

## **Procedural Programming**

### **Project**

Your project is to write a program that implement the game called "War" which is outlined below.

#### **Game Overview:**

The game is based on the card game War - where each player flips over a card, highest card wins.

Number of players: Between 2-10 players

Game lasts: 13 rounds

Setup: Each player gets 13 cards - one of each "suit": 2,3,4,5,6,7,8,9,10, J, Q, K, A

Note: Player can see that cards in their hand but cannot see their opponent's cards.

Scoring:

- 2-10 = 2-10 points (face value)
- J (Jack) = 11 points
- Q (Queen) = 12 points
- K (King) = 13 points
- A (Ace) = 14 points

#### **How the game is played?**

Round: Each round, a player chooses 1 card to play from their hand. The goal is to have the highest UNIQUE card. You can only use each card once.

Round 1 example:

- Player 1 = 5
- Player 2 = 8
- Player 3 = K
- Player 4 = 2

- Player 5 = 8

In this round, Player 3 wins this "war" with the highest card of a K. So Player 3 gets 5+8+13+2+8 points (36 points). Let's continue on with a round 2 example:

Round 2 example:

- Player 1 = A
- Player 2 = Q
- Player 3 = A
- Player 4 = 4
- Player 5 = Q

In this round, Player 4 wins since he has the highest UNIQUE card. The two Aces take themselves out, and the 2 Queens take themselves out - leaving the winning card the 4. (So player 4 gets 56 points!)

Continue for a total of 13 rounds. The goal is to have the highest played unique card each round. At the end of the game, the person with the most points win.

### **Special Rules & Notes:**

ALL TIE RULE: If all cards tie another (example of 5 players: A,8,8,A,8) - then those points are rolled over to the next round - making the next round a VERY important round. If a tie like this happens on the last round, then those points are just "lost on the battlefield"

### **How the program should operate:**

- Every time a new game is created the program should ask how many players are playing and the player's cards should then be randomly generated.
- If a saved game is loaded, then the program should display the current status of the game.
- Once the game is loaded the users should be presented with a number of options:
  - Complete the next round
  - Save the game
  - Output the games status
  - Exit the game without saving

- Selecting this option exits the game but you are then given the following options:
  - Open a new game
  - Open a previously save game
  - Exit the application

### **Project Submission**

Each student should submit the code developed to support the game. In addition to the code each student should submit a document explaining the various design decisions that were made during the project and how their code works.

### **Submission Deadline**

Project Submissions to be submitted to [martin.hynes@gmit.ie](mailto:martin.hynes@gmit.ie) by 5pm on the 5th January 2018

### **Points to Note**

- Comments expected.
- Good programming practice is expected
- Ensure that any files that are needed to run the program are included in your submission.
- Your code has to run. Serious deduction of marks will occur for code that is not running correctly.
- Plagiarism is not acceptable.
- You will be asked to present your code/application