

## Lab 5&6. Planning Write-up. Team 2 (Yasmeen, Yvette)

### Division of labor:

We both worked on all aspects of the lab (planning, coding, testing, diagram). Yvette fixed the game-end logic as per conversation with instructor during presentation on 10-9. Everything else works as it did during the presentation on 10-9, only the logic to end the game was added/modified. Diagram updated as well to add the new methods and attribute.

Here is the high level algorithm: (also see UML class diagram on next page)

### **1) Initialization:**

- Create decks of 20 cards for both players using a linked list.
- Each player has an initial HP (20).
- Both players draw a starting hand (1 card).

### **2) Gameplay Loop:**

- Until a player's HP reaches 0, repeat the following for each turn:

#### **a) Draw Phase:**

The current player draws one card from their deck.

#### **b) Play Phase:**

- The player can play cards from their hand (either a unit or spell card).
- If a **unit card** is played, it is placed on the battlefield. Name of unit card, which is an attack card: *Savage Strike*
- If a **spell card** is played:
  - + **Damage:** Opponent loses 1 HP. Name of Damage card: *Viper's Bite*
  - + **Heal:** Player gains 2 HP. Name of Heal card: *Divine Renewal*
  - + **Defend:** Player loses 1 HP but can defend against attacks. Name of Defend card: *Ironclad Wall*.

#### **c) Attack Phase:**

- The player can attack with units on the battlefield.
- The opponent can choose to defend with their own units
- Apply unit effects (player attacked loses 2 HP points when Savage Strike is used by their opponent).

#### **d) End of Turn:**

- Switch to the next player.

### **3) End Condition:**

- The game ends when a player's HP  $\leq 0$ , and the opponent is declared the winner.

**Note:** the code will use a linked list, use the concepts of polymorphism and inheritance and will have a GUI interface.

UML diagram (see next page)

