**Game Name:**

Heroes of Miracles

**Overview of the game:**

A hero wanders the field of the game board to find and defeat the dragon to restore peace in the land. On the way to the dragon the hero will encounter all types of dangers and collecting spoils from enemies to fight the dragon.

**Functions of the game:**

1. Player move the heroes around the map with the “wasd” (up, left, down, right) keys.
2. Player can open the menu item to view the hero’s statistics and to view and equip new weapon power ups earned by killing enemies.
3. Player fight enemies with turn based between the characters, they can choose which move to take from the battle menu.
4. Player has to kill enemy to earn rewards from them. Bring their HP to 0. If enemy brings your characters to 0 HP, you lose the game.
5. Player’s character’s HP will be carried across different battles, they can choose to heal up outside a fight or during a fight.
6. Player can choose to exit the game when they open the item menu and select to quit the game.
7. Player wins the game when their hero defeats the dragon, bring their HP to 0 while yours is above 0.

**Design Patterns Used:**

Strategy:

* Used to return directly what the weapon power up is at run time when an enemy is defeated and the hero collects the power up.

Decorator:

* Used when the player equips the power up to a weapon. The power up can be equip one after another (Max of 2) to a weapon.

State:

* Used when the player moves the hero character around the “game board”.

Template:

* Used as a default template for the type of different enemies that can spawn in the game with their statistics.
* Used as a default template for the type of different characters that the player can control in the game with their statistics.

Observer:

* Used when the dragon (boss) summons drakes during the battle. The boss gives its current data of statistics to the drakes.

MVC:

* Used when the UI sends command objects (containing the states of the map, the character and the enemies) to the controller, who in turn updates the models, and have them calculate the results, and send back the the results (the new states) to the UI.