

**Mr. Giansante**



# **Visual Basic**

**Senet**

**August 2016**

# Senet

## Difficulty Level:

Beginner  
Intermediate  
Advanced

## Help References:

Random Numbers



Design a Visual Basic program that simulates the ancient Egyptian game of Senet. Senet, which means "passage" is a backgammon-like game. Several variations of the game exist (including one in "Tomb Raider: The Last Revelation").

A player cannot move one of their pieces on a square that already contains one of their pieces.

If a player's piece lands on a square occupied by the other player's piece, the other player's piece is sent back to the start.

The first player to get all of their pieces off the board wins the game.

## How to play Senet

As mentioned above, there are many variations of Senet, each with slightly different rules. For the purposes of our game, we will use the following set of guidelines ...

Both player start with three pieces.

Players throw four flat sticks to see how many squares they will move their piece forward. Each flat stick has a white side and a dark side.

**1**  If one white side is showing, the player can move a piece one space forward.

**2**  If two white sides are showing, the player can move a piece two spaces forward.

**3**  If three white sides are showing, the player can move a piece three spaces forward.

**4**  If four white sides are showing, the player can move a piece four spaces forward.

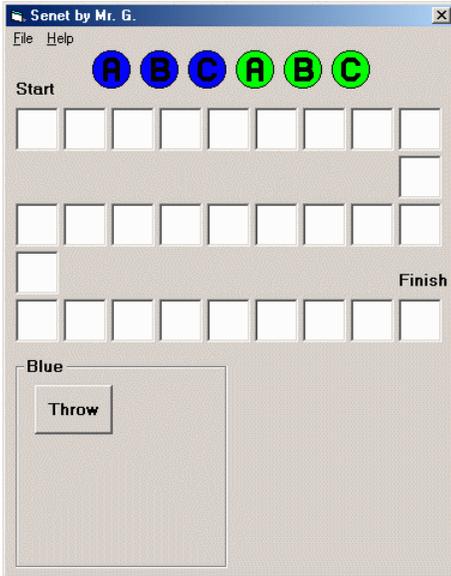
**6**  If no white sides are showing, the player can move a piece six spaces forward and also gets another turn.

# Senet

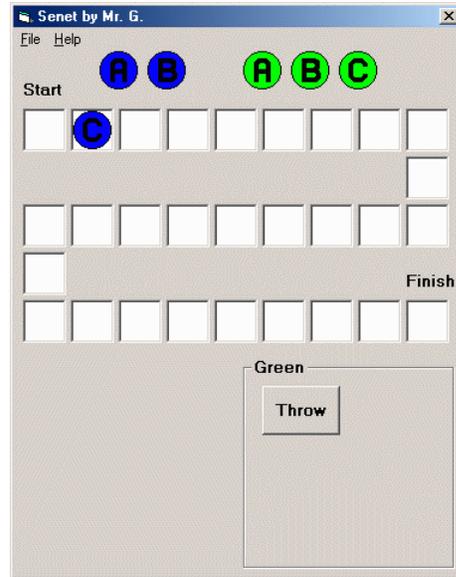
## Senet Screenshots

You are free to setup the user interface of your game as you see fit. The screenshots below are just suggestions.

The game starts.

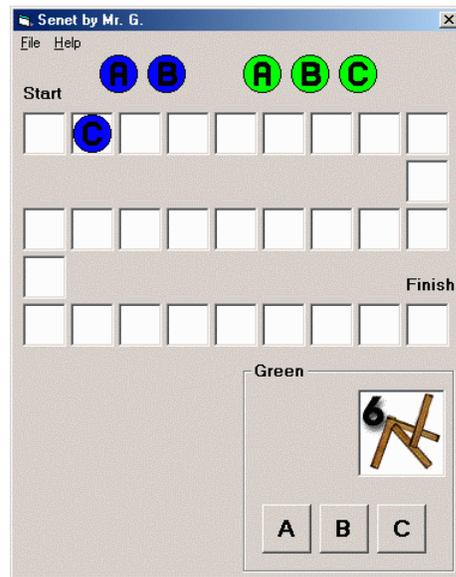
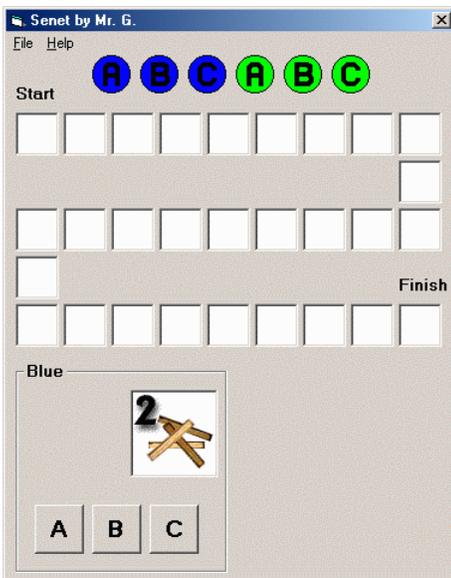


It is now the green player's turn to throw the sticks..



Once the sticks have been thrown (in this case the player got "6"), the player has the choice of which piece to move. Because the green player got a six,

Once the sticks have been thrown (in this case the player got "2"), the player has the choice of which piece to move.



Please fill out all the information in this column using a pen before getting this assignment marked.

\_\_\_\_\_  
Name

\_\_\_\_\_  
Date

\_\_\_\_\_  
Class

## Academic Honesty

The work I am submitting is completely my own creation and has not been copied from anyone else's work. If I have received help on this project, the names of those who have assisted are listed below.

\_\_\_\_\_  
Signature

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## Pre-Marking

The following people have pre-marked this assignment: (minimum of two)

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## Deductions

Each  which is checked indicates the item was not satisfactory and results in the loss of one mark.

Deduction for late assignment is 1 mark per day.

Deduction for program crashing is 2 marks.

## User Interface

- Appropriate Text in Title Bar (Form.Text)
- Separators and Access Keys in Menu, Capitalization
- "Exit" DialogBox meets requirements
- "About" DialogBox meets requirements
- Efficient and Esthetically-Pleasing User Interface
- Appropriate controls are used
- Spelling and Grammar are correct

## Code / Programming Style

- Variables are declared and logically named
- Code is commented where appropriate
- Code is indented and spaced to show
- Code is efficient

## Work Ethic / Problem Solving Skills

- Problem Solving Skills are demonstrated
- Makes Productive Use of Time
- Only Seeks Help when Necessary

## Program-Specific Criteria

- Users can play in turn
- Users randomly move 1, 2, 3, 5 or 6 spaces
- When a user rolls a 6, they go again
- Users can select which token to move
- Users cannot move onto a tile they already occupy
- When a token lands on an opponent, the opponent is sent back to the start
- Tokens that get to the end are scored
- A winner is detected and a message is displayed
- New Games can be generated

## Comments

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Mark** \_\_\_\_\_ / 10