**Programming Assessment - Game Report**

**Unit Code and Title**: 6G4Z0020 Programming

**Submitted By**: Siddhant Frederick

**Student ID Number**: 24833739

**Course**: 1st Year BSc (Hons) Cyber Security

**Assessment Set By**: Dr David McLean

**Assessment ID**: 2CWK 50%

**Game Report – Introduction**

In the Programming module, we were taught concepts of Programming using Java as the main language through Processing which is a Java based open-source programming and development language to learn the concepts such as Variables, Loops, Object Oriented Programming, Animation and we had to implement these concepts in developing a game based on the game specification titled, “Don’t Crash!! Game” where the player must stay static and in the centre of the screen and the obstacles should be moving around the player and must avoid colliding with the player that is present at the centre of the screen. The obstacles are controlled by the arrow keys and along with having their own movement, they should move according to the specified direction done using arrow keys that are pressed by the player.

The Game that I have developed involves an aeroplane which is the player, present in the centre of the screen and the obstacles include 2 objects which are birds and hot air balloons, and they move around the aeroplane with their own movement and controlled by the arrow keys to avoid colliding with the aeroplane.

There are some snippets of the steps I took to put together all the pieces in developing the game which I would to like to share in this report.

**Game Report – Steps and Procedures**

1. **Development of the Player - Aeroplane**

**A screenshot of a computer

Description automatically generated**

This involved the development of the player aeroplane that must stay in the centre of the screen by developing a relevant class for the same.

It also includes the code for collision detection with both the obstacles which are bird and the hot air balloon in our game.

1. **Development of the Obstacles – Bird and Hot Air Balloon**
2. **Bird**

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer screen

Description automatically generated

In this code, we can see the development of the obstacle bird and the relevant class designed for the same.

1. **Hot Air Balloon**

A screenshot of a computer

Description automatically generated

A blue and black rectangular object

Description automatically generated

For our obstacle Hot Air Balloon, we have developed a separate class for the same however we can see that we have also used the concept of inheritance to derive all the properties of the obstacles from the class Bird.

1. **Development of Background for the game – Sky**

**A screenshot of a computer

Description automatically generated**

This involved the development of the class Background using a simple image of sky to add as a background for the game.

1. **Development of the Main Class**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

The Main Class file includes the code that is required to initiate the working of the game. The code in this file includes creation of object to access methods, designing the array list to store obstacles, conditional loop statements to access the obstacles and to end the game once the obstacles collide and appearance of Game Over text once the obstacle collides.

1. **Result**

A screenshot of a video game

Description automatically generated

**Game Report – Highlights and Conclusion**

Some of the things I have learned and the things I have used, and I could’ve improved upon are as follows:

1. **Use of Generative AI**: I did use Generative AI (ChatGPT) to help me correct my code and to generate the code for the logic where I was stuck, however I have ensured that I haven’t used many bits of it that is entirely the copy, and I have just used the logic to develop the game.
2. **Polymorphism and abstract class**: I was not able to use it since I was not quite clear on how to approach it however, I did at one point edit the code by using both, but it was generating too much error, and I wish to clear my understanding for both the topics in future.
3. **File Handling**: I didn’t find it necessary to include the same however I wish to learn more about it and improve upon it in future.

Overall, I found the assessment quite interesting, challenging and a fun to do which also helped me to develop my problem solving and critical thinking skills and I wish to enhance and broaden my knowledge for Programming in future. I would like to sincerely extend my gratitude to Dr David McLean and the entire Programming team for their support and help during this time and delivering the content in the best way possible which I found well-suited and up to the mark.