

# Week 11 & 12 Practical

## En Garde Game

---

### Objectives

The objectives of this practical is to test your knowledge of C#.NET by creating an application which satisfies the specification given below.

### Reading

To complete this practical you are expected to have completed practicals for Weeks 2 - 10. You will need the skills described in those practicals to complete this practical.

### En Garde Specification:

En Garde is a card game designed by Dr. Reiner Knizia which simulates the sport of fencing. In brief, the fencers start at opposite ends of a mat and move backwards and forwards trying to touch their opponent with a foil or fencing sword.

In the game, the mat has 23 spaces. The two players start at opposite ends and move forwards or backwards — there is no sideways movement. You cannot move through the other player or backward off the mat, so your movement is limited to the spaces between your start space and your opponent.

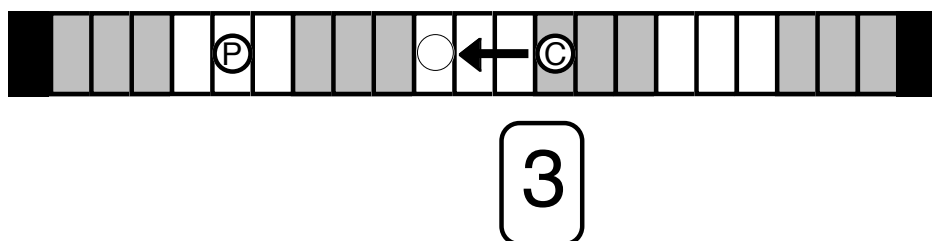
There is a 25-card deck consisting of five each of the numbers 1 through 5. Each player has a five card hand. Players take turns to play one card, move the exact number of spaces forward or back, then draw a replacement card. If you play a card which would take you to your opponent's space, instead of moving, you attack (thrust with your foil) from where you are. In the simple game, every attack leads to a touch (victory). In the normal game, your opponent can parry the attack by playing a card equal to the number card you played.

The round ends when someone is touched, or cannot move, or the 25th card is drawn. If a player cannot play a card, then they have lost the round. If no one has made a touch when the 25th card is drawn, then the winner is the player who has moved furthest down the mat.

## Basic Game

At the start of each round, the two Swordsman figures are placed onto the two end spaces of the board and all the cards are shuffled. Each player is then dealt 5 cards which they look at without showing them to their opponent. The rest of the cards are placed face down in a pile by the board. The human player will always start first.

Players alternately take turns. The player whose turn it is to play, chooses one of their cards and then moves their Swordsman the exact number of spaces on the board as shown on the card (Diagram 1). The player can decide whether to move forwards or backwards. The swordsman may not move off the board, nor may they move onto or over the space occupied by the other swordsman.

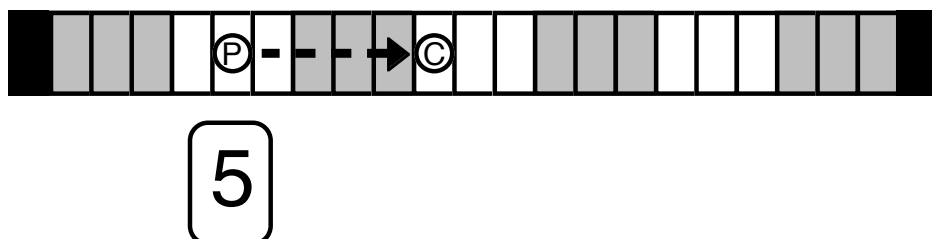


*Diagram 1: Movement of three spaces*

At the end of their turn the player takes a replacement card from the pack so they once again have five cards in their hand, then their opponent takes their turn in the same way. Cards played are discarded.

## Attacks and Touches

When a player is able to play a card which will allow their swordsman figure to move exactly onto the square occupied by their opponent then, instead of moving, they call "Attack". In the Basic game every attack leads to a touch (Diagram 2).



*Diagram 2: Attack and touch*

## Ending and Winning

The game ends as soon as a player achieves a touch, this player wins the game. Alternatively the game ends when one player can

make no legal move. This player loses the game. If no player has won or lost, the game ends when one player draws the last card from the pack. The game should show the remaining number of cards in the pack at all times. Once the last card has been drawn neither swordsman figure may be moved further, but the opponent has one last chance to attack and so win the game.

If the game ends without a player winning it outright, it is won by the player whose figure has moved furthest down the board, if both figures have moved the same distance then the game is drawn.

### **Standard Game (Optional)**

As in the basic game, a player attacks by playing a card which would allow their figure to move exactly onto the space occupied by their opponent. They then take a replacement card from the pack so they once again have five cards. The opponent now has the opportunity to "parry" an attack. To do this they must play a card with the same value as the attacker's card.

For example, if an attacker has played a "3" card as an attack, then their opponent can only parry by playing a "3". If the defender is unable to parry then the attacker scores a touch and the game is over. If able to parry, the defender replenishes their hand back up to five cards and then proceeds to take their turn as normal.

### **Requirements:**

Write a C# application to play En Garde. The user will always play the swordsman on the left, and the computer will always play the right swordsman.

You can represent the board by drawing rectangles in a picture box. To represent each player you can fill a rectangle with a different colour for each player or if you want to get fancy display a bitmap image for each player.

You need to decide how you will display the human player's hand and how the player can choose which card to play. You could use a button for each card in the player's hand which display the value of the card and the user could click on that button to play that card. We are not really worried about the usability of the application so as long as the player can see their hand and choose a card to play, that is fine. The user also needs to be able to indicate whether it is a move forwards or backwards. If the move is invalid, the user must be prompted to make another move. After each move, the board must be re-displayed showing the current state of the game.

The program must select the computer's move. It does not have to be particularly smart, but it cannot make an illegal move.

The program must automatically detect when a player cannot move. The program must detect when the game is over. It should display a congratulatory message to the winner.

## Program Design

You will need to decide how to store the information about the mat, the players positions on the mat, the deck of cards and the cards in both players hands. We would suggest you use a List to store the information.

For instance you could use a list of integers to represent the mat with each element in the list being a position on the mat. A value of 0 could represent an empty position, a value of 1 could represent the human player is at that position and a value of 2 could represent the computer player is at that position.

You will need to think carefully of how to break up the code so that you don't all of the code in click event methods. Think what tasks the application will need to perform and then write methods for those tasks. You could draw a diagram showing which methods will call other methods with the click event methods at the top of the diagram.

Possible methods might be: **IntialiseGame**, **DrawMat**, **SuffleDeck**, **PlayHumanHand**, **PlayComputerHand**, **ReplenishHand**, **CheckWinner**.

See Nilesch to talk about the design of the game before starting.

**Weeks 11&12: Practical REVIEW PAGE    Name: \_\_\_\_\_**

Questions:      Complete the following questions. (Note: this must be done before you seek a verification).

1.      What did you learn from the gadgets workshop?
  
  
  
  
  
  
  
  
  
  
- 2      Did you have fun creating the En Garde game?
  
  
  
  
  
  
  
  
  
  
3.      What was the most important thing you learned in COMP103.

Verification:      To assess your competency of this material your demonstrator will verify that you have:

1. Created the game.
2. Answered the set questions.