Some updates to the design rationale

**Class NewActor :**

An abstract class extends from the Actor class in the engine code.

It is used by all the characters in the game.

Compare to the Actor class, it has a Boolean attribute isStunned which records if the actor is stunned or not in order to implement the affect caused by throwing a stun powder to an character (the character will skip 2 turns).

**Class NewPlayer :**

Class extends from new actor, uses to replace the player class in the engine .

Compare to the player class, this class, in each turn, it will check if the player is stunned or not. If the player is stunned, the playturn function will return SkipTurnAction for 2 turns.

Disadvantage:

Since the class is extends from the newActor class, it cannot extends from the player class in the engine. Hence, in the playturn function, a part of code in the engine player class is copy-pasted to this class.

**Interface Throwable:**

Contains two abstract methods:

* isHit( ): a method to control the possibility of hitting a target.
* affect( ) : the effect caused by the throwable object when it hits a character.