C Programming Game Project

B37VB – Praxis Programming

Heriot Watt University

## Contact details:

**Name:** James Dickson

**Student Number:** H00431469

**Email:** [jd2080@hw.ac.uk](mailto:jd2080@hw.ac.uk)

**GitHub Username:** jamesjwd

**GitHub Repository:** https://github.com/jamesjwd/B37VB-Game-Project.git

## Revision History Table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number:** | **Revision Date:** | **Nature of Revision:** | **Author Name:** |
| 1 | 26/03/2024 | README- JAMES DICKSON  Initial Draft | James Dickson |
| 2 | 01/04/2024 | README- JAMES DICKSON  Additions of the introduction, snippet of the code and the conclusion | James Dickson |

**Table of Contents:**

[Contact details: 1](#_Toc162952988)

[Revision History Table: 1](#_Toc162952989)

[Introduction: 2](#_Toc162952990)

[Game instructions: 2](#_Toc162952991)

[Game Design: 2](#_Toc162952992)

[Figure 1- a flow diagram to show the structure of the game 3](#_Toc162952993)

[Implementation of the code: 3](#_Toc162952994)

[Snippet of the code: 4](#_Toc162952995)

[Conclusion: 5](#_Toc162952996)

[References: 5](#_Toc162952997)

## Introduction:

The purpose of this report is to document the development and implementation of my game ‘Blackjack’ for the Praxis Programming Game Project Assignment. This game was programmed in C and is a text-based game with no graphical interface required. The only controls the user needs is the spacebar to roll the dice and to type either yes or no depending on the decision whether they wish to roll again.

## Game instructions:

This game takes inspiration from the famous card game, ‘Blackjack’ (also known as ‘21’). In this interpretation of the game, the player is to roll a 6-sided die multiple times in hopes they land on 21 in which case they win and achieve ‘Blackjack’. With each roll they choose to take, the player gains 1 point. If the player reaches 21 without surpassing 21, they gain an extra 5 points. If the player surpasses 21, they lose and they are unable to continue any more rounds. The player’s objective is to score as many points as possible over 3 rounds.

## Game Design:

The game starts with a brief explanation of the rules and objectives of the game. The player is then asked if they wish to start the game. Assuming the player wishes to continue, a for loop is run which allows the game to run for a limit of 3 rounds. In this for loop, a dice is rolled, and this number is printed for the player to view. Every time the dice is rolled, the player earns a point. Every number that is rolled is summed. An if statement is then run which confirms whether the players’ summed numbers that were rolled have not reached or surpassed 21. If this summed number is equal to 21, the player gets 5 points added to their score for the round. If this summed number is greater than 21, the player loses, their total score across all rounds is printed and the game ends. If the summed number is less than 21, the player is then asked if they wish to continue the round; if yes, the for loop will run again; if not then the round will end, and the player’s number of points for the round will be added to their total score, and they will move on to the next round. This for loop will repeat 3 times and after the final score will be printed along with a thank you message.

Figure 1 - a flow diagram to show the structure of the game.

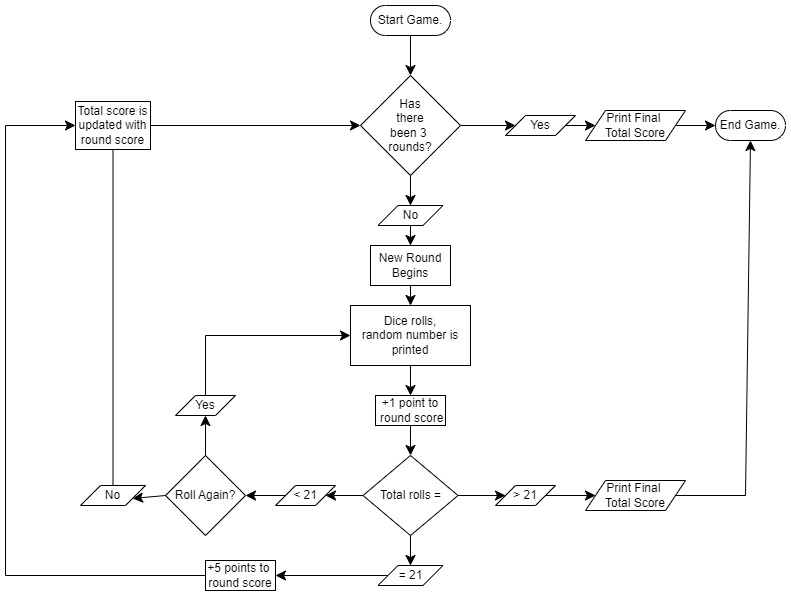


Figure 1- a flow diagram to show the structure of the game.

### Figure 1- a flow diagram to show the structure of the game

## Implementation of the code:

The game was coded using the C programming language using basic functions, loops and arrays to allow the user’s inputs and decisions to affect the outcome of the game. Some key components of the code were the ‘do…while’ loop which allows the code to run provided that the player has not already exceeded 21; the ’scanf(“…”)’ function allows the user to input yes or no the determine whether they wish to continue rolling and risk surpassing 21; and if statements allow the program to output several different texts depending the outcome of the player’s decisions.

## A Snippet of the code:

A screen shot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generated

Figure 2 - a screenshot of the code for the game.

## Conclusion:

In conclusion, the C programming game project provided practical experience and was a successful introduction to C. Overall, the project was a success as the game runs effectively and offers a simple interactive experience that anyone would be able to understand and enjoy. The game could have had some improvements, including the addition of a graphical interface to allow the player to see an animation of a dice rolling to make the user’s experience more immersive.

## References:

Here are some references to some of the code I used:

* Lines 45&46 that generate a random number between 1 and 6 are taken from:

Bro Code (2021) – C random numbers. Available at: <https://www.youtube.com/watch?v=CJ37J_Cdd8Q> (Accessed on 15/03/2024)

* Line 32&43 allow the user to press any button to continue and are taken from:

PREPBYTES (2023)- getch() function in C. Available at:

https://www.prepbytes.com/blog/c-programming/getch-in-c/#:~:text=Explanation%20for%20the%20code%3A,terminates%20using%20the%20return%20statement. (Accessed on 25/03/2024)

* Gained information about the strcmp() function for my yes/no conditional loop from:

Shiksha Online (2023)- Compare strings with strcmp function in C. Available at:

https://www.shiksha.com/online-courses/articles/strcmp-function-in-c/