|  |  |
| --- | --- |
| **Course code and name:** | B37VB – Praxis Programming |
| **Type of assessment:** | **Individual** |
| **Coursework Title:** | Game Project Dubai |
| **Student Name:** | Mohammed Ayman Munshi |
| **Student ID Number:** | Game Project Dubai |

|  |
| --- |
| **Declaration of authorship. By signing this form:**   * **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section. * I confirm that I have read, understood and followed the University’s Regulations on plagiarism as published on the [University’s website](https://www.hw.ac.uk/uk/students/studies/examinations/plagiarism.htm), and that I am aware of the penalties that I will face should I not adhere to the University Regulations. * I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on [Academic Integrity and Plagiarism](https://heriotwatt.sharepoint.com/sites/skillshub/SitePages/Academic-Integrity-and-Plagiarism.aspx)   **Student Signature** *(type your name): Mohammed Ayman Munshi*  **Date**: *13/04/2025* |

Copy this page and insert it into your coursework file in front of your title page.

For group assessment each group member must sign a separate form and all forms must be included with the group submission.

**Your work will not be marked if a signed copy of this form is not included with your submission.**

Praxis Programming Game Project

By Mohammed Ayman Munshi

(H00472078)

(mm3047)

# INDEX

1. **Introduction** …………………………………………………….… 4
2. **Method/Design**…………….……………………………………… 4-5
3. **Game flow**……….…………………………….…………….…… 6
4. **Conclusion**………………………………………………………… 7
5. **References**………………………………………………………… 7

# INTRODUCTION

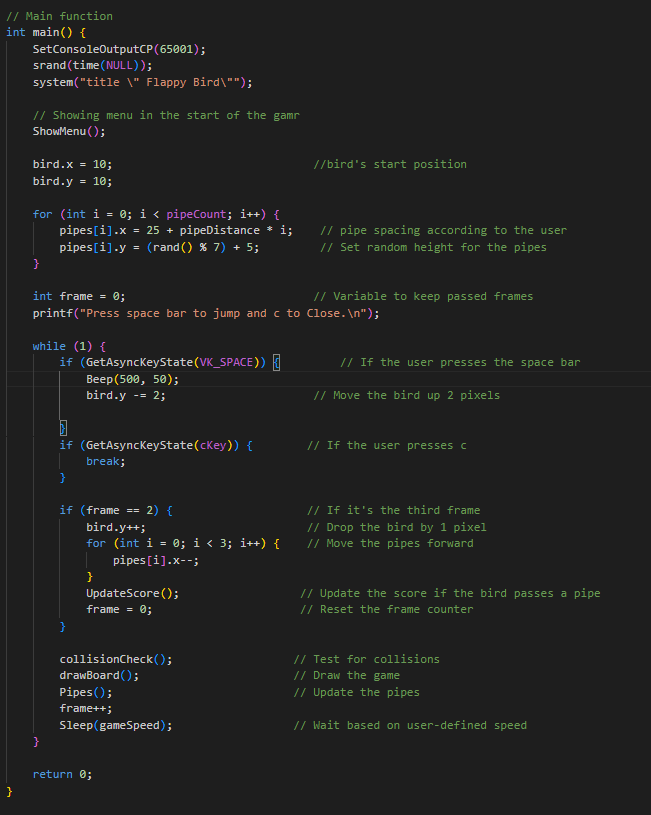
This game is made in C the game is Flappy Bird. It is simple yet fun. I made this code in VSCode. The game is straight forward about its objectives you must go through the endless number of pipes for as long as you can and get a high score. By pressing the space bar to jump and press c to close the code.

The code has many additional features such as you can choose or customize the game difficulty, color of the pipe and the color of the bird. There is also sound the plays each time the bird flaps and each time the bird passes through the pipe it makes a different noise and when the player crashes into the pipe or into the floor it plays a death tune.

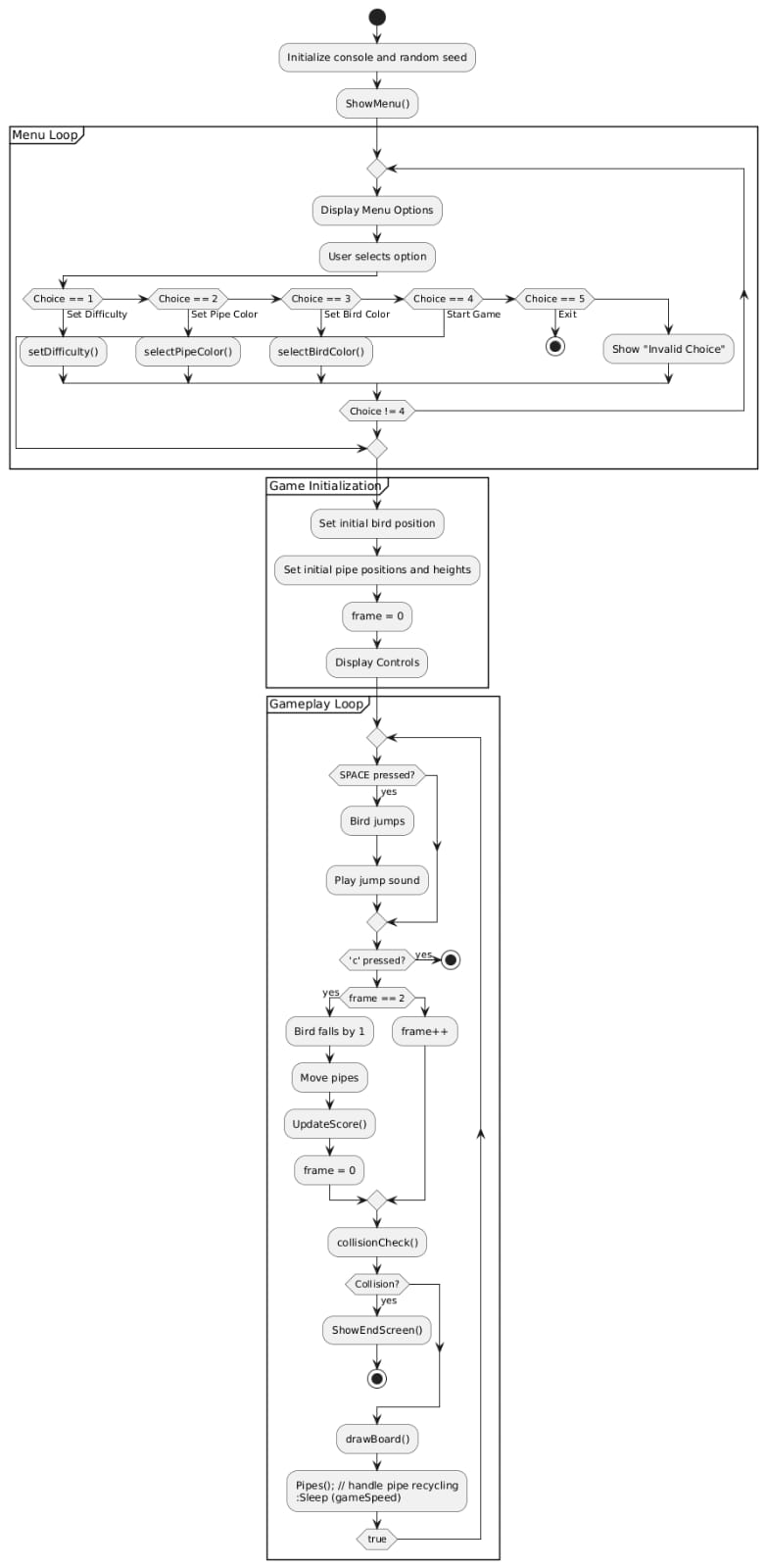
It also has an end screen where it displays your score and the high score.

# METHOD/DESIGN

* **Step 1** Declaring the variables, headers, constants etc.
* **Step 2** Customized settings for difficulty, colors for pipes and the bird and the menu
* **Step 3** The drawboard function being essential as it creates the board/screen that the whole game runs on. In the starting bit of the code, we are printing the score and creating the border for the game after that beginning half of the function, we are essentially creating the pipes on the screen and basically spawns them in. in the second half of the function we are creating the bird and we are also setting the color for them
* **Step 4** The pipes function gets rid of the pipes which have gone off the screen.
* **Step 5** The collision checks function check whether the bird has collided with either the ground or the pipes if so, it shows the user the end screen if not it continues the gameplay as normal
* **Step 6** The update score function updates the score every time the bird goes through the pipe. The load high score function loads or creates a text file which holds the information of the high score and the save high score function writes and saves the new high score if your score is higher than the high score
* **Step 7** The show end screen function initially creates a tune for the death sound after which it prints the end screen with your score and the high score it also displays wether if you have beaten the high score or not.
* **Step 8** The Main Function



# GAME FLOW



# CONCLUSION

In the end I got to learn a lot from doing this game project such as game design and how to set different things up. In the end the game works as desired and implements different functions which work together in unison.

# REFRENCE

* I took inspiration for the code from   
  fcharlie (<https://gist.github.com/fcharlie/b88b9011a0553fd65d5279e3cb25bc28>) for some parts of the code.