## 

## UNIVERSTY OF MUMBAI

**PROJECT REPORT ON**

**PATH MAKER GAME**

**SUBMITTED**

**BY**

****

## DTSS COLLEGE OF COMMERCE

### UNDER THE GUIDANCE OF

### MRS. TEJAL BHAVE

##### BSCIT SEM-IV [2023-2024]



Ref No: Date:

**Certificate**

###### This is to certify that the project entitled **PATH MAKER GAME** is under taken at the **D.T.S.S COLLEGE OF COMMERCE & SCIENCE** by

###### In partial fulfillment to B.Sc.IT degree (Semester IV) Examination had not been submitted for any other examination and does not form part of any other course under gone by the candidate. It is further certified that he has completed all required phases of project.

###### Signature of Internal Guide Signature of External

HOD/In-Charge/Co-Ordinator

**CONTENT**

## Chapter No. Title Page No.

## ABSTRACT……………………….……….……………………………01

## ACKNOWLEDGEMENT……….……….…………..…………….......02

## INTRODUCTION..……………...……….………………………...…....03

## AIM AND PURPOSE…….……………...………………….…….…….04

## PROJECT OVERVIEW….…………….………………………...…….05

## PROJECT DESCRIPTION…………….………………………...…….06

## SCOPE………………….….…………….………………………...…….07

## MODULES……………..….…………….……....………………...…….08

## INTERFACES…………….…………….……….………………..…….11

## REQUIREMENTS SPECIFICATION.….…….………………..…….12

## SYSTEM REQUIREMENTS.………………….………………..…….13

## ADVANTAGES AND DISADVANTAGES……...……………..…….14

## RESULT ANALYSIS…..….……………………………………..…….15

## CONCLUSION & FUTURE WORK…………………………..……..17

## REFERENCES…………………………………………………..……..18

**ABSTRACT**

This project aims to create a web-based Path Maker game using HTML, CSS, and JavaScript.

HTML will be utilized to structure the game layout, dividing it into distinct sections such as the game area, score display, and user interface elements. CSS will be employed to style the game elements, including the stick figure character, platforms, and background, ensuring a visually appealing presentation that remains faithful to the game's aesthetic.

JavaScript will play a crucial role in implementing the game's functionality. It will handle user interactions, such as controlling the length of the stick and timing the jumps to land on platforms. Additionally, JavaScript will manage game physics, calculating the distance the stick extends and determining whether the player successfully reaches the next platform or falls short.

The game will feature responsive design principles, allowing it to adapt seamlessly to various screen sizes and orientations, ensuring an optimal gaming experience across desktop and mobile devices. Furthermore, it will include intuitive touch or mouse controls, enabling players to interact with the game effortlessly.

Pg. 01

**ACKNOWLEDGEMENT**

In completing this project report on project title **Path Maker Game**. I had to take the help and guideline of a few respected people, who deserve my greatest gratitude.

The completion of this project report gives me much Pleasure. I would like to show my gratitude to **MRS. TEJAL BHAVE** for giving me a good guideline for project throughout numerous consultations. I would also like to expand my deepest gratitude to all those who have directly and indirectly guided me in writing this project report.

Many people, especially my classmates and friends themselves, have made valuable comments and suggestions on this proposal which gave me inspiration to improve my project. Here I thank all the people for their help directly and indirectly to complete this project report.

Pg. 02

**INTRODUCTION**

Path Maker game, has captivated players with its simple yet addictive gameplay. In this game, players control a stick figure character who must navigate through a series of platforms by extending a stick to bridge the gaps. The challenge lies in accurately timing the stick extension and releasing it to land safely on the next platform. As players progress, the platforms become increasingly distant, testing their reflexes and precision.

This project aims to bring the thrill of Path Maker to the web by developing a browser-based version using HTML, CSS, and JavaScript. The game will provide an engaging experience for players. Through the use of HTML for structure, CSS for styling, and JavaScript for interactivity, the game will offer an intuitive and visually appealing interface.

In this document, we will explore the design and implementation of Path Maker, detailing the structure of the HTML markup, the styling applied with CSS to create an engaging visual environment, and the functionality implemented using JavaScript to enable user interactions and game mechanics. This project promises to deliver an exciting and enjoyable gaming experience right in your browser.

Pg. 03

**AIM AND PURPOSE**

**Aim:**

The aim of this project is to create a web-based version of Path Maker game using HTML, CSS and JavaScript. The project providing players with an accessible and engaging gaming experience directly within their web browsers.

**Purpose:**

The purpose of this project is to showcase the capabilities of HTML, CSS, and JavaScript for developing interactive web-based games by creating a browser-based version of Path Maker. Through this project, developers can demonstrate the potential of web technologies for delivering immersive and engaging gaming experiences while providing players with a fun and accessible way to enjoy the game.

Pg. 04

**PROJECT OVERVIEW**

The Path Maker Game project will consist of several key components:

**1. Game Layout and Structure:** The game will be structured using HTML to define the layout of various elements such as the game area, score display, and user interface components.

**2. Visual Design and Styling:** CSS will be utilized to style the game elements, including the stick figure character, platforms, background, and user interface elements. The visual design will aim to provide a familiar experience for players.

**3. Game Mechanics and Interactivity:** JavaScript will handle the game's functionality and interactivity. This includes controlling the stick length, timing the stick extension and retraction, calculating the distance to the next platform, and determining whether the player successfully lands or falls short. JavaScript will also manage the game's physics to simulate realistic stick movement and platform interactions.

**4. Responsive Design:** The game will be designed with responsive principles in mind, ensuring compatibility and optimal gameplay experience across various screen sizes and device orientations. This will allow players to enjoy the game seamlessly on desktop computers, laptops, tablets, and smartphones.

**5. User Controls:** Intuitive mouse controls will be implemented to allow players to interact with the game effortlessly. Players will be able to control the stick length and timing of jumps using mouse clicks.

**6. Scoring and Progression:** The game will feature scoring mechanisms to track the player's progress and performance. Players will earn points based on the distance traveled and the successful completion of jumps.

Pg. 05

**PROJECT DESCRIPTION**

The Path Maker web-based game recreates the engaging game using HTML, CSS, and JavaScript. Players control a stick figure character who must navigate through a series of platforms by extending a stick to bridge the gaps. The primary goal is to reach as far as possible without falling short or extending the stick too far.

* **Key Features**

**1. Dynamic Platform Generation:** Platforms are automatically created at random intervals horizontally across the screen, providing a constantly changing environment for the player to navigate.

**2. Intuitive Path Creation Controls:** Players can easily create paths by tapping or clicking and holding the screen, with the path extending from the character's starting position to the current cursor position.

**3. Scoring System Based on Distance Traveled:** The player's score increases as they travel further through the level, encouraging them to aim for longer paths and higher scores.

**4. Responsive Design for Compatibility Across Devices:** The game is designed to adapt its layout and controls dynamically based on the user's device, ensuring a consistent and enjoyable gameplay experience across desktop computers, laptops, tablets, and smartphones.

**5. Visual and Audio Feedback for Immersive Gameplay:** Visual animations and sound effects provide feedback on successful path creation and platform navigation, enhancing the player's immersion and engagement in the game.

Pg. 06

**SCOPE**

This project can be considered as a starter to the Html/CSS/JavaScript based game grounds, therefore large upgradations can be done to it. In the game will offer a challenging and engaging experience accessible directly from web browsers on various devices, providing players with an enjoyable way to test their reflexes and strategic skills.

Additionally, the scope may include future enhancements such as additional game modes, customization options, and social features to further enrich the gaming experience and foster community engagement.

Pg. 07

**MODULES**

**1. HTML Structure:** Create the main HTML structure including the game canvas, score display, and any other necessary elements.

**2. CSS Styling:** Apply CSS styling to the game canvas, background, stick figure, platforms, and score display to make the game visually appealing.

**3. Game Initialization:** Initialize the game canvas and set up necessary variables such as stick length, stick position, platform position, etc.

**4. Stick Control:** Implement functionality to control the length and angle of the stick using mouse or touch events.

**5. Platform Generation:** Generate platforms dynamically with varying widths and positions to create a challenging game environment.

**6. Character Movement:** Implement logic for the stick figure to move forward, stick, and fall based on the player's input and timing.

**7. Collision Detection:** Check for collisions between the stick figure and platforms to determine whether the stick figure successfully lands on the platform or falls.

**8. Score Calculation:** Calculate the score based on the distance covered by the stick figure and display it on the screen.

**9. Game Over Handling:** Handle game over scenarios such as when the stick figure falls off the platform or fails to land properly.

**10. Restart Functionality:** Implement functionality to restart the game after it ends, allowing the player to play again.

Pg. 08

**11. Responsive Design:** Ensure the game layout adjusts appropriately for different screen sizes and orientations to provide a consistent experience across devices.

**12. Optimization and Testing:** Optimize the game code for performance and test it thoroughly to ensure smooth gameplay and functionality across different browsers and devices.

Pg. 09

**INTERFACES**

* **Hardware Interface:**
* **Input Devices:**
* Mouse: Used for controlling the length and angle of the stick.
* Touchscreen (for mobile devices): Similar to mouse input, used for controlling the stick length and angle by tapping and swiping.
* **Display Device:**
* Monitor or screen: Displays the game environment, including the stick figure, platforms, and score.
* **Software Interface:**
* **Web Browser:** The game runs within a web browser environment, utilizing HTML, CSS, and JavaScript.
* **JavaScript Libraries:** Libraries such as jQuery or Phaser may be used to simplify certain aspects of game development, such as event handling or rendering.
* **Graphics Library (e.g., Canvas):** Utilized for drawing the game elements, including the stick figure, platforms, and background, providing a graphical interface for the game.
* **User Interface:**
* **Game Canvas:** Displays the game environment where the stick figure navigates across platforms.
* **Score Display:** Shows the player's current score, updating in real-time as the player progresses through the game.
* **Game Controls:**
* Mouse or touchscreen controls for interacting with the game, including:
* Clicking or tapping to extend the stick.
* Releasing the click or tap to place the stick and make the stick figure jump.
* **Game Over Screen:** Displays when the game ends, showing the player's final score and providing options to restart the game.

Pg. 10

**REQUIREMENTS SPECIFICATION**

* **Functional Requirements:**

**1. Stick Control:**

* Players should be able to control the length and angle of the stick using mouse input or touchscreen gestures.
* The stick should extend when the player clicks or taps and retract when the player releases the input.

**2. Character Movement:**

* The stick figure should move forward automatically at a constant speed.
* Upon releasing the input to extend the stick, the stick figure should jump and attempt to land on the platform.

**3. Platform Generation:**

* Platforms should be generated dynamically with varying widths and positions to create a challenging game environment.
* The distance between platforms should increase progressively as the player advances.

**4. Collision Detection:**

* Collision detection should be implemented to determine whether the stick figure successfully lands on the platform or falls.
* If the stick figure lands on the platform, the game should continue, and the player's score should increase.
* If the stick figure falls, the game should end, and the player's final score should be displayed.

**5. Score Calculation:**

* The score should be calculated based on the distance covered by the stick figure, with higher scores awarded for longer jumps.
* The score should be displayed in real-time during gameplay.

**6. Game Over Handling:**

* When the stick figure falls off the platform, the game should end, and the player should be notified.
* Players should have the option to restart the game after it ends.

Pg. 11

* **Non-functional Requirements:**

**1. Performance:**

* The game should run smoothly on various devices and screen sizes without significant lag or slowdown.
* Response time to player input should be minimal to provide a seamless gaming experience.

**2. Scalability:**

* The game should be scalable to accommodate potential future updates, such as additional features or levels.

**3. Compatibility:**

* The game should be compatible with modern web browsers, including Chrome, Firefox, Safari, and Edge, ensuring accessibility for a wide range of users.

**4. User Interface Design:**

* The game should have an intuitive and visually appealing user interface, with clear instructions on how to play.
* Visual elements, including the stick figure, platforms, and background, should be designed to enhance the gaming experience.

**5. Accessibility:**

* The game should be accessible to users with disabilities, including support for screen readers and keyboard navigation.

**6. Security:**

* The game should be secure against common web vulnerabilities, such as cross-site scripting (XSS) and cross-site request forgery (CSRF), to protect user data and ensure a safe browsing experience.

Pg. 12

**SYSTEM REQUIREMENTS**

* **Software Requirements:**
* **Operating System:**
* Path Maker game can be developed and run on various operating systems including such as,
* Windows, MacOS, Linux, Android, IOS.
* **Web Browser:**
* Path Maker game runs in a web browser environment. It should be compatible with modern web browsers such as:
* Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge, Opera.
* **Development Tools:**
* Text Editor or Integrated Development Environment (IDE) for coding HTML, CSS, and JavaScript such as:
* Visual Studio Code, Sublime Text, Atom, WebStorm.
* **Hardware Requirements:**
* **Processor (CPU):** Path Maker game does not have high processor requirements as it is not computationally intensive. A modern processor with decent performance is sufficient.
* **Memory (RAM):** Path Maker game typically does not require a large amount of memory. A minimum of 2GB RAM should be sufficient for smooth gameplay.
* **Graphics Processing Unit (GPU):** Path Maker game does not have demanding graphics requirements. Integrated graphics or a basic dedicated GPU should be adequate.
* **Storage:** The storage requirement for Path Maker game is minimal as it primarily consists of game assets (HTML, CSS, JavaScript files) which are small in size. A few megabytes of storage space should be enough.
* **Input Devices:**
* Mouse for controlling the game.
* Keyboard (optional) may be used for additional controls if implemented.
* **Output Device:**
* Display device (monitor or screen) to view the game graphics and interface.
* Speakers or headphones for audio output if sound effects or background music are included in the game.

Pg. 13

**ADVANTAGES AND DISADVANTAGES**

* **Advantages**
* **Addictive Gameplay:** Path Maker’s simple mechanics provide addictive gameplay that keeps players engaged.
* **Accessibility:** Being web-based, Path Maker is easily accessible across various devices with internet access.
* **Portability:** Players can enjoy Path Maker on the go, making it ideal for casual gaming during breaks or commutes.
* **Low Hardware Requirements:** Path Makerruns smoothly on a wide range of devices, including older or less powerful hardware.
* **Social Sharing:** The game's competitiveness encourages social sharing of high scores, fostering friendly competition among friends.
* **Learning Opportunity:** Developing a Path Maker provides an excellent opportunity to practice HTML, CSS, and JavaScript skills, particularly in game development and user interaction.
* **Disadvantages**
* **Repetitive Gameplay:** The gameplay can become repetitive over time due to its simple mechanics.
* **Limited Depth:** It lacks depth compared to more complex games, potentially leading to shorter play sessions.
* **Dependency on Internet:** Requires an internet connection to play, limiting accessibility in areas with poor connectivity.
* **Limited Revenue Potential:** May have limited revenue potential compared to games with more extensive monetization features.
* **Competitive Market:** Faces competition in a crowded gaming market, making it challenging to stand out.
* **Development Challenges:** Developing a polished version with smooth animations and responsive controls can be challenging, especially for novice developers.

Pg. 14

**RESULT ANALYSIS**

Pg. 15



Pg. 16



**CONCLUSION AND FUTURE WORK**

* **Conclusion**

In conclusion, the Stick Hero game clone offers simple and addictive gameplay accessible to a wide audience. However, it may suffer from repetitiveness and limited depth compared to more complex games. While it provides learning opportunities for developers, it faces challenges such as dependency on internet access and competition in the gaming market. Overall, Stick Hero clone projects can be enjoyable but may have limitations in long-term engagement and revenue potential.

* **Future Work**
* **New Levels:** Design and implement additional levels with different challenges and obstacles to keep players engaged.
* **Power-Ups:** Introduce power-ups that provide temporary boosts or advantages to the player, adding variety to the gameplay.
* **Customization Options:** Allow players to customize their stick figure character or unlock new characters with unique abilities.
* **Multiplayer Mode:** Implement a multiplayer mode where players can compete against each other in real-time, either locally or online.
* **Social Features:** Integrate social features such as leaderboards, achievements, and sharing options to encourage competition and interaction among players.
* **Expanded Platforms:** Port the game to different platforms such as mobile devices, consoles, or VR headsets to reach a wider audience.
* **Story Mode:** Create a story mode with narrative elements and objectives to provide a deeper and more immersive gaming experience.
* **Community Content:** Implement features that allow players to create and share their own levels or challenges with the community, increasing replay ability.
* **Accessibility Improvements:** Make the game more accessible by adding features such as adjustable difficulty levels, colorblind mode, or support for assistive technologies.

Pg. 17

**REFERENCES**

Pg. 18

* [**https://www.google.com**](https://www.google.com)
* [**https://www.Youtube.com**](https://www.Youtube.com)
* [**https://developer.mozilla.org/en-US/docs/Web/HTML**](https://developer.mozilla.org/en-US/docs/Web/HTML)
* [**https://devdocs.io/javascript/**](https://devdocs.io/javascript/)
* [**https://devdocs.io/css/**](https://devdocs.io/css/)
* [**https://www.geeksforgeeks.org/**](https://www.geeksforgeeks.org/)
* [**https://www.simplilearn.com/**](https://www.simplilearn.com/computer-vision-article)
* [**https://github.com/**](https://github.com/)
* [**https://chat.openai.com/**](https://chat.openai.com/)
* [**https://www.Wikipedia.org**](https://www.Wikipedia.org)