

# The Class Player

## 1 Specification:

- 1) The aim of the game is to collect as much gold as possible by defeating enemy characters.
- 2) If the player defeats an enemy he will gain the enemy's basic defence value in gold.
- 3) pick up – character attempts to add item on square to inventory (if the resulting total weight exceeds the characters strength, this will fail).
- 4) drop – drops one item (prompt user for the item to be dropped) to the current square (if the square already contains an item, request is ignored).
- 5) North, South, East, West – character moves to the square in the indicated direction .(north=up, south=down, east=right, west=left)
- 6) Player character attribute is divided into day(night), ie. Orcs are very good at night, but poor at day.
- 7) pick up – character attempts to add item on square to inventory.(if the resulting total weight exceeds the characters strength, this will fail)
- 8) drop – drops one item (prompt user for the item to be dropped) to the current
- 9) square (if the square already contains an item, request is ignored)
- 10) attack – character attacks enemy on current square (if this results in the characters health dropping to below zero, it is defeated)
- 11) look – prints out information about the current location (coordinates of the field and information about any enemies or items on that field)
- 12) Inventory – prints out the list of items the player is currently carrying and the amount of gold that the player has earned.
- 13) exit – ends the game.
- 14) The game starts on daytime. Daytime and Night-time alternate every 5 moves.
- 15) Each character can carry items up to a weight indicated by its strength.
- 16) If the health of any character drops to zero or below, that character is defeated(exit game automatically).

- 17) A character can carry only one item of each category(four categories), except rings.
- 18) Value is divided into day(night), i.e. Orcas are very good at night, but poor at day.
- 19) The attacker resolve the attack success determined by the attack chance of the attacker's race. On successful attack, the defender resolves the defence success determined by the defence chance of the defender's race.

## **1.1 Specification of Rand Generation**

### **1. Player (): construction.**

- Pre: none.
- Post: create a new player, default type is "human", and don't have any item.
- Para: none.

### **2. player (character choose):create a player**

- Pre: a player has been created.
- Post: create a player and the race as the user chooses. Don't have any item.
- Para: character choose- the player race

### **3. ~player(): destruction.**

- Pre: none.
- Post: delete the player and release the resources.
- Para: none.

### **4. PickUpInventory (NewInventory): pick up the item in this square.**

- Pre: existed a item in this square.
- Post: pick up this item from the square and the item disappear in this square.
- Para: inventory NewInventory.

### **5. PickupAddAttributes(inventory &NewInventory): change the player's attribute.**

- Pre: existed a player and have picked up an item.

- Post: change the attribute of this player by add the attribute value of the item.
- Para: inventory NewInventory.

**6. Inventory DropInventory(inventory&NewInventory,int&DropPrintNum):drop the inventory.**

- Pre: an item is held by the player.
- Post: drop the inventory and the item resource will be released.
- Para: inventory NewInventory, int DropPrintNum.

**7. Void DropSubAttributes(OldInventory): change the attribute after dropped the item.**

- Pre: drop the inventory successfully.
- Post: change the attribute of the player as without the inventory.
- Para: inventory OldInventory.

**8. void GoldAdd(int &GoldAddNum):add the gold after defected enemy succeffully.**

- Pre: defeat enemy successfully.
- Post: the gold number is added.
- Para: GoldAddNum, the number of gold will be added

**9. Void Defence(int &HarmVaule): overload the Defence to get a character.**

- Pre: select a role of the five types and user type the Attack command.
- Post: get the role player selected.
- Para: int HarmValue.

**10. void OrcAlter(int &steps): overload the OrcAlter and change the attribute of Orc every 5 turns.**

- Pre: selected the Orc as the role.
- Post: the attributes are changed every 5 turns (day to night or night to day).
- Para: int steps.

**11. void PlayerListPrint() const: print all the current information of the player.**

- Pre: get a role of the five types.
- Post: print the current information to the screen.
- Para: none.

## **2 User's View of Services**

- `player();`
- `player(character choose);`
- `virtual ~player();`
- `void PickUpInventory(inventory &NewInventory) ;`
- `void PickUpAddAttributes(inventory &NewInventory);`
- `inventory DropInventory(inventory &NewInventory, int &DropPrintNum);`
- `void DropSubAttributes(inventory &OldInventory);`
- `inline void GoldAdd(int &GoldAddNum);`
- `void Defence(int &HarmVaule);`
- `void OrcAlter(int &steps);`
- `void PlayerListPrint()const;`

### 3 Internal Data Representation

Variable	Type	Access
GoldNumber	int	Private
LifeRingNumber	int	Private
StrengthRingNumber	int	Private
PlayerType	character	Private
PlayerWeapon	inventory	Private
PlayerArmour	inventory	Private
PlayerShield	inventory	Private
PlayerRing	inventory	Private

### 4 Remaining Definitions

### 5 Coding