Table

Description automatically generated with low confidence

Table

Description automatically generated

Test plan

**HeadGear class:**

testHeadGear() Eg: initialize a new headGear which adjName is “happy”, use assertEquals and call getAdjName() to check whether it returns “happy”

testGetAdjName() Eg: initialize a new headGear which adjName is “happy”, check whether the getAdjName() is “happy”

testSetAdjName() Eg: initialize a new headGear which adjName is “happy”, call method setAdjName(“sad”), then check whether the getAdjName() is “sad”

testGetName() Eg: initialize a new headGear which adjName is “happy” and nounName is “hats”, check whether the getName is “happy hats”

testGetAttack() Eg: initialize a new headGear which attack strength is 0, check whether getAttack return 0

testGetAttackException() Eg: initialize a new headGear which attack strength is 2, since headGear is only useful for defense, so check whether the exception is thrown.

testGetAttackException1() Eg: initialize a new headGear which attack strength -2, check whether the exception is thrown.

testGetDefense() Eg: initialize a new headGear which defense strength is 0, check whether getDefense return 0

testGetDefense() Eg: initialize a new headGear which defense strength is 2, check whether getDefense return 2

testGetDefenseException() Eg: initialize a new headGear which defense strength is -2, check whether the exception is thrown.

testSetDefense() Eg: initialize a new headGear which defense strength is 0, call method setDefense(2), check whether getDefense() returns 2

testCombine() Eg: initialize a new headGear object(Happy hats -- defense strength: 2, attack strength 0), and take a parameter(Good Helmets -- defense strength: 3, attack strength 0), since player can only pick up 1 piece of headGear, check whether the exception is thrown.

**HandGear class:**

Similar to HeadGear class, except these tests.

testHandGear() Eg: initialize a new handGear which adjName is “happy”, use assertEquals and call getAdjName() to check whether it returns “happy”

testGetAttack1() Eg: initialize a new handGear which attack strength is 2, check whether getAttack return 2

testSetAttack() Eg: initialize a new handGear which attack strength is 0, call method setAttack(2), then check whether getAttack() returns 2

testGetAttackException1() Eg: initialize a new headGear which attack strength -2, check whether the exception is thrown.

testGetDefenseException() Eg: initialize a new handGear which defense strength is 2, since handGear is only used for attack, check whether the exception is thrown.

testCombine() Eg: initialize a new handGear object (Happy gloves -- defense strength: 0, attack strength 1) and take a parameter(Good swords -- defense strength: 0, attack strength 2) and use getter function to check whether they are combined(Happy, Good swords -- defense strength: 0, attack strength 3).

testCombineDiffTypeException() Eg: initialize a new handGear object (Happy gloves -- defense strength: 0, attack strength 1), and take a parameter(Good boots -- defense strength: 0, attack strength 2), since only same type items can be combined, check whether the exception is thrown.

**FootWear class:**

Similar to the above, except these tests.

testFootWear() Eg: initialize a new footWear object which adjName is “happy, use assertEquals and call getAdjName() to check whether it returns “happy”

testCombine() Eg: initialize a new footWear object (Sad Boots -- defense strength: 1, attack strength 0), and take a parameter(Happy HoverBoard -- defense strength: 3, attack strength 1) and use getter function to check whether they are combined(Sad, Happy HoverBoard -- defense strength: 4, attack strength 1).

testCombineDiffTypeException() Eg: initialize a new footWear object (Scurrying Sandals -- defense strength: 2, attack strength 0), and take a parameter(Happy Hats -- defense strength: 3, attack strength 0), since only same type items can be combined, check whether the exception is thrown.

**Character class:**

testCharacter(): Eg: initialize a new character object with name “A”, attackBaseValue 0, defenseBaseValue 0, use assertEquals to check the return values from corresponding getter methods

testGetName(): Eg: initialize a new character object with name “A”, check whether getName() return “A”

testSetName(): Eg: initialize a new character object with name “A”, then call setName(“B”), so check whether setName(“B”) return “B”

testGetAttackBaseValue(): Eg: initialize a new character object with attackBaseValue 0, check whether getAttackBaseValue() return 0

testGetDefenseBaseValue(): Eg: initialize a new character object with defenseBaseValue 0, check whether getDefenseBaseValue() return 0

testGetHead(): Eg: initialize a new character object which has 1 headGear, then check whether it returns that headGear

testGetHand(): Eg: initialize a new character object which has 1 handGear, then check whether it returns an array which has 1 handGear and a NULL

testGetFoot(): Eg: initialize a new character object which has 2 footWear, then check whether it returns an array which has 2 footWear

testAddGear(): Eg: initialize a new character object which has 2 footwear, 1 hand gear and 1 helmet, since only hand gear has available slot, call getHand method to check whether the character has 1 more hand gear.

testHeadAvailable(): Eg: initialize a new character object which has 1 headGear, then check whether it returns False

testHandAvailable(): Eg: initialize a new character object which has 1 handGear, then check whether it returns True

testFootAvailable(): Eg: initialize a new character object which has 2 footWear, then check whether it returns False

testMaxAttack(): Eg: let maxAttack take an array which includes several gears, then check whether it returns the gear that has max attack value

testMaxDefense(): Eg: let maxDefense take an array which includes several gears, then check whether it returns the gear that has max defense value

testAddGear(): Eg: initialize a new character object which has 1 handGear, then call addGear() method which takes a handGear as parameter, call getHand() to check whether it returns an array which has 2 handGear

testTotalAttack():Eg: initialize a new character object with 2 pieces of footWear(Scurrying Sandals -- defense strength: 2, attack strength 0; Happy HoverBoard -- defense strength: 3, attack strength 1), then use assertEquals to check whether attack strength is 1.

testTotalDefense():Eg: initialize a new footWear object with 2 pieces of footWear(Scurrying Sandals -- defense strength: 2, attack strength 0; Happy HoverBoard -- defense strength: 3, attack strength 1), then use assertEquals to check whether defense strength is 5.

testStatus(): Eg: initialize a new character object with 2 pieces of footWear(Scurrying Sandals -- defense strength: 2, attack strength 0; Happy HoverBoard -- defense strength: 3, attack strength 1),then check whether status() returns a string “Player1: Scurrying Sandals, Happy HoverBoard; defense strength: 5, attack strength 1.”