|  |  |  |
| --- | --- | --- |
| **Testing construction** | **Input Value** | **Expected Value** |
| Small values | Player(abc) | Pass construction |
| Big values | Player(abcdesdsd) | Pass Construction |
| empty | Player( ) | “IllegalArguementException” |

|  |  |  |
| --- | --- | --- |
| **Testing attack** | **Input Value** | **Expected Value** |
| The attack method should always know which player is attacking and not resulting in loss of other player’s health | Attack() | Health of player 2 should get changed on attack call by player 1 |
| The attack method should always make a change in health when necessary | Attack() | Health change should be there if there is no health change when the attack is of greater value |

|  |  |  |
| --- | --- | --- |
| **Testing Player methods** | Input Value | Expected Value |
| Defend() | Defend() | The defend method should decrease the effect of attack and change health accordingly |
| Defend() | Defend() | The defend method should affect the player that called it not the player attacking. |
| SelectHeadGear() | SelectHeadGear() | The assignment of the head gear should be done randomly for testing purposes another method should be done implementation. |

|  |  |  |
| --- | --- | --- |
| beltSelect() | beltSelect() | The assignment of the belt should be done randomly for testing purposes another method should be done in implementation |
| feetWear() | feetWear() | The assignment of the feet should be done randomly for testing purposes another method should be done in implementation |
| selectWeapon() | selectWeapon() | The assignment of the weapon should be done randomly for testing purposes another method should be done in implementation |
| enterArena() | enterArena() | [Player object, Player object] |

|  |  |  |
| --- | --- | --- |
| **Testing arena methods** | Input Value | Expected Value |
| Default game status true | Arena(true) | “IllegalArgumentException” |
| Default game status false | Arena(false) | pass |
| checkTurn() | checkTurn() | The method should return the players turn in correct order as a Boolean |
| checkStatus() | checkStatus() | The method should return the right status of the game ending as a Boolean |
| gameStart() | gameStart() | This method should make the calculations and start turn for the player |
| checkWinner() | checkWinner() | The method should return the correct winner of the game |