**1)** One of the limitations of “Elasticity” is collision resolution. If you take a look at the UML diagram, you will see that the only components related to collision handling is the box collider component and ray tracing. We only detect if a cast ray is colliding with a box collider or not. We are not resolving collision among two actors with colliders. For example; when a bowling ball collides with a pin, the ball should change trajectory and the pin should bounce depending on the speed of the ball, and the masses of the ball and the pin. Since we don’t support rigid bodies, any calculation that considers speed, mass, acceleration, gravity etc. is not supported by our game engine.

**2)** If we want to make our game engine to support third person adventure RPG games, some of the changes we need to make on our game engine is as follows;

- Implement RigidBody Component that is going to be a child of the Actor Component. This component will add properties like mass, speed, acceleration, gravity etc. to the attached actor. Any actor with a rigid body will be affected by physics.

- Implement a Terrain Component that generates a terrain with different textures and heights along with a collider of its own. This will be used to create the 3D world of said RPG game. This component will have its own interface and will be child of the actor component.

- Implement an Animation Interface where the programmer can create a series of transforms on a mesh to animate it.

- Implement an Animation Controller Interface / Component that will enable to programmer to transition between animations attached to a certain mesh with ease. The animation Controller Component will act like a state machine and transition between given Animations based on transition conditions. The interface will be used by the programmer to add animations to the animator controller, create transitions and write transitions conditions. The Interface along with the component will go under the Actor component as its child.