**DEPARTMENT OF INFORMATION TECHNOLOGY**

**DELHI TECHNOLOGICAL UNIVERSITY**



**Platformer Game**

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

**BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY**

Submitted by: -

***KUNAL CHAUHAN 2K19/IT/069***

***IKROOP SINGH KALSI 2K19/IT/061***

Under the Supervision of

**Mrs. Ritu Agarwal**

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

**CANDIDATE’S DECLARATION**

We, **Kunal Chauhan 2K19/IT/069 and Ikroop Singh Kalsi 2K19/IT/061**, students of B.Tech. (INFORMATION TECHNOLOGY), hereby declare that the project Dissertation titled “**Project SNEK**” which is submitted by us to the Department of INFORMATION TECHNOLOGY, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi Ikroop Singh Kalsi Date: Kunal Chauhan

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

**CERTIFICATE**

I hereby certify that the Project Dissertation titled “**Project SNEK**” which is submitted by **Kunal Chauhan 2K19/IT/069** and **Ikroop Singh Kalsi 2K19/IT/061**; INFORMATION TECHNOLOGY, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi **Ritu Agarwal**

Date: 02-12-2020 **SUPERVISOR**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of

Engineering) Bawana

Road, Delhi-110042

**ACKNOWLEDGEMENT**

I would like to convey our heartfelt thanks to our supervisor Mrs. Ritu Agarwal for her ingenious ideas, tremendous help and cooperation.

I am extremely grateful to my friends who gave valuable suggestions and guidance for completion of my project. The cooperation and healthy criticism came handy and useful with them.

Finally, I would like to thank all the above-mentioned people once again.

**ABSTRACT**

In today’s world when each and every one is running after something or another, we humans are ignoring the most important thing that is ourselves. And here video games are one of the most efficient things that helps in distress and calming one. We all at some point in our life loved to play games and have encountered one or another game that becomes our favorite.

And Here we came and tried to remaster a classic game in a retro fashion inspired by the retro game developed in 1997 by Taneli Armanto. Contrary to the traditional belief that gaming is merely an addictive source of entertainment and diversion, recent research has proved that gaming has numerous benefits and key among them, is the development of cognitive skills in both children and adults. Just as physical exercise helps in improving and strengthening your muscles, cognitive games help to indulge one's brain in constant stimulation, thus improving the brain's performance. The following are some of the cognitive benefits of playing video games.

**1. Improves coordination**

When an adult or child is playing a video game, he or she is not only staring at the computer inactively. The activities and actions on the screen provide a lot of mental stimulation. For one to play, he or she will need to coordinate their visual, audial and physical movement.

**2. Improves problem-solving skills**

Video games involve certain rules. This means that the player has to think carefully before making any move to ensure that they stay within the required rules of that particular game. The player needs to make split- second decisions that will determine whether or not he or she will advance to the next level.

**3. Enhances memory**

Playing your favorite video game may require both visual and audial memory. The player is required to read or listen to the instructions which might only be provided at the beginning of the game, thus the need to remember them throughout the entire game. Mastery of the keys on your keyboard helps you easily move your characters in the game. This helps improve your memory, whether short- term or long-term.

**4. Improves attention and concentration**

Video games especially action games, have proven to be able to capture the player's attention for the entire period of the game. This is brought about by the player's need to achieve certain objectives within the game, and be able to progress to the next level.

**5. It is a great source of learning**

Gaming is not only beneficial to adults and teenagers, but to children as well. Many modern education institutions incorporate video games as a teaching methodology. This helps these children improve their academic skills by providing video games that are specifically aimed at enhancing their cognitive and creative skills.

**6. Improves the brain's speed**

While gaming, the brain receives multiple stimulations, both Visual and audial. According to research, individuals who play video games frequently can process these stimulators faster than others. These stimulators ensure that the brain is continuously working to interpret them.

**7. Enhances multitasking skills**

An action game, for example, may require you to be very observant. It requires you to be able to move your joystick or keys while looking at the various features on your screen such as energy levels, oncoming adversaries, ammunitions left, available time among other factors, all which are vital to winning. This ensures that the player can observe and react accordingly to all requirements of that particular game.

**8. Improves social skills**

Online gaming enables many players to engage in a particular game simultaneously. As such, there is constant communication between the players which in turn results in the development of meaningful as well as casual relationships among them.

This helps players meet new friends while also strengthening bonds with their old friends. Though computer games might be beneficial, there is need to play them in moderation. It is also important to pick the right game as not all of them provide the same cognitive benefits. Age should also be a factor. Small children should not be exposed to violent games.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Topic** | **Page** |
| 1. | Abstract | 5 |
| 2. | Table of Contents | 8 |
| 3. | Chapter 1: Introduction   * 1.1 Introduction to SNEK * 1.2 Modes and functionalities * 1.3 History & Inspiration for SNEK * 1.4 Pygame(The Core of Game) | 9  10 |
| 4. | Chapter 2: Methodology   * 2.1 Components   + 2.1.1 Cube   + 2.1.2 Snake   + 2.1.3 Spot   + 2.1.4 Grid * 2.2 Achieved Roadmap | 14  15 |
| 5. | Chapter 3: Result and Conclusions   * 3.1 Result * 3.2 Conclusion * 3.3 Challenges * 3.4 Scope and Future Work | 16 |
| 6. | Chapter 4: References | 17 |
| 7. | Chapter 5: GitHub | 17 |

**Chapter1. INTRODUCTION**

**1.1 Introduction to SNEK**

Snake is the common name for a video game concept where the player maneuvers a line which grows in length, with the line itself being a primary obstacle. The concept originated in the 1976 arcade game Blockade, and the ease of implementing Snake has led to hundreds of versions (some of which have the word snake or worm in the title) for many platforms. After a variant was preloaded on Nokia mobile phones in 1998, there was a resurgence of interest in the snake concept as it found a larger audience. There are over 300 Snake-like games for iOS alone.

We aimed to remaster The Classic Snake game in a modernized fashion using Python. Objective of the game will be to get the entire screen covered with the snake without any intersection with its own body in the fastest possible time.

Mainly, there will be three things --

1) A snake which can move Up, Left, Down or Right in the constrained area.

2) Randomly appearing collectable items which will affect the size of the snake.

3) Randomly appearing obstacles.

**1.3 Modes & Functionalities**

There will be 4 modes: -

1)User Mode

2)CPU Mode

3)Zen Mode

4)Multiplayer Mode

**USER MODE: -** The player grows one unit for collecting each fruit. The game goes on until they intersect their own body. Data Structures will be used for constructing the snake.

**CPU MODE: -** Here we will show the user how we use Graphs and a suitable algorithm to introduce an AI to our snake for finding the shortest path between two nodes to help Computer play the game itself in the most efficient way.

**ZEN MODE: -** Endless Freeplay mode, where the snake never dies, you keep playing for as long as you wish to. The only consequence of biting your own tail is that the body gets cut off from the point of collision, and the rest keeps moving.

**MULTIPLAYER MODE** *(Under Development)***: -** This is the flagship mode of SNEK. In MM a synthetic server is been created which is shared between the players through which they can get connected to play a whole new version a game. This is a competitive strategy minded game with a whole set of new rules and techniques.

**1.3 History & Inspiration for SNEK**

The Snake design dates back to the arcade game Blockade,[3][4] developed and published by Gremlin in 1976.[5] It was cloned as Bigfoot Bonkers the same year. In 1977, Atari released two Blockade-inspired titles: the arcade game Dominos and Atari VCS game Surround.[6] Surround was one of the nine Atari VCS (later the Atari 2600) launch titles in the United States and was also sold by Sears under the name Chase. That same year, a similar game was launched for the Bally Astrocade as Checkmate.

The first known personal computer version, titled Worm, was programmed in 1978 by Peter Trefonas of the US on the TRS-80,[3] and published by CLOAD magazine in the same year. This was followed shortly afterwards with versions from the same author for the Commodore PET and Apple II. A microcomputer clone of the Hustle arcade game, itself a clone of Blockade, was written by Peter Trefonas in 1979 and published by CLOAD.[8] An authorized version of Hustle was published by Milton Bradley for the TI-99/4A in 1980.[9] In 1982's Snake for the BBC Micro, by Dave Bresnen, the snake is controlled using the left and right arrow keys relative to the direction it is heading in. The snake increases in speed as it gets longer, and there's only one life; one mistake means starting from the beginning.

Nibbler (1982) is a single-player arcade game where the snake fits tightly into a maze, and the gameplay is faster than most snake designs. Another single-player version is part of the 1982 Tron arcade game, themed with light cycles. It reinvigorated the snake concept, and many subsequent games borrowed the light cycle theme.

Starting in 1991, Nibbles was included with MS-DOS for a period of time as a QBasic sample program. In 1992, Rattler Race was released as part of the second Microsoft Entertainment Pack. It adds enemy snakes to the familiar apple-eating gameplay.

Meerca Chase is a snake game available on Neopets.

Slither.io (2016) is a popular multiplayer interpretation of Snake.

In 2017, Google released their version of the game as an easter egg, whenever the phrases "snake", "play snake", "snake game" and "snake video game" are typed.

**1.4 Pygame(The Core of Game)**

Python is the most popular programming language or nothing wrong to say that it is the next-generation programming language. In every emerging field in computer science, Python makes its presence actively. Python has vast libraries for various fields such as Machine Learning (Numpy, Pandas, Matplotlib), Artificial intelligence (Pytorch, TensorFlow), and Game development (Pygame,Pyglet).

In this tutorial, we are going to learn about game development using the Pygame (Python library)

**Pygame:-**

Pygame is a cross-platform set of Python modules which is used to create video games.

It consists of computer graphics and sound libraries designed to be used with the Python programming language.

Pygame was officially written by Pete Shinners to replace PySDL.

Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable.

**Sprite, Surf, and Rect:-**

**Sprite: -** Sprite is just a 2d object that we draw on the screen. We can use them by extending the sprite class.

**Surf: -** Surfaces are like blank sheets of paper on which we draw. Our screen object is also a Surface. They can hold images as well.

**Rects:-** Rectangular area that we define on a surface.

Note: We are using RGB format of color coding in which we can form different colors by giving Red, Blue, and Green color values in the range of 0-255. A tuple of (0, 0, 0) is the color black as there is no color present and a tuple of (255, 255, 255) is the color white.

In our code, we are extending the sprite class so that we can use surfs and rects to draw our squares. We have made a surface of dimension 25x25px and filled it will a color: (0, 200, 255)

**Pygame.init():-**

pygame.init()

screen = pygame.display.set\_mode((800, 600))

square1 = Square ()

square2 = Square ()

square3 = Square ()

square4 = Square ()

The above lines of code initialize pygame using the command pygame.init() which is necessary to use the pygame module commands. After that, we define our screen object and its dimensions in pixels. Then in the next lines, we initialize our four squares.

### **Game Loop: -**

The most important part of the game development code is the game loop. This loop continuously runs in the background until the user ends the game or the game is over. Right now, our game loop is not much to look forward to. It only tracks two basic events. We can add more events according to the game state, user inputs, and much more but the point to remember here is that the game loop must end after some condition else the user will be stuck in the game loop forever.

**Blit and Flip: –**

**Blit:-** Blit keyword is used to draw a surface on another surface. In simple words when we draw a surface, we just blit it onto some other surface.

**Flip:** It is used to update the entire screen after everything is drawn. Remember that the flip only works after drawing all the necessary surfaces otherwise, it will update nothing.

**Chapter 2. METHODOLOGY**

**2.1 Components**

The whole project is divided into 4 main files:

1. **Snek\_game**

The screen functionality happens in this file. From drawing every component on to the screen to spanning between the menus and getting input from the user.

1. **Constants**

Here we have all the data that remains constant throughout the program, such as the screen size, grid size, number of rows/columns, etc.

1. **Algorithms**

In this file we have the functions that perform different shortest pathfinding algorithms on the given Grid type object (More in detail in the next section)

1. **Elements**

This is the core of our program. Here rests all the Classes responsible for carrying out everything we see in this project. This module contains four classes:

* 1. Cube
  2. Snake
  3. Spot
  4. Grid

**2.1.1 Cube**

This class is responsible for storing data and drawing every node in the snake’s body

**2.1.2 Snake**

Here the all the data related to snake is stored and the essential methods that require the snake to move across the grid in constraints to the basic rules of the snake game.

**2.1.3 Spot**

Each box in a grid (which can be imagined as a 2x2 Matrix) is made up of type Spot. A Spot stores the information such its weight, whether if it is a wall or not, its x and y coordinate and all its neighbor Spots.

**2.1.4 Grid**

All the grid related data and results of shortest path algorithms are stored in here. In fact the algorithm functions take a Grid type object itself to carry out the operation.

**2.2 Achieved Roadmap**

We have completed every mode that we mentioned in our synopsis and gone even further by creating a new and exciting Multiplayer mode. It’s a totally new concept that you would not have seen before and will be very fun to play with when finished.

The good thing about our project is that we had created this project on Github from the very start so you can view all the progress step by step from September 2020 till November 2020 in the commits section.

**Chapter 3. RESULT AND CONCLUSIONS**

**3.1 Result**

The project SNEK was one the most challenging project for our team. What we learned was the most important part of this journey that includes intensive use of Object-Oriented Programming, Data Structures and Algorithms, acquiring knowledge of networking, Library implementation and manipulation, gained experience on using Discrete Structures, a lot of coding and team work.

**3.2 Conclusion**

A fully fledged working game that we planned and worked is now officially completed to be played and published.

**3.3 Challenges**

The challenges that were faced in production of Project SNKE includes: -

* Giving more realistic motion and smooth movements using physics.
* Making our game light so that it could run on any machine without a huge load on processing units.
* Adding various mode for our user to attract likeness of our audience.
* Working on networking part for our multiplayer mode to give more and more immersive and bug free experience to our user.

**3.4 Scope and Future Work**

* As discussed above we are planning to add player missions to our game and even trying to make more themes to our game which will open a whole new level of opportunities including different environment, various playing modes and much more
* The team is planning to work and learn Android module on python by which not only on PC but even on android we would be able to enjoy SNEK as android market has a huge audience for these kinds of games, it will be highly fortunate if we get a developer license on Google’s Play Store over which we can publish our app.

**Chapter 4. REFERENCES**

* <https://www.pygame.org/wiki/tutorials>
* <https://www.javatpoint.com/pygame>
* <https://www.tutorialspoint.com/python/python_classes_objects.htm>

**Chapter 5. GITHUB**

<https://github.com/Kunal-Chauhan/SNEK>

**BY:**

**KUNAL CHAUHAN 2K19/IT/069**

**IKROOP SINGH KALSI 2K19/IT/061**