**Design Rationale for Requirement 2**

* Abstract classes ‘**HighGround**’ and ‘**LowGround**’ inherited from abstract class ‘Ground’ added.

**Justifications:**

* This allows us to easily differentiate between which grounds can be jumped to or walked down to.
* The HighGround objects will have slightly different characteristics than the LowGround objects and inheriting them respectively will make it easier for us to implement the ‘Jump’ action for our player.
* JumpActorAction class added.

**Justifications:**

* JumpActorAction is inherited from the abstract class action, as it is something the player can perform.
* It does not associate with ‘MoveActorAction’ as we were unsure whether jumping counted as moving. And, moving can be forwards, backwards, left or right, whilst jumping is done vertically, moving up layers. Hence, making it its separate class was deemed necessary.
* JumpActorAction alters the location of the player, similar to moving the player. They will be able to jump if they are next to the high ground and the location will change if the jump is successful. If the jump is unsuccessful, the location will not change.
* I have associated this action with the Location class as it is the one that shows the current Ground that the player is on. If we associated it with HighGround or LowGround for instance, there wouldn’t be anything that can show them.
* I have also added an association between Player and JumpActorAction. This is because only the player can perform the JumpActorAction, not any enemies. Hence, I thought it was necessary to associate this action with the player specifically, not the Actor class.
* Due to the JumpActorAction class being added, the MoveActorAction class needs to be modified to accommodate for both players and enemies moving to lower grounds. This modification will allow enemies to wander off high ground to lower ground and allow players to move from high ground to lower ground.