A Project Report

On

# The Capstone of Education

# A Reflective Clouser

# TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

From childhood through our journey to graduation, our education system often prioritizes conformity over individuality. The project developed with OpenGL and in Visual Studio 2022 highlights the significance of having an educational system that places a strong emphasis on originality, research, and critical thinking. It calls for clarity from the traditional memorization-based approach and toward a more equitable, comprehensive, and tolerant system that encourages students to understand, question, and create. It's a message to break free from the symbolic chains that might restrict our potential and encourage us to follow orders without question.

## 1.1 Problem Statement

We have divided the project into six chapters which includes our educational journey from childhood to graduation and problems arise through the journey to achieve our goals.

Kindergarten and primary levels are the foundation of academic activities of students and then they enter the secondary to higher secondary phase where they choose their interest of whether they want to take science, commerce or arts. Then we proceed to the graduation from universities with some traditional majors like medicine or engineering which are often unwanted by students.

## 1.2 Aims and Objective

The primary objectives of this project are to address the widespread problems found in our educational process, where students are frequently conditioned to adhere to strict procedures and memorize material, with little support given to their own ideas. The project intends to close the achievement gap in education, promote fair access to opportunities, facilitate a smooth transition from school to the workforce, and increase public understanding of the value of creativity and critical thinking in both the classroom and the workplace. By achieving these goals, the project hopes to transform education, freeing students from the mentioned limitations and enabling them to take charge, explore, and create in a world that is constantly changing.

## 1.2 System Requirements

### a. Programming Language

**C++**: We have used C++ as our primary programming language to define the graphics, interaction and behavior of the code.

### b. Operating System

**Windows 10 or 11**: Windows 10 and 11 both is chosen as the operating system to run the project effectively. These windows are suitable for projects specific features and system integration which is also beneficial to collect APIs for user interaction.

### c. IDE:

**Visual Studio 2022**: Visual Studio 2022 was chosen for its strong C++ support, integrated Windows development tools, and a rich system of resources, ensuring an efficient and compatible environment for our graphics project.

### d. Graphics Library

**GL/glut.h**: In this graphics project, window management, user input, and event handling are performed by the graphics library GL/glut.h, which is frequently connected to the GLUT (Graphics Library Utility Toolkit) framework. It makes a lot of things easier, like managing the OpenGL rendering context, handling input events, and creating windows.

### e. OpenGL 2.0

The OpenGL standard version on which the project is built is OpenGL 2.0. It offers all the features required for rendering graphics, such as support for different rendering techniques, texture mapping, and shaders.

# CHAPTER 2

# SYSTEM DESIGN

System design is the process of creating a comprehensive plan for a project's architecture and functionality. It involves defining the system's structure, components, data flow, interfaces, and communication methods. This phase also addresses security, scalability, error handling, and testing strategies.

System design for a computer graphics project refers to the process of planning and specifying the architecture, components, and functionalities of a software system or application that generates and manipulates visual images and animations. This involves determining the rendering techniques, graphics libraries, data structures for representing images, and user interface elements, as well as how these components interact. It also includes considerations for optimizing rendering performance, managing graphical assets, handling user input, and ensuring the overall visual quality and responsiveness of the graphics system. System design in computer graphics projects plays a critical role in creating engaging and visually appealing applications, such as video games, simulation software, graphic design tools, or scientific visualization programs.

## 2.1 Flowchart

A flowchart is a visual representation or diagram depicting a sequential series of operations, procedures, or tasks. It serves as a potent instrument for visually illustrating the arrangement and progression of a system, procedure, or algorithm. Flowcharts are flexible tools for representing and conveying intricate processes, rendering them more comprehensible and controllable. They hold a pivotal position in the assessment of processes, addressing challenges, and guiding decision-making in various fields and sectors.

Here we've created a flowchart that systematically dissects the project, dividing it into chapters that address distinct topics and issues.

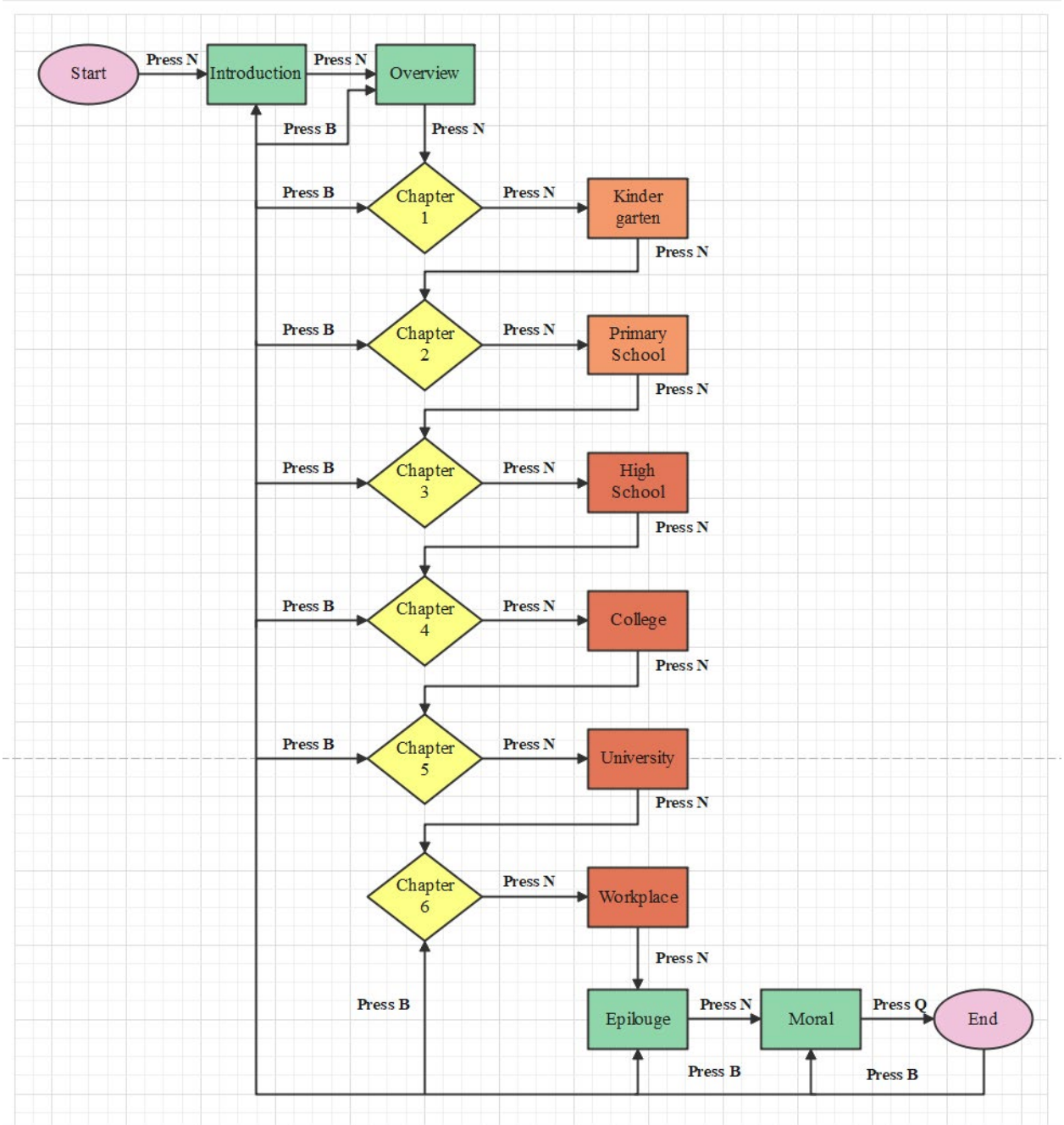


**Figure 1 Flow chart**

## 2.2 Workflow

When launching the project, an **introductory page** displaying our project's title will precede the start of each chapter. To advance to the next page, users will need to press the '**N**' key. Following that, a page providing an **overview** of the project's topic will follow. Subsequent pages dedicated to individual chapters can be accessed by pressing '**N**' repeatedly."

### Chapter – 1: Kindergarten

Kindergarten is the first phase of our educational life, from here our educational journey begin.

### Chapter – 2: Primary School

In primary school, children often face persistent academic demands, with parents requiring them to finish their homework before allowing outdoor playtime. While youngsters at this stage yearn to engage in play and socialize with friends, the reality often falls short of these desires.

### Chapter – 3: High School

During high school, a significant portion of our time is devoted to being out of the house from morning till night. Our routine typically involves attending school in the morning, going for additional tutoring in the afternoon, and returning home in the evening. This is how we navigate our way through the high school years, a pivotal and lengthy phase of our lives.

### Chapter – 4: College

In college, students typically have three primary academic paths to choose from: Science, Commerce, and Arts. A significant number of students opt for the Science major, often influenced by parental expectations that this path offers the best opportunities for success in life. While Commerce and Arts are also viable choices, they are typically less popular among students.

### Chapter – 5: University

University life often deviates from initial plans for many students. A significant portion of students find themselves pursuing fields they had not originally intended, such as some ended up with engineering, some ended up with study in medical studies. In most cases, students spend four years of their university journey and complete their graduation under the pressure of completing degrees they hadn't initially aspired to pursue.

### Chapter – 6: Workplace

After completing university education, individuals typically enter the workforce in alignment with their chosen career paths. In the end, a significant number of engineers, regardless of their specific branches, find themselves predominantly working in front of computer screens. Job satisfaction often remains elusive due to limited opportunities for personal innovation, given the hierarchical nature of many workplaces. This frustration sometimes leads to contemplation about whether this is the lifelong path they truly desire.

Upon pressing '**N**' again, the user will be directed to an **epilogue page**, and with another '**N**' press, they will access the page that outlines the **moral of our project**. To exit the project, the user should press '**Q**.' If the user wishes to navigate back, they can do so by pressing '**B**.' Pressing 'B' allows the user to return to previous pages.

# CHAPTER 3

# METHODOLOGY

A computer graphics project methodology is a systematic process for planning, creating, and executing a graphics-based project. It includes stages such as definition, design, development, integration, testing, quality assurance, documentation, deployment, post-deployment support, review, evaluation, and closure. The methodology ensures efficient and effective project execution, meeting goals and producing excellent outcomes. It involves setting goals, creating a schedule, evaluating risks, and developing design plans.

There exist various computer graphics algorithms, and for our proposed project, we have implemented a selection of these algorithms. The specific algorithms we employed are listed below:

## 3.1 DDA Algorithm:

The Digital Differential Analyzer algorithm is often used for scan converting lines due to its simplicity, conceptual clarity, and ability to generate continuous lines. It is often introduced as a fundamental line-drawing algorithm for educational purposes and can be customized to fit specific requirements. The choice of line-drawing algorithm depends on project requirements, performance considerations, and code simplicity.

## 3.2 Bresenham's line algorithm:

Bresenham's line algorithm is efficient, uses integer arithmetic, and uses fixed steps for color selection. It is predictive, predicting pixel placement based on line direction and slope, enhancing performance. Bresenham's algorithm guarantees accurate pixel placement, ensuring high-quality images. It can draw lines at various angles, including vertical and horizontal lines, and is widely supported and well-understood.

## 3.3 Midpoint Circle Algorithm:

The Midpoint Circle Algorithm is a popular choice for drawing circles in OpenGL projects and computer graphics applications. Its efficiency, predictive nature, and accuracy make it a valuable tool for real-time graphics applications. The algorithm uses integer arithmetic, making it faster and more suitable for hardware acceleration. It can be customized to fit specific requirements, such as anti-aliasing or pattern filling.

## 3.4 Flood Fill Algorithm:

The Flood Fill algorithm is a crucial tool in OpenGL projects and computer graphics applications for filling enclosed regions with a specific color. It is used for polygon filling, area highlighting, image processing, brush and bucket tools, interactive graphics, selection and masking, and terrain generation. The algorithm is particularly useful for areas enclosed by lines

or contours and for applying consistent colors or patterns. However, implementing and handling edge cases can be challenging, especially in interactive real-time applications.

### 3.5 Boundary Fill Algorithm:

For filling closed regions with particular colors or patterns in OpenGL projects and computer graphics applications, the Boundary Fill algorithm is an essential tool. Polygon filling, area highlighting, interactive graphics, masking, creating terrain, art and design, and quality assurance and inspection are among its uses. When applying consistent colors or patterns, the algorithm is especially helpful for areas that are surrounded by lines or contours. But addressing and implementing edge cases can be difficult, particularly in real-time interactive applications.

### 3.6 Translation:

For a variety of applications, including user interface design, 2D games, HUD, data visualization, animation, texture mapping, orthographic projections, GUI development, and particle effects, 2D transformations are essential in OpenGL projects and computer graphics software. They make it possible to integrate particles, perform orthographic projections, and precisely position, animate, and integrate 2D objects into 3D environments.

### 3.7 Scaling:

2D scaling is a crucial tool in OpenGL projects for modifying 2D elements' dimensions, size modification, UI elements, zooming, animation, textures, responsive design, collision detection, 3D environments, data visualization, and special effects. It uses matrix transformations to apply a scaling matrix to vertices, ensuring flexible UIs and responsive data visualization.

### 3.8 Cohen Sutherland Line Clipping Algorithm:

Cohen Sutherland's "line clipping" technique is used in OpenGL projects and computer graphics to trim lines beyond a viewport or window, reducing computational load and improving performance. The algorithm uses quick tests to identify lines outside the viewport, resulting in instant rejection. It can also clip lines indicating UI elements' borders. The Cohen-Sutherland algorithm is beneficial for cross-platform development due to its compatibility with various graphics libraries and APIs.

### 3.9 Sutherland Hodgeman line clipping algorithm:

The Sutherland-Hodgeman line clipping algorithm is a versatile tool used in OpenGL projects and computer graphics for precise line clipping against polygonal windows or viewports. It can be used for complex boundary clipping, masking effects, custom views, polygon clipping, and user interface clipping. It's also used in games and interactive applications for image processing and special effects.

# CHAPTER 4

# IMPLEMENTATION AND RESULT

The fundamental steps in starting this project were defining the structure, initializing variables, and configuring the OpenGL and GLUT programming environment for a graphics or multimedia project.

## 4.1 Variable Declaration

Variables are used in graphics or multimedia projects to manage animations and transitions, controlling the visibility of elements like titles, subtitles, paragraphs, and chapters. These variables are often used in combination with OpenGL to smoothly transition and fade elements. Adjusting the values of these variables allows for a dynamic and visually appealing user experience.

### a. Fading in a Ball Element:

We have used this variable to fade the balls gradually. We have set kg_ball_fade to 0 initially (completely transparent) and then increase its value over time using interpolation or animation logic. As kg_ball_fade increases, the ball becomes less transparent, gradually becoming fully visible.

### b. Moving a Kid Character:

These variables can be used to move kid characters horizontally. By modifying trans_x_kid1, we can make a character move from left to right on the screen.

### c. Sun Position:

This variable determines the position of the sun. For example, increasing sun_move_left might move the sun from right to left, simulating a day-to-night transition.

### d. Schoolboy's Position:

The schoolboy character's position is determined by these variables. The schoolboy character can be moved to precise screen coordinates by modifying schoolboy_x and schoolboy_y.

### e. Moving Characters in the Scene:

Character movements within a particular scene is managed with the help of these variables. For example, we can move a character to the right by adjusting trans_x_ec_sb1 and change the vertical position by adjusting trans_y_ec_sb.Color

**f. Morphers:**

These variables control the color of various elements in the project. For instance, window_top_r, window_top_g, and window_top_b could control the top part of a window's color. By adjusting these values, we create color morphing effects, such as changing the sky color from day to night or the appearance of different environmental scenes.

**g. Turn-based Switching:**

These boolean variables likely determine the state or progression of your project. For example, sun_moved_half might indicate whether the sun has moved halfway across the screen, triggering a change in the scene. chapter_1_done, chapter_2_done, etc., could mark the completion of each chapter, influencing the narrative flow.

**h. Random Star Generation:**

The characteristics of the stars in the project are determined by the variables associated with star formation. The number of stars that are visible is determined by no_of_stars, whereas star_alpha may regulate the transparency of stars. The coordinates of the stars on the screen are probably stored in the stars_array.

### 4.2 Function Declaration

**a. drawWoman:**

The drawWoman function is a versatile tool for creating female characters in graphic or multimedia projects. It allows for precise translation and scaling parameters, allowing for resizing and positioning of the character. The function constructs various female elements, including neck, facial features, hair, eyes, smile, and nose, and customizes their colors and dimensions. It also defines the woman's arms, upper body, skirt, and ribbon, allowing for distinctive clothing styles.

**b. drawKidsShirtAndTrousers:**

To create a kid character with customizable shirt and trouser colors, the code offers two functions: `drawKidsShirtAndTrousers` and `drawKid`. The `drawKid` function defines the RGB color of the shirt and captures the child's facial features. The kid's shirt and pants are drawn using the RGB color parameters by the `drawKidsShirtAndTrousers` function. In multimedia projects, these features improve interactivity and visual storytelling.

**c. drawTree:**

The `drawTree` function is a declared for creating tree-like graphics or multimedia projects. It allows for precise placement and dimensions using translation parameters and scaling parameters. The tree consists of three parts: bark, crown, and apples. The bark is realistically rendered in brown, the crown is green, and the apples are positioned around the crown. These

parameters allow for the creation of vibrant tree landscapes, enhancing visual storytelling and multimedia content engagement.

We have employed various additional functions to draw essential elements like the door, door knob, blackboard, table, ceiling, wall, floor, and more within our project. These functions ensures that story, instructional material, and interactive experiences are effectively communicated to the audience while also improving visual appeal and interactivity.

### 4.3 Main Function

The main function is the entry point for the graphics or multimedia project, initializing the OpenGL environment and orchestrating key components. It sets up the window, configures rendering settings, and associates the display function. User interactions are handled through keyboard and mouse input functions. The program's initial state is established through the `initializeScreen` function, and a timer function, `update`, ensures dynamic updates. The main loop, powered by `glutMainLoop`, maintains responsiveness and user engagement.

### 4.4 User Interaction

These key commands enable users to easily navigate through the content, moving forward or backward between pages, and provide a convenient way to exit the application when they are done.

Using Alphabetical Key:

**‘N’ or ‘n’ for Moving to the next page:**

When a user hits the 'N' or 'n' key, the project changes to the next page, allowing for sequential exploration as they move from one page to the next.

**‘B’ or ‘b for Moving to the previous page:**

The user can access the previous page of their project by pressing 'B' or 'b', ensuring a seamless and intuitive experience for reviewing earlier material.

**‘Q’ or ‘q’ for Quitting:**

If a user wishes to exit or close the application, they can do so by pressing the 'Q' or 'q' key. This function provides an efficient way to terminate the project and return to the main menu or desktop.

### 4.5 Output

Project output is the final measurable result achieved after completing all planned tasks and activities. The output of our project is described here with the snapshots.

**Introduction Page:**

The initial section of our project provides an introduction to our topic and presents information about the team members.

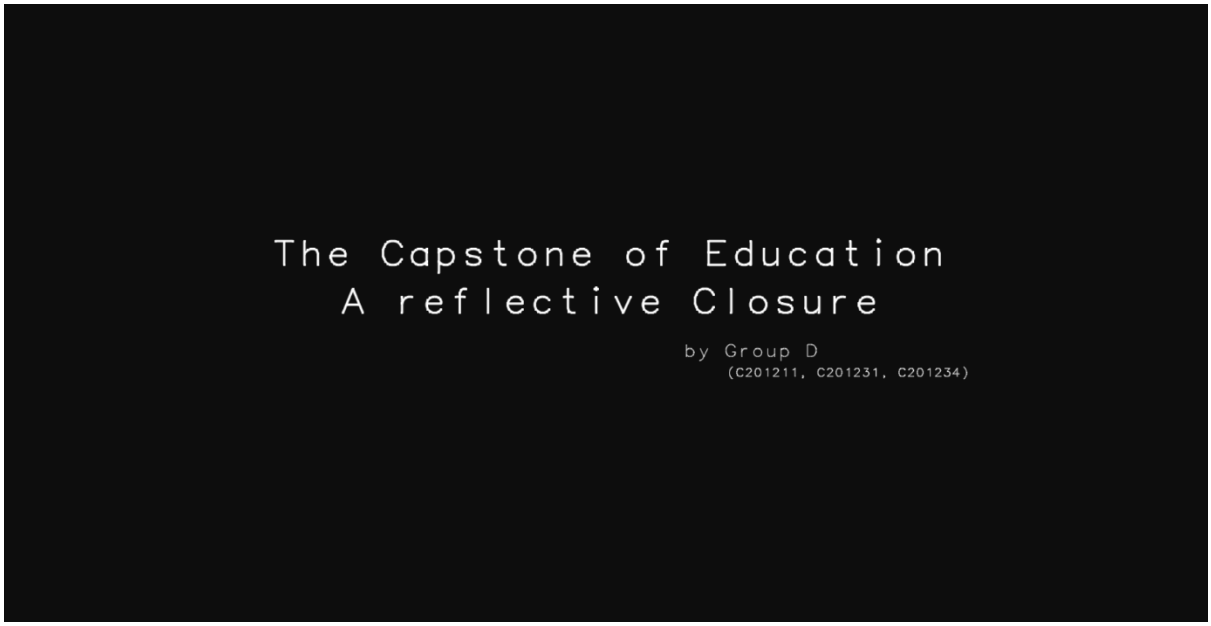


**Figure 2: The Introduction Page**

**Overview Page:**

This page offers an overview of our project's topic.

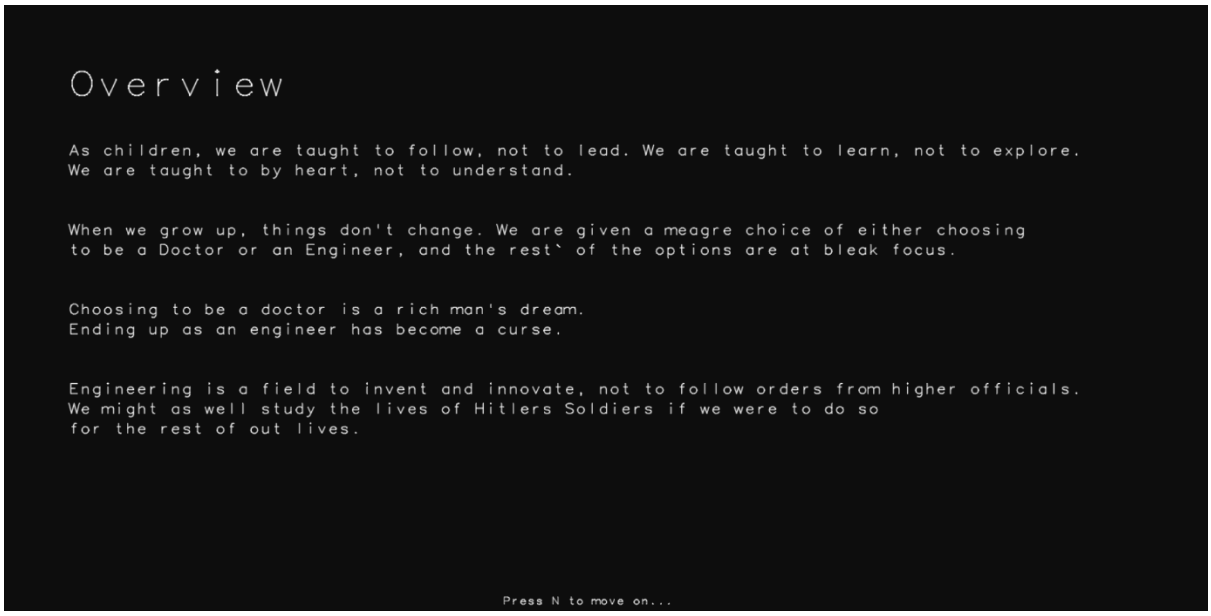


**Figure 3: Overview Page**

**Chapter 1 Screen:**

Our project's kindergarten phase is covered in Chapter 1.


**Figure 4: Kindergarten Intro Page**

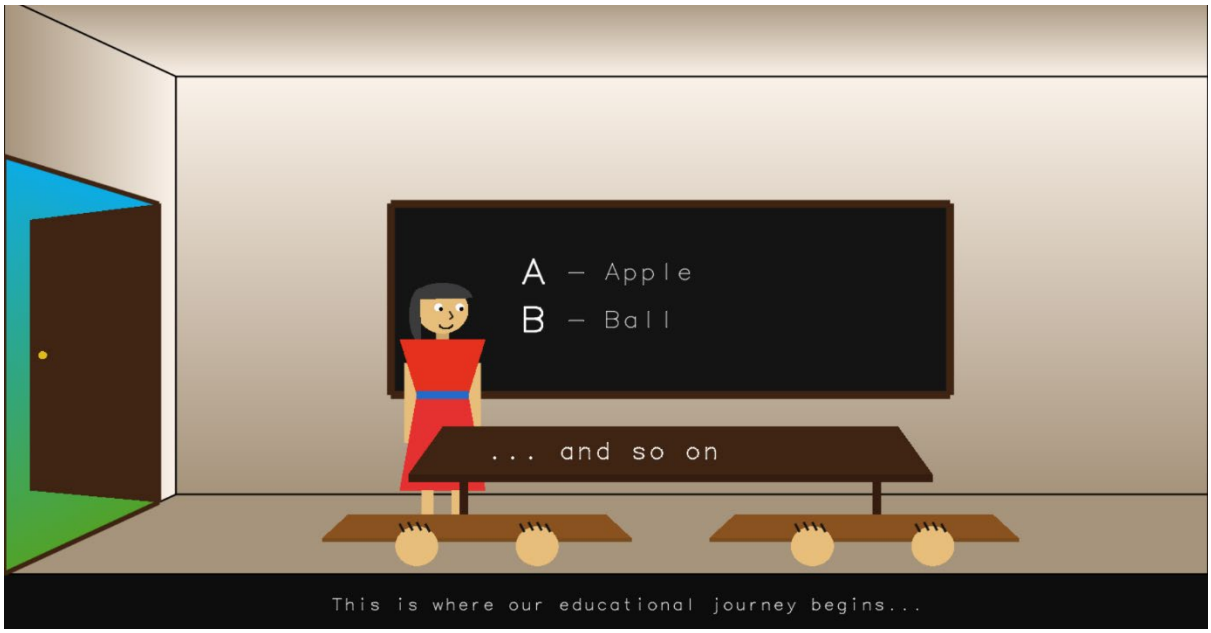


**Figure 5: Kindergarten Screen**

**Chapter 2 Screen:**

Primary School phase is covered in Chapter 2

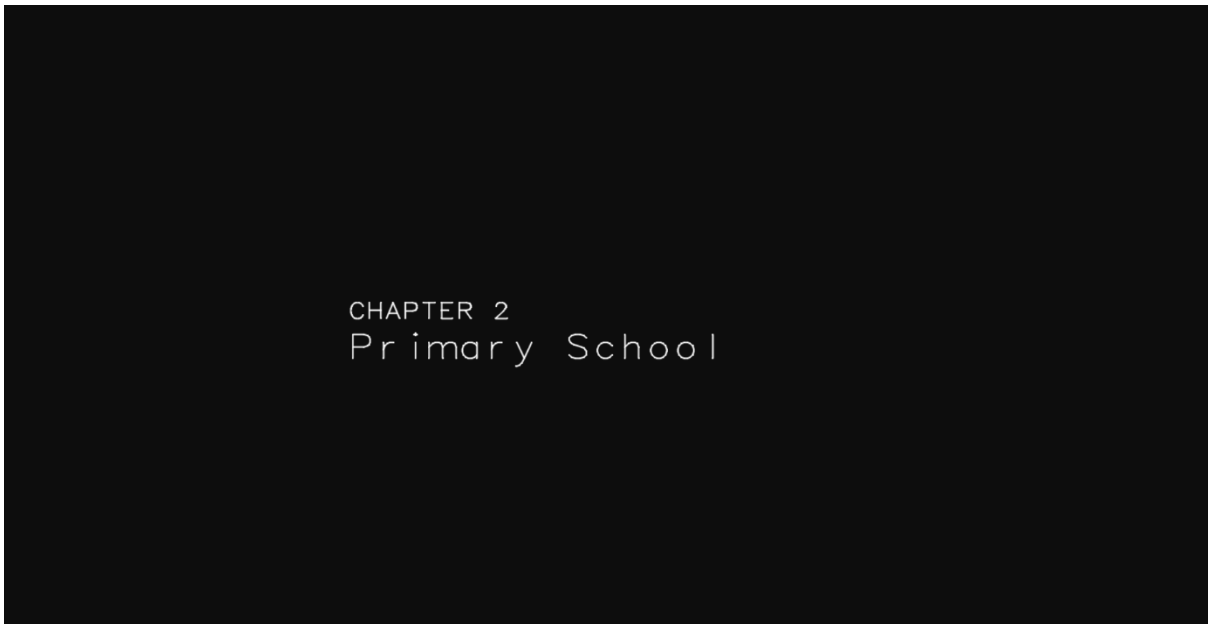


**Figure 6: Primary School Intro Page**

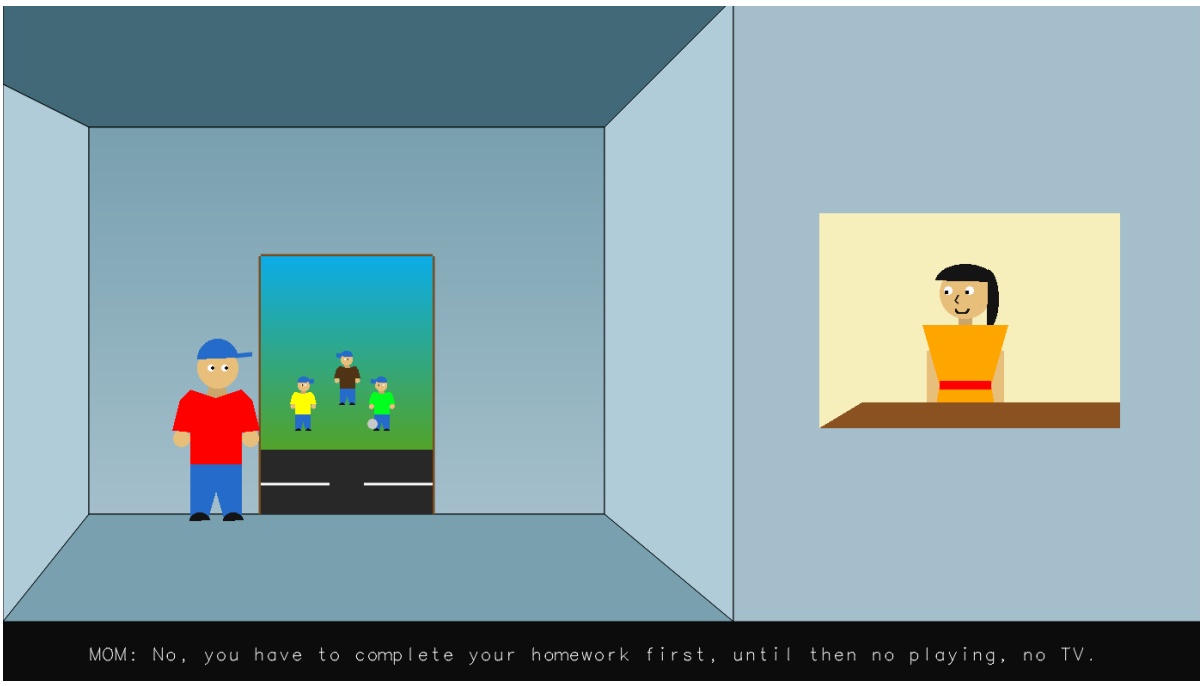


**Figure 7: Primary School Screen**

**Chapter 3 Screen:**

High School phase is covered in Chapter 3

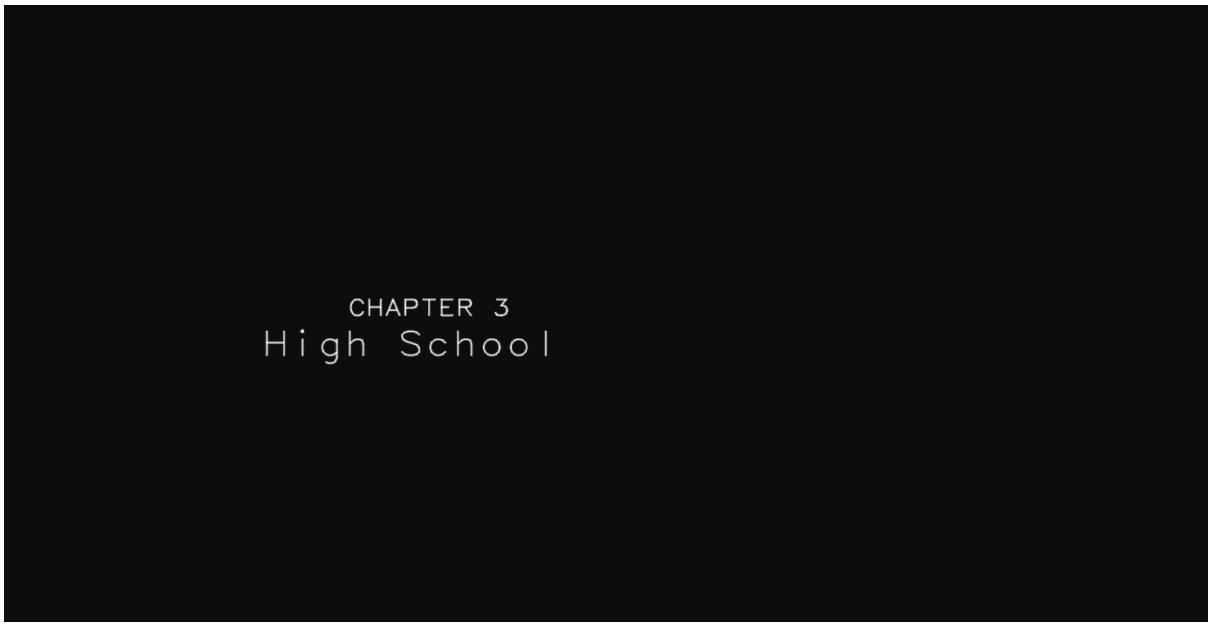


**Figure 8: High School Intro Page**

In High School, we usually follow this schedule: we go to school in the morning, get extra tutoring in the afternoon, and then come home in the evening.


**Figure 9: School in the morning**


**Figure 10: Going Tuition from School in the afternoon**


**Figure 11: Spending time at tuition till evening.**

**Chapter 4 Screen:**

College phase is covered in Chapter 4


**Figure 12: College Intro Page**

Some students select Science in college, while others select Commerce, and so on.

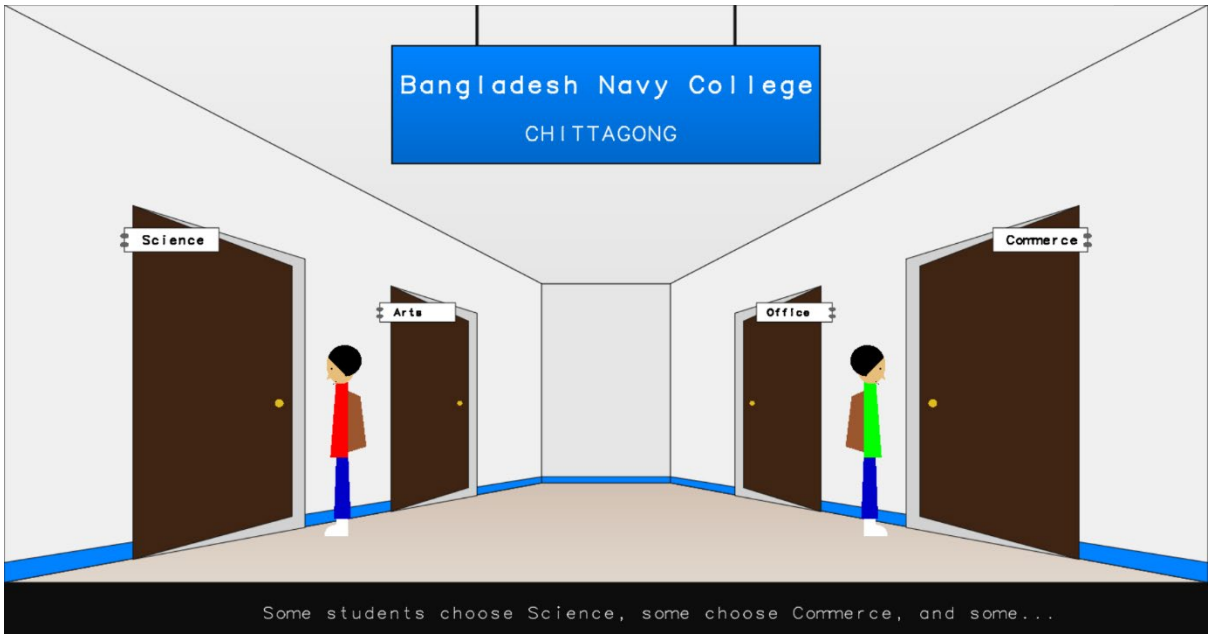


**Figure 13: College Screen**

**Chapter 5 Screen:**

University phase is covered in Chapter 5

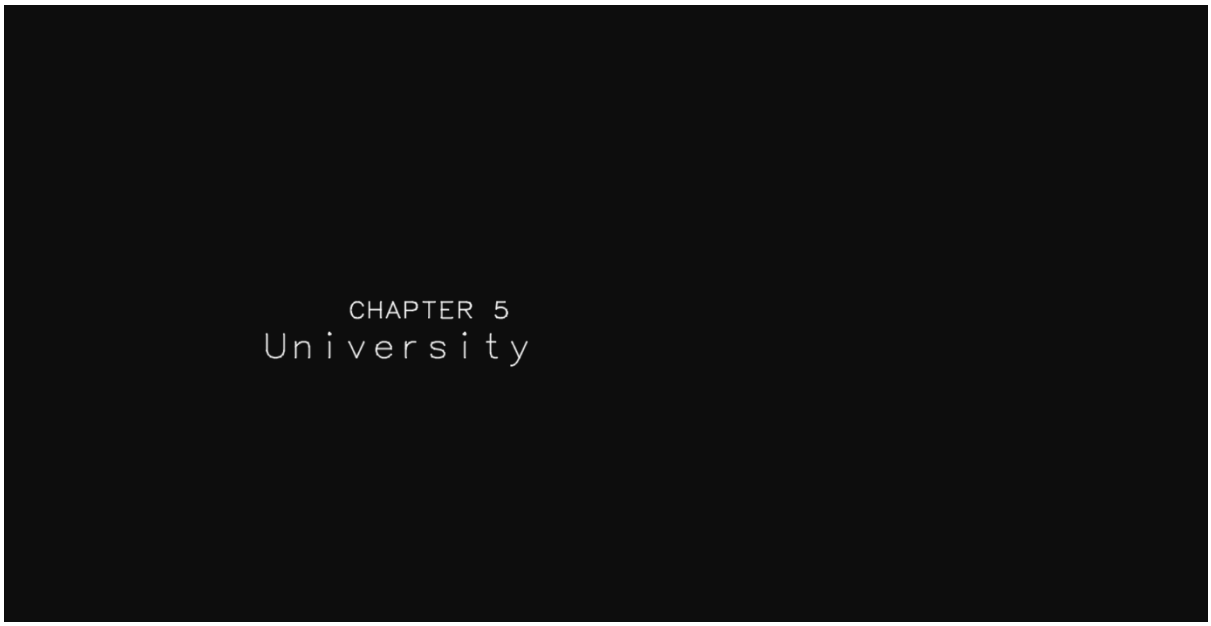


**Figure 14: University Intro Page**

In university, we select our degrees based on the fields we previously selected to study. But some student ended up in Engineering, irrespective of what path they took in the past.

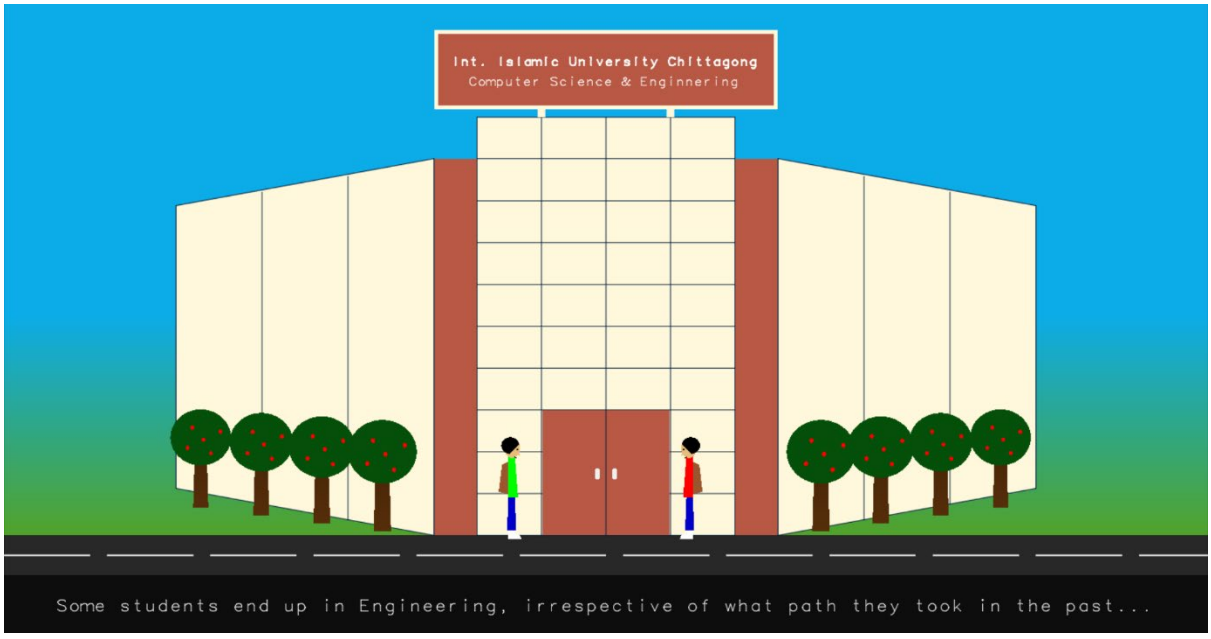


**Figure 15: University First year**

Completing the degree that a student has choose to study whether he wanted it or not.

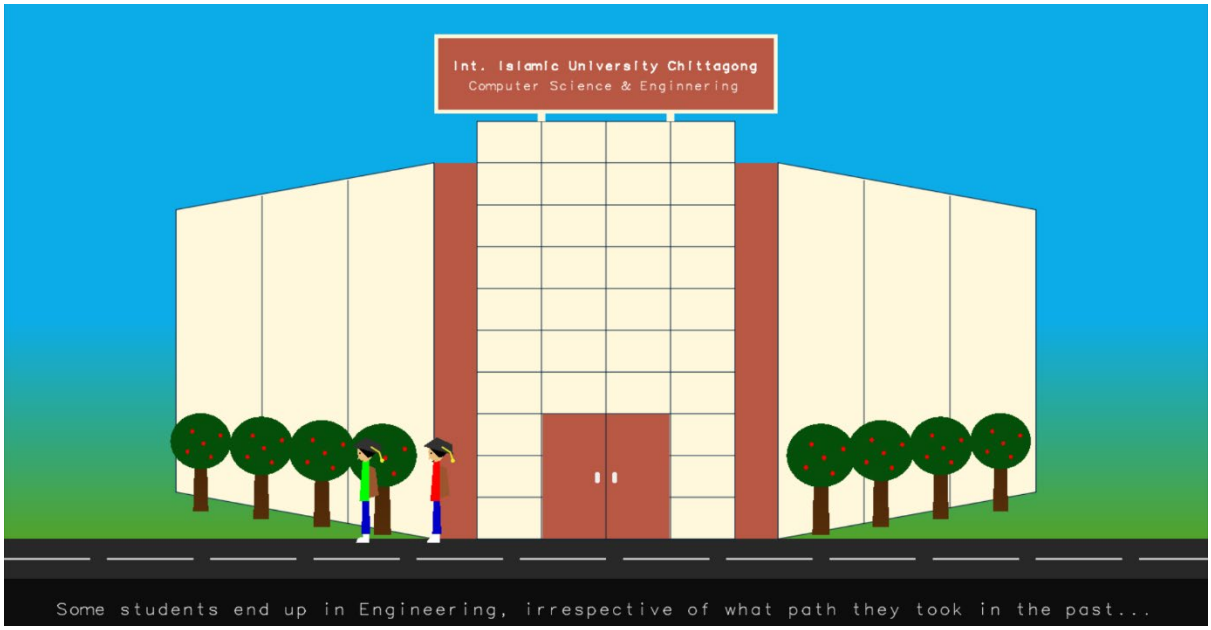


**Figure 16: Completed graduation**

**Chapter 6 Screen:**

Workplace is covered in Chapter 6

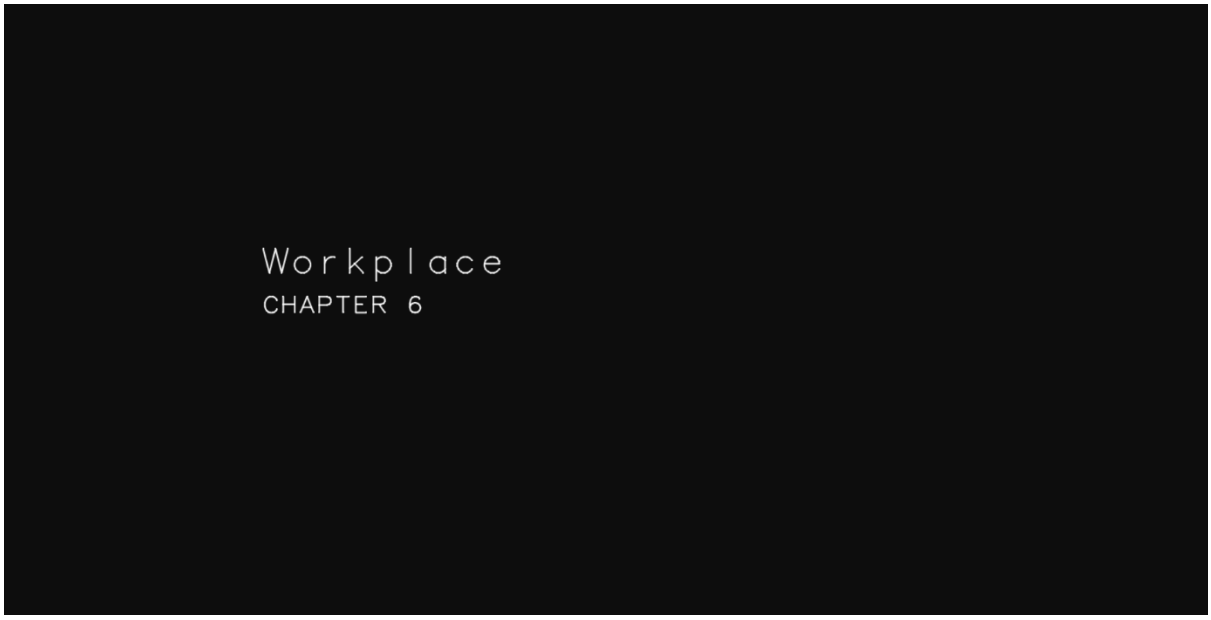


**Figure 17: Workplace Intro Page**

In the end, a significant number of engineers, regardless of their specific branches, find themselves predominantly working in front of computer screens. This frustration sometimes leads to contemplation about whether this is the lifelong path they truly desire.


**Figure 18: Workplace Screen**

**Epilogue Page:**

This page contains the epilogue of our topic.

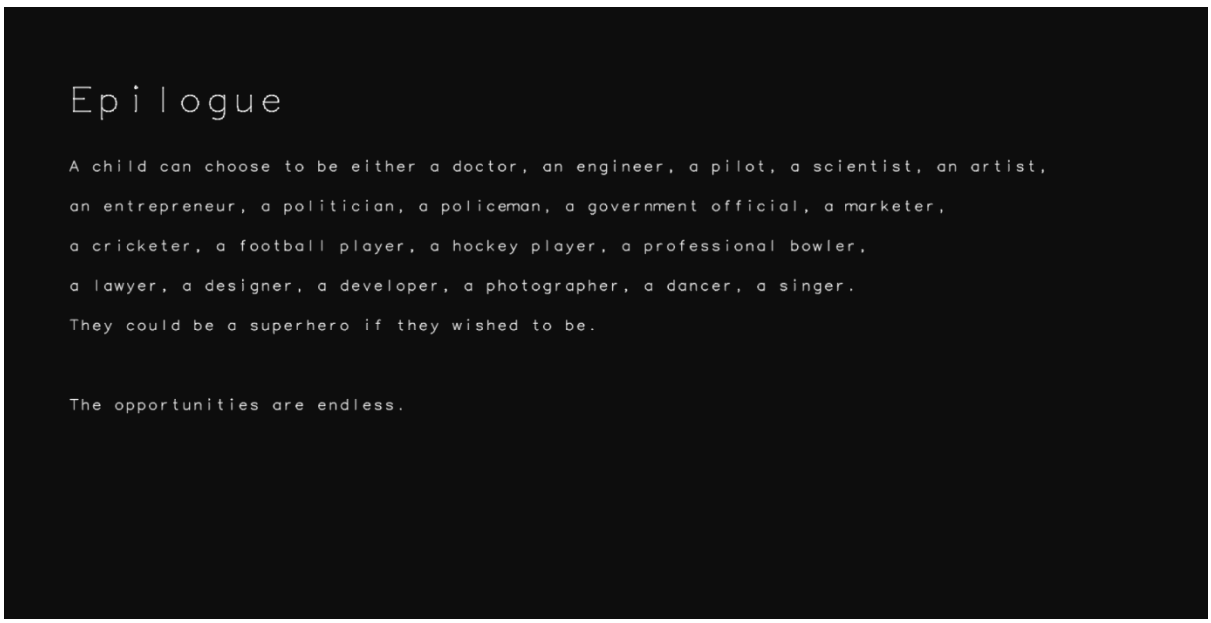


**Figure 19: Epilogue Page**

**Moral Page:**

This page shows us the moral of the story

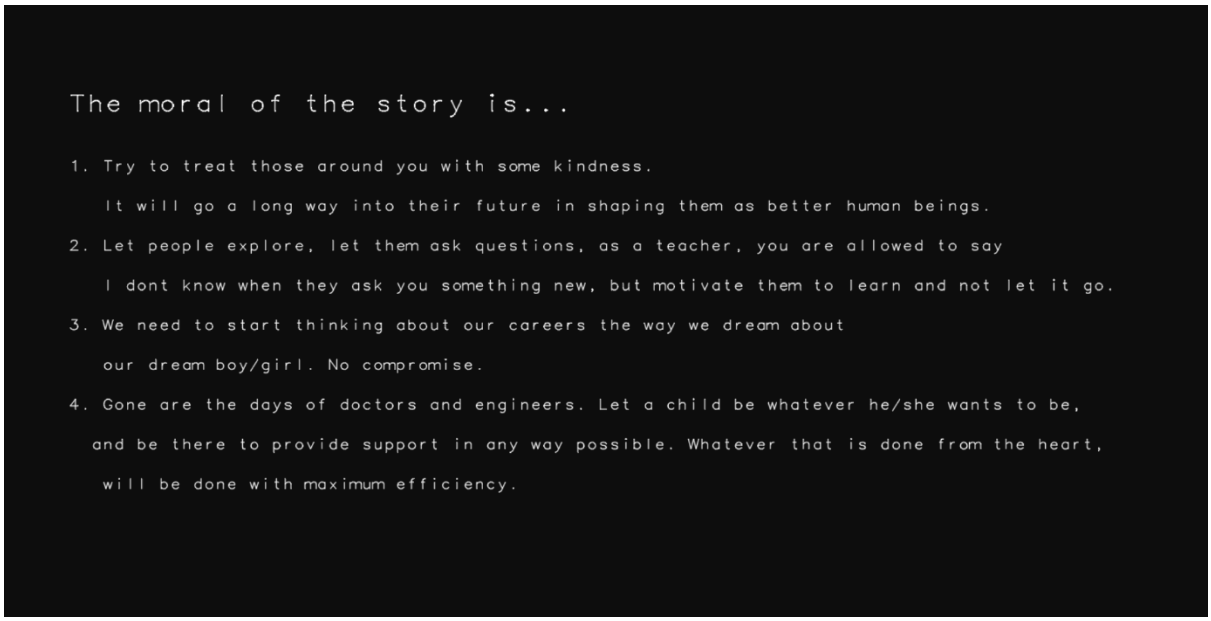


**Figure 20: Moral Page**

# CHAPTER 5

# CONCLUSION

The project advocates for a fundamental transformation in the educational system, focusing on fostering originality, research, and questioning and creating. It seeks to break free from symbolic chains that constrain potential and challenge us to accept orders without question. The project aims to close the achievement gap, ensure equal access to opportunities, and create a seamless transition from the classroom to the workforce. It also seeks to elevate the understanding of creativity and critical thinking in both educational and professional settings. The ultimate goal is to pave the way for a reimagined education system that liberates students from conformity and empowers them to take charge of their learning journey.

## 5.1 Advantages

- Through this project, we can raise awareness in our society about the need for educational system enhancements.
- This project encourages students to reconsider their life goals and aspirations.
- It challenges the notion that individuals from certain backgrounds are the sole candidates for careers in fields like engineering or medicine, highlighting the potential for changing such beliefs.

## 5.2 Disadvantages

- Omitting the inclusion of music or voice systems in the project might lead to a less captivating and immersive user experience, particularly when audio elements could enhance the scenarios.
- The absence of voice systems means forfeiting the opportunity for interactive and dynamic user interactions, such as voice-activated commands and responses.
- Relying exclusively on visual components could restrict the project's accessibility, as certain users may prefer or require audio cues for comprehension and engagement.
- Visual Studio 2022 is a powerful development environment, but it can be demanding on system resources. Neglecting performance optimization in the project could result in sluggish execution or potential hardware constraints on certain systems.

### 5. 3 Future Work

We see future prospects for the project's growth and evolution. Our forward-looking vision involves a dedicated commitment to adjusting and expanding in reaction to the evolving landscape of educational and technological challenges and opportunities. Several other areas of future work we plan to focus on:

- **In-depth Educational Enhancement Scenario:** We aspire to create a comprehensive and detailed scenario that outlines practical steps for improving our educational system, emphasizing originality and critical thinking.

- **Enhanced Music Integration:** We will further refine the integration of music systems into each scenario, ensuring a seamless and engaging audiovisual experience for users.

- **Compatibility Testing:** We aim to conduct extensive compatibility testing by running the project with different compilers and environments to ensure broader accessibility and reliability.

- **User Feedback Incorporation**: We plan to gather and incorporate user feedback to continuously refine and expand the project's scope based on the needs and preferences of our audience.

- **Research and Innovation:** The project will remain open to ongoing research and innovation in the field of education, technology, and user experience, with a commitment to staying current and relevant.