1. Git Repo: <https://github.com/RunningOnEmptyGBC/ThomasTheTankEngine.git>

2. Create your project structure.

* Assets
  + Holds Assets
* Docments
  + Documentation
* Game
  + Release builds
* Source
  + .cpp and .h files
* Temp
  + Temporary files
* Test
  + Debug builds
* ThomasTheTankEngine
  + Solution and project files

3. Agreement and Coding standards

1. {} always on the next line
2. variable naming convention camelCase
3. Class naming convention PastelCase
4. For loop generic variable [i,j,k]
5. Fixed array size when identified
6. Proper spacing

CODE EXAMPLE:

**EXAMPLE\_CLASS.cpp**

**#include “EXAMPLE\_CLASS.h”**

**Example:: Example()**

**{**

**//Constructor stuff;**

**}**

**Example:: Void Function()**

**{**

**if(numberOfThings > 0)**

**{**

**for(int i = 0; i < 10: i++)**

**{**

**Cout << arrayOfThings[i]<<endl;**

**while(int j = 0; j < 10; j++)**

**{**

**do**

**{**

**cout << endl;**

**}while(int k = 0; k > 10; k++)**

**}**

**}**

**}**

**}**

**Example:: ~Example()**

**{**

**//Destructor stuff;**

**}**

4. Component Architecture

1. Transform(Position, Rotation)
2. Mesh
3. Collision
4. RigidBody

<?xml version= "1.0"?>

<Actor>

<CubeComponent>

<Transform>

<Position x="1.385883" y="10.21463" z="-2.88255"/>

<Rotation x="3.461321" y="355.6992" z="204.3322"/>

<Scale x="7.2182" y="7.2182" z="5"/>

</Transform>

<Mesh>Cube</Mesh>

<Collider>

<IsTrigger>false</IsTrigger>

<Material>None</Material>

<Center x="0" y="0" z="0"/>

<Size x="1" y="1" z="1"/>

</Collider>

<MeshRenderer>

<CastShadows>true</CastShadows>

<RecieveShadows>true</RecieveShadows>

<Materials>

<Size>1</Size>

<Element0>White</Element0>

</Materials>

<UseLightProbes>false</UseLightProbes>

</MeshRenderer>

<RigidBody>

<Mass>7.2182</Mass>

<Drag>0</Drag>

<AngularDrag>0.05</AngularDrag>

<UseGravity>true</UseGravity>

<IsKinematic>false</IsKinematic>

<Interpolate>None</Interpolate>

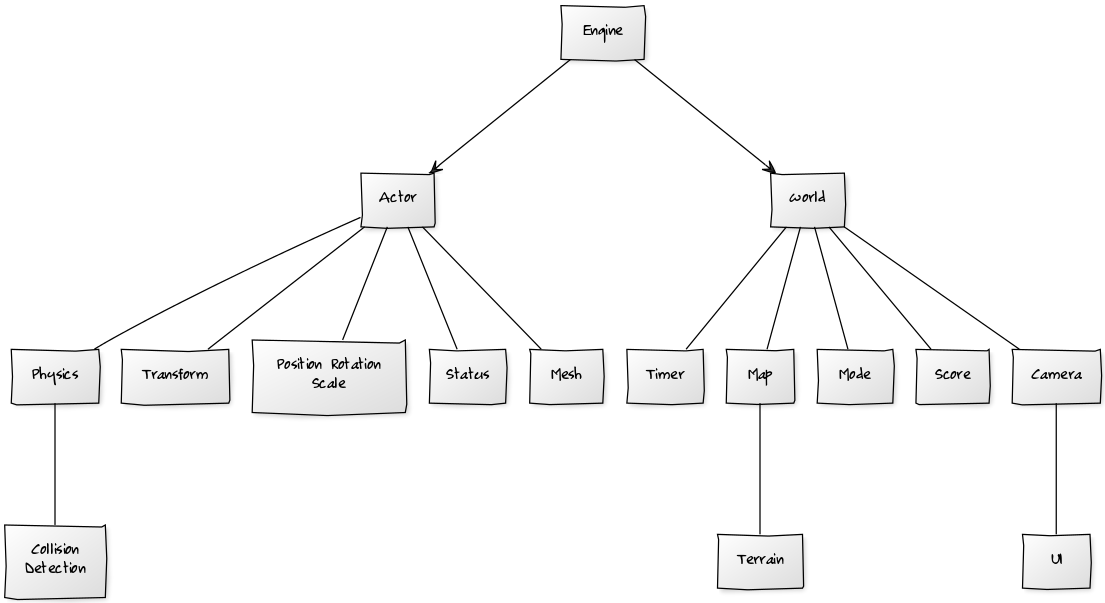
<CollisionDetection>Continuous</CollisionDetection>

</RigidBody>

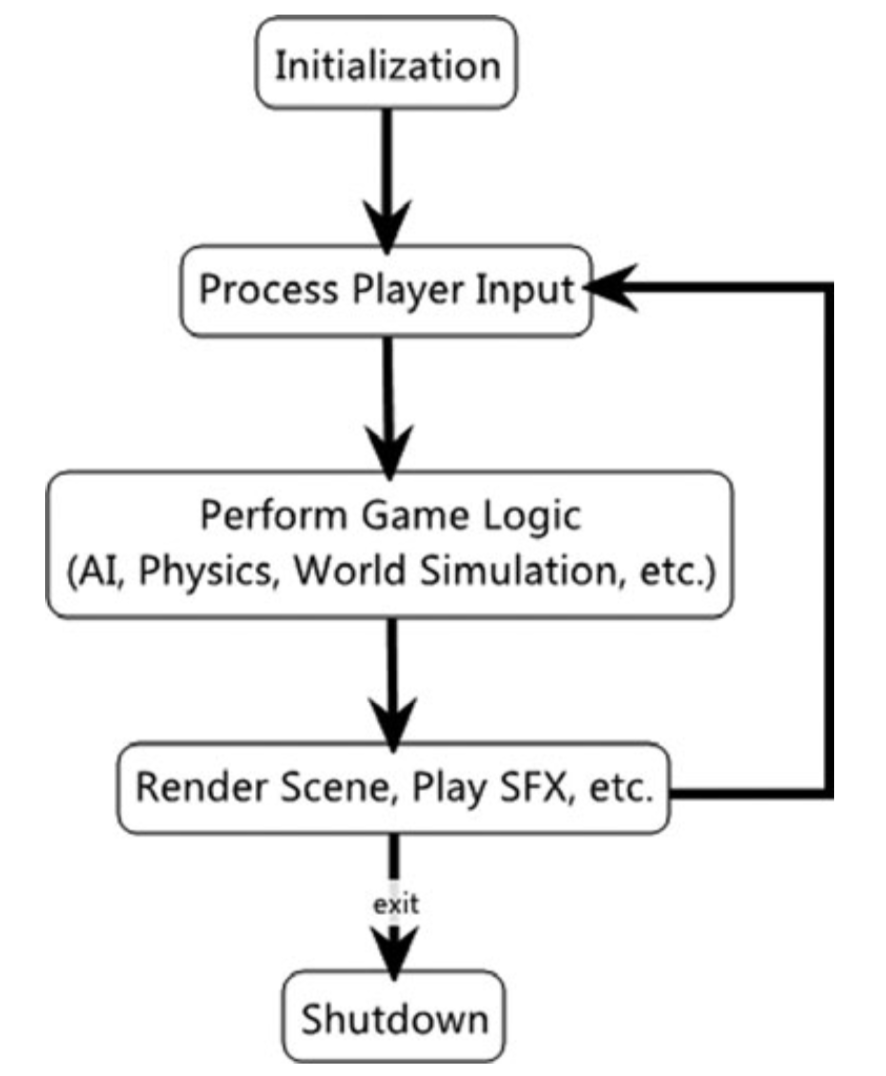
</CubeComponent>

</Actor>

5. A document that contains the UML engine class structure designed.

****

**6. Description of MaimLoop**



**7. Skeleton Class Files(.cpp and .h)**

**EXAMPLE\_CLASS.cpp**

**#include “EXAMPLE\_CLASS.h”**

**Example:: Example()**

**{**

**//Constructor stuff;**

**}**

**Example:: Void Function()**

**{**

**if(numberOfThings > 0)**

**{**

**for(int i = 0; i < 10: i++)**

**{**

**Cout << arrayOfThings[i]<<endl;**

**while(int j = 0; j < 10; j++)**

**{**

**do**

**{**

**cout << endl;**

**}while(int k = 0; k > 10; k++)**

**}**

**}**

**}**

**}**

**Example:: ~Example()**

**{**

**//Destructor stuff;**

**}**

**EXAMPLE\_CLASS.h**

**Class Example**

**{**

**private :**

**Int numberOfThings = 0;**

**String arrayOfThings[10] = [“Hello”,“Hello”,“Hello”,“Hello”,“Hello”,“Hello”,“Hello”,“Hello”]**

**public :**

**String Name = null;**

**Example()**

**Void Function()**

**~Example()**

**}**