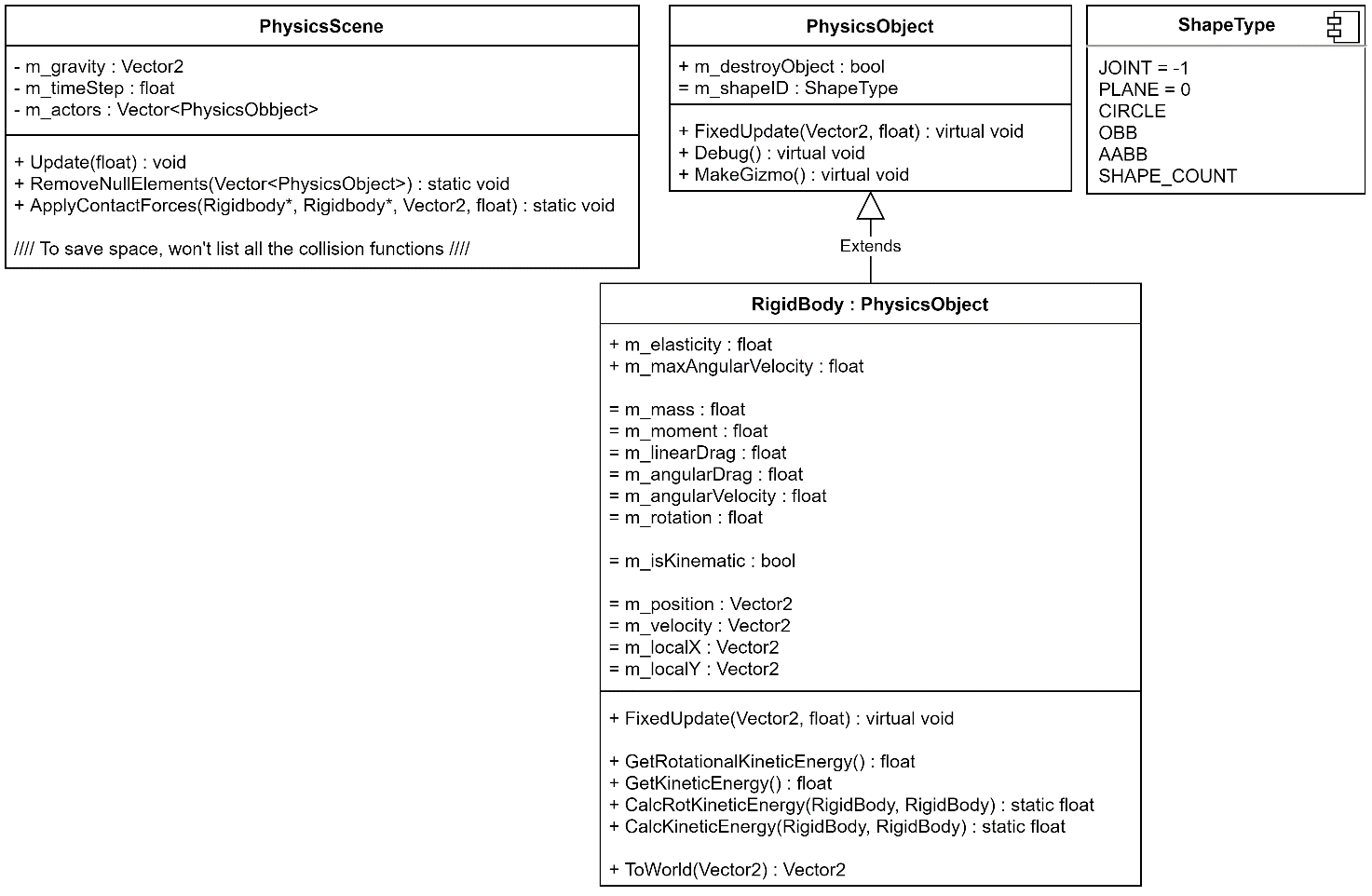
**2D Physics Game TDD**

The intent of this document is to help design and streamline the process of creating a 2D engine, and by extension game, using the aieBootsrap to handle graphical elements.

**Designing the Framework for the Engine:**

The above diagram showcases the basic structure of the higher levels of the engine (minus each individual class for each shape as this would be repetitive). From what is expressed above, the idea is to have a PhysicsScene that manages all the PhysicsObject currently loaded. A PhysicsObject being an arbitrary placeholder for all of its child classes (RigidBody and Plane). The next level down from this would be all the arbitrary shapes deriving from RigidBody so that they could be affected by physics events (such as collisions and applying forces), in contrary, the Plane class would derive from PhysicsObject directly as it won’t be movable since its an infinitely long and infinity thin object.

The proposed ShapeType will help to identify what type of PhysicsObject is being acted upon, of course due to polymorphism this wouldn’t usually be necessary however this may prove to be quicker and more efficient in its task.

**2D Physics Game Brief:**

The objective of the game will be to aim a small ball like character using the mouse to try and collect different kinds of collectibles in the game scene. The more collectibles that are acquired the higher the player’s score will be and that will allow the player to upgrade some basic stats, and in turn earn more points.

**Controls:** The main controls of the game will be clicking the left mouse button and moving the mouse to aim the direction that the player will be fired into, and then allowing gravity to carry out its work. There will also be a mouse operated menu system.

**Menu:** The menu will contain options for: playing the game, showing the tutorial and a set of upgrade parameters.

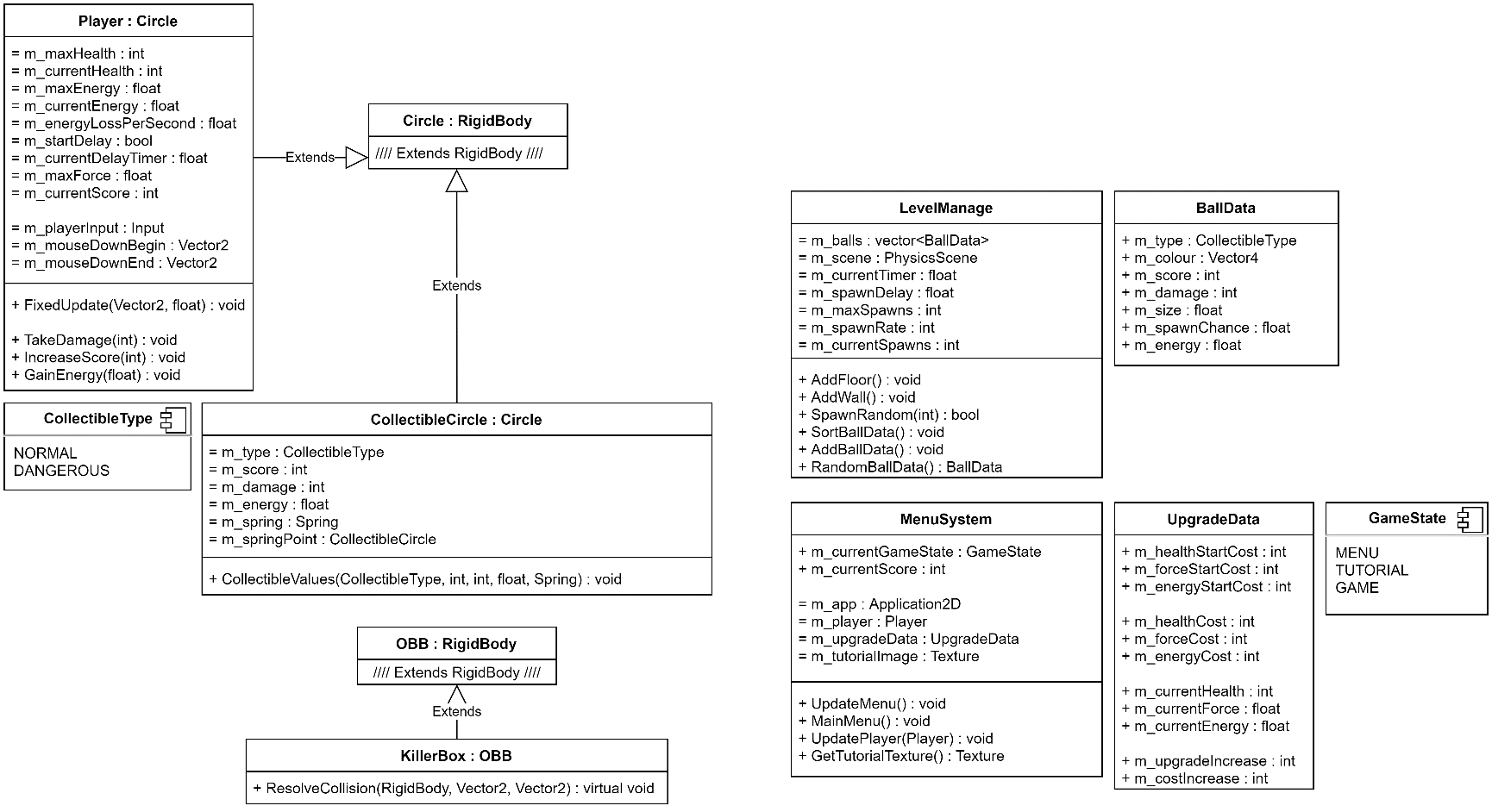
**Gameplay Loop:**  The player will collect balls that are scattered around the scene and gain points, when the player dies, they can spend these points to upgrade some base stats so that they can earn more points next game.

**Stats:** The driving force for the game, the player will collect points to upgrade these so that they can collect more points and so on.

1. **Health:** Upgrading health will give the player more life and allow them to survive more dangerous balls before dying.
2. **Force:** This will represent the max speed in which the player character can move based on the direction the user fires them and how far back they drag.
3. **Energy:** Energy is what allows the player to slow down time when they are aiming, having more allows for the player to slow down time for longer.

**Gameplay Mechanics:** The only to mechanics present in this game will be:

1. **Aiming:** As described above, the player will be able to aim the direction they go in and control how much force they exert.
2. **Slow Time:** The other mechanic is tied to aiming, so that whenever the player is aiming time will be stopped for a certain amount of time (based on the player’s current energy) and will be slowly depleted.

**Designing the Core Game Framework:**

From the basic framework of the physics engine I will be deriving a several more classes to aid the creation of the game. In this diagram it is clear to see that I have created a new ColelctibleCircle class that derives from Circle (which in turn derives from RigidBody because I want them to be affected by physics). This will help streamline the creation of collectibles in the scene thanks to the already predetermined functions and classes.

In order to manage the game, I created a separate class (LevelManage) that will be responsible for taking care of the game aspects of the engine (rather than baking it into PhysicsScene). The main purpose of this class will be to spawn in new collectibles when possible (based on a max number of collectibles at once). I also created another class to make it easy to create new CollectibleCircles in a orded random way, the LevelManage will choose a random BallData (defined by me) and spawn that type of CollectibleCircle with those details. This should streamline the process and make it very easy to add in new collectibles.

I thought it would also be a good ide to implement a menu and graphical interface for the player, this also ties in with how the player will upgrade their stats. To upgrade stats, I thought it would be a good idea to create a class that simply deals with how much stats will upgrade by, how much they will cost and other relevant information. This should make it very simple to tweak values and stats as necessary.