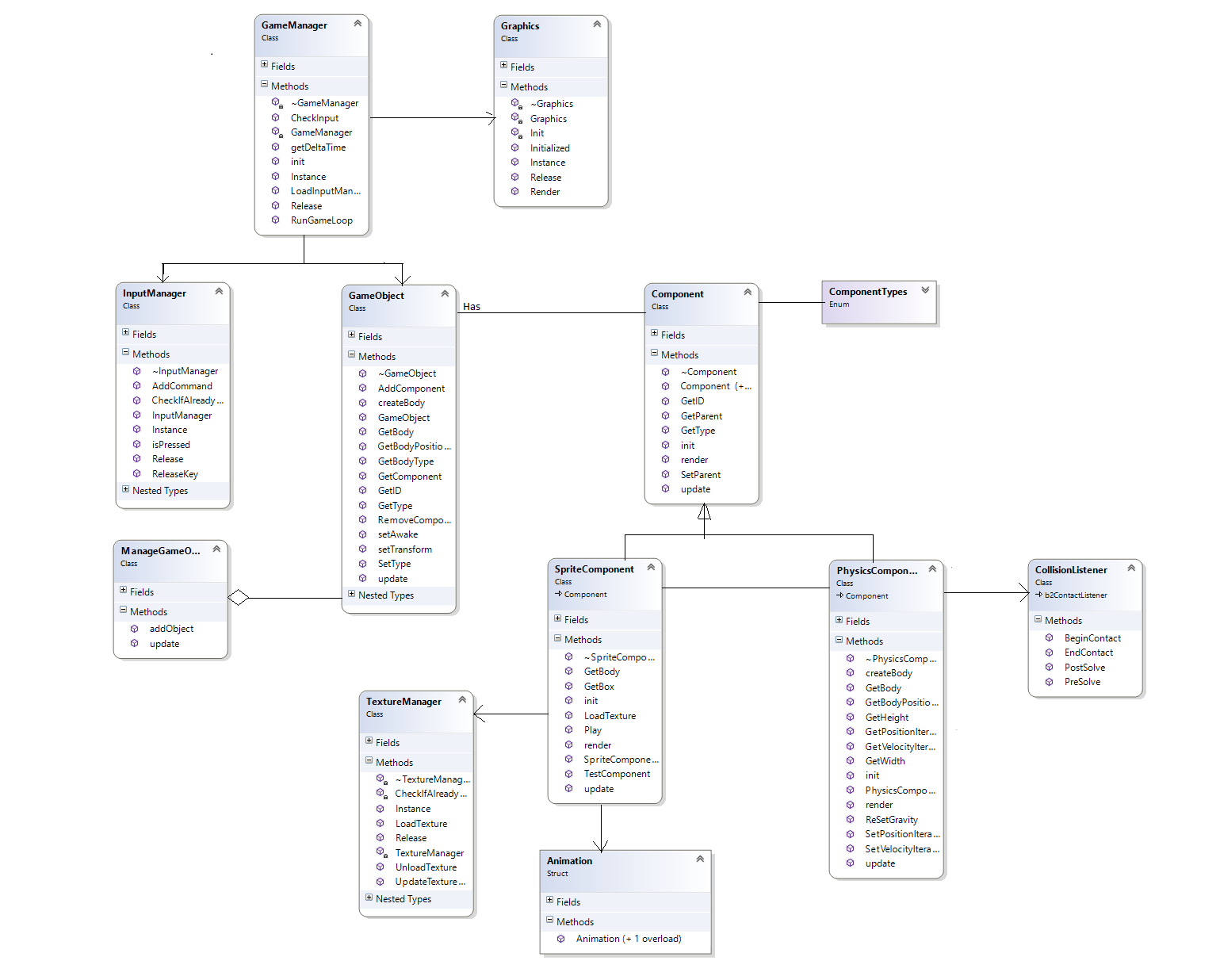
This Class hold the structure required for playing the sprite animation. Starting Index, number of frames to animate and at what speed.

* **Physics Component:**

This class is responsible for all physics related operations. With the help of Box2D Physics engine, it creates the world, body and perform all other physics operations on it.

* **Input Manager:**

Input Manager is singleton class. This class is responsible for checking if any input is pressed or not. If yes then notify the Game manager about it.



Class Diagram