جامعة دمشق

كلية الهندسة المعلوماتية

مشروع مادة لغات البرمجة

إعداد الطلاب:

ضياء محمد قومان

عبد الناصر أحمد غدي

عبد الهادي معتصم بالله جراد

الفئة: 9

Team-Work Tasks:

Dyaa Koman for GUI interface.

Abdulhadi Jarad and Abdulnaser Ghadi for all console classes and functionalities.

The project contains all requirements in all functionalities.

Functionality 1:

ReadHero:

- Data structure: A readable file called (NewHeroes) contains all heroes' information.
- Algorithm: Read each line in the file, store each line as a hero and create object Hero that match the exact line by subtracting methods.

GroupFilter:

- Data structure: list of five random heroes that is matching the Groups that player want to be just in the game.
- Algorithm: Filter methods.

Store:

- Data Structure: list of champions that contains 5 random heroes to make the player buy one of them.
- Algorithm: created with the help of ReadHero & GroupFilter Classes, by reading from NewHeroes.txt , filter the groups and store 5 random of them in a list.

Hero:

- Data Structure: class that contains the Hero attributes (as inner class) like Hero ID, Owner and hero's position and put him in the Arena.
- Algorithms: some methods like:
  1- moveUp: This method accept number of move to a direction (Up for example) and checking if this move is valid then move hero to this place.
  2- incrementMana: Increment the mana for a hero.
  3- DisplayHeroInformation: Print All Hero's Attribute.
  4- isStunned: checking if hero is stunned to prevent him to do anything.

Player:

- Algorithm: methods like checkHeroLevelty, buyHero, Sell…etc.

InformationList (Extra):

- Data Structure: A writable file that records every changes of moving of heroes. We created it to simplify the project more.
- Algorithm: Normal writing file-methods.

Plan:

- Data Structure: An extendable abstraction class that contains list of players and a player.
- Algorithms: Abstract method getPlan.

PrimaryPlan:

- Data Structure: an overridden method getPlan from Plan class.
- Algorithms: For the first 9 rounds, the player decides if he want to buy or not a random hero from the store. This situation happening in getPlan method.

SecondaryPlan:

- Data Structure: an overridden method getPlan from Plan class for reading the plans. executePlan-method to execute them all.
- Algorithms: After the 9 rounds (In the execution rounds)
  This method will read a plan from a player (Attack, move, special ability activation), and execute it in the exceutePlans method.

Functionality 2:

Ability:

- Data Structure: an extendable abstraction class
- Algorithm: doTheAbility, abstract method enable the eachability on hero.

Aatrox, Azir… Ability classes:

- Data Structure: 48 classes each of them is the ability of a hero.
- Algorithms: an overridden method doTheAbility enable the special ability of the class considering the attack scope and magic scope.

Group:

- Data structure: an extendable abstraction class.
- Algorithm: enableAbility and disableAbility.

Assassin, Ninja...etc:

- Data Structure: 21 extended class from Group.
- Algorithm: enableAbility is an overridden method that enable the group ability, and disableAbility is an overridden method the disable the group ability.

Functionality 3:

Bench:

- Data Structure: a list of 5 random heroes that each round after the 9th will be displayed.
- Algorithms: methods:
  1- Buy Hero:

     Buy hero from bench store and put him in the list.

  2- Sell Hero:
     Sell hero in the arena and gain 50% of his price.
  3- Replace hero:
     Replace hero between the arena and the bench.

Functionality 4:

In the beginning of the game the user will determine the game rules like the arena's size, maximum number of replaces in each round, number of players, bots...etc.

Functionality 5:

- We used high qualified GUI library called Monkey in java, this library offers a lot of things like 3D scenes, real game-simulation

Functionality 6:

- We developed a bot that can play randomly.

Handling Exceptions:

- Handling Logic Exceptions:
  Like if the hero were killed, he will be instantly be removed from the list. Remove 3 heroes when the new hero will be at the next level…etc.
- Handling Input Exceptions:
  Like entering a string instead of number…etc.

Note: The project's **CLASS DIGRAM** is attached.