**JSS MAHAVIDYAPEETHA**



**Mini Project / Internship Assessment**

|  |
| --- |
| **Subject Name: Mini project / Internship Assessment**  **Subject Code : KCS-354** |

COURSE: B.Tech. SEMESTER: IIIrd

**by**

Name of Student (Roll No)

PRASHANT YADAV (2100910100131)

**Department of Computer Science and Engineering**

**JSS ACADEMY OF TECHNICAL EDUCATION**

**C-20/1, SECTOR-62, NOIDA**

**VISION AND MISSION**

**VISION OF THE INSTITUTE**

**JSS** **A**cademy of **T**echnical **E**ducation Noida aims to become an Institution of excellence in imparting quality **O**utcome **B**ased **E**ducation that empowers the young generation with **K**nowledge, **S**kills, **R**esearch, **A**ptitude and **E**thical values to solve **Contemporary Challenging Problems.**

**MISSION OF THE INSTITUTE**

**D**evelop a platform for achieving globally acceptable level of intellectual acumen and technological competence

**C**reate an inspiring ambience that raises the motivation level for conducting quality research

**P**rovide an environment for acquiring ethical values and positive attitude

**VISION OF THE DEPARTMENT**

“To spark the imagination of the Computer Science Engineers with values, skills

and creativity to solve the real-world problems.”

**MISSION OF THE DEPARTMENT**

To inculcate creative thinking and problem-solving skills through effective teaching, learning and research.

To empower professionals with core competency in the field of Computer Science and Engineering.

To foster independent and lifelong learning with ethical and social responsibilities.

**PROGRAM OUTCOMES (POs)**

**Engineering Graduates will be able to:**

**PO1: Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2: Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3: Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4: Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5: Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO6: The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7: Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8: Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9: Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10: Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11: Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12: Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**PROGRAM EDUCATIONAL OUTCOMES (PEOs)**

PEO1: To apply computational skills necessary to analyze, formulate and solve engineering problems.

 PEO2: To establish as entrepreneurs, and work in interdisciplinary research and development organizations as an individual or in a team.

 PEO3: To inculcate ethical values and leadership qualities in students to have a successful career.

 PEO4: To develop analytical thinking that helps them to comprehend and solve real-world problems and inherit the attitude of lifelong learning for pursuing higher education.

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

PSO1: Acquiring in depth knowledge of theoretical foundations and issues in Computer Science to induce learning abilities for developing computational skills.

PSO2: Ability to analyse, design, develop, test and manage complex software system and applications using advanced tools and techniques.

**COURSE OUTCOMES (COs)**

|  |  |
| --- | --- |
| **C224.1** | Undertake problem identification, formulation and design a solution |
| **C224.2** | Solve the real-world problems effectively and adapt with real life working environment. |
| **C224.3** | Acquire skills and knowledge on latest tools and technologies |
| **C224.4** | Develop effective communication skills for presentation of project related activities |
| **C224.5** | Effectively communicate solution to problems through technical reports |

**CO-PO-PSO MAPPING**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO 10** | **PO 11** | **PO 12** | **PSO1** | **PSO2** |
| **C224.1** | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 |
| **C224.2** | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3 |
| **C224.3** | 2 | 2 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 1 | 2 | 3 | 3 | 3 |
| **C224.4** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 3 | 2 | 2 |
| **C224.5** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 3 | 2 | 2 |
| **C224** | 2.40 | 2.40 | 2.60 | 2.60 | 2.40 | 2.40 | 2.60 | 2.60 | 2.60 | 2.40 | 2.20 | 3.00 | 2.60 | 2.60 |

***DECLARATION***

*I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.*

*Signature:*

*Name : PRASHANT YADAV*

*Roll No.: 2100910100131*

*Date : 13-03-2023*

## CERTIFICATE

This is to certify that Mini Project/Internship Assessment Report entitled “ **Game Development Process ( 2d-game using JavaScript )** ” which is submitted by **Prashant Yadav** in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science and Engineering of Dr. APJ Abdul Kalam Technical University, Uttar Pradesh, Lucknow is a record of the candidate’s own work carried out by him/her under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

**ACKNOWLEDGEMENT**

I would like to express my heartfelt gratitude to all those who have contributed to the successful completion of my mini-project.

First and foremost, I want to thank my project guide for their invaluable guidance, support, and encouragement throughout the project. Their expertise in the field and constructive feedback helped me to improve the quality of my work and stay on track.

I would also like to extend my appreciation to the internet community and open-source developers who provided me with access to a wide range of resources, tools, and software which was instrumental in the development of the project.

Lastly, I want to thank my family and friends for their unwavering support and encouragement throughout this journey. Their constant motivation and support helped me to stay focused and committed to the project.

Thank you all once again for your contributions and support towards the successful completion of this mini-project.

**ABSTRACT**

This mini project focuses on designing and developing a 2D game using JavaScript programming language. The game has been built using the Phaser game engine, which is an open-source framework for game development. The game follows a storyline where the player has to navigate through various levels, avoiding obstacles and enemies to reach the end goal. The game's development process involves implementing various game mechanics such as object movement, collision detection, score tracking, and game physics. The project aims to provide practical experience in game development and offers a comprehensive understanding of game design principles and practices using JavaScript. The outcome of the project demonstrates the developer's ability to design and develop a 2D game using JavaScript that is engaging and challenging for players. Through this project, the developer has gained knowledge of various game development techniques, which can be applied in future game development projects. Overall, the project is a success in achieving its objectives of providing hands-on experience in 2D game development using JavaScript

**LIST OF FIGURES:**

1.1 Image of 2d game

2.1 GitHub of the project

2.2 Gimp photo editor

2.3 Vs code with live-server

3.1 Stylesheet editing using Gimp

3.2 List of all HTML, CSS and JavaScript files

3.3 index.html file

3.4 styles file

3.5 Background.js

3.6 Main.js

3.7 UI.js

3.8 The actual game

3.9 Deployment using itch.io

3.10 Analytics after publishing

**LIST OF TABLES:**

2.1………….Technologies Used

**TABLE OF CONTENTS:**

DECLARATION...........................................................................................6

CERTIFICATE…………………………………………………………….7

ACKNOWLEDGEMENTS….....................................................................8

ABSTRACT…..............................................................................................9

LIST OF TABLES...........................................................................................10

LIST OF FIGURES….................................................................................... 10

CHAPTER-1: INTRODUCTION....…………………………………….……12

CHAPTER-2: TOOLS AND TECHNOLOGY USED...……………….……16

CHAPTER-3:DEVELOPMENT……………..……………………………….19

CHAPTER-4:CONCLUSION………………………………………………...27

REFERENCES…...............................................................................................28

**CHAPTER 1: INTRODUCTION**

Game development is a complex process that requires a significant amount of planning, creativity, and technical expertise. The game development process cycle typically consists of several stages, each with its own set of tasks and objectives. In this report, we will provide an overview of the game development process cycle and discuss each stage in detail.

**Stage 1: Conceptualization**

The first stage in the game development process cycle is conceptualization. During this stage, the development team comes up with the initial idea for the game. The idea may be based on a story, a gameplay mechanic, or a combination of both. The development team will also research similar games to ensure that the idea is original and viable.

The main objective of the conceptualization stage is to develop a clear and concise game concept that can guide the rest of the development process. This stage typically involves brainstorming, idea generation, and initial design work.

**Stage 2: Design**

The second stage in the game development process cycle is design. During this stage, the game's mechanics, characters, levels, and overall look and feel are fleshed out. The design team will often create concept art and storyboards to help visualize the game.

The main objective of the design stage is to create a detailed game design document that outlines the game's features, mechanics, and art style. This document serves as a blueprint for the development team and helps ensure that everyone is working towards the same goals.

**Stage 3: Prototyping**

The third stage in the game development process cycle is prototyping. Once the design is finalized, a prototype of the game is created. This is a simplified version of the game that allows the team to test the mechanics and gameplay. The prototype may be refined and tested multiple times before moving on to the next stage.

The main objective of the prototyping stage is to test and refine the game's mechanics and gameplay. This stage typically involves rapid iteration and feedback from the development team.

**Stage 4: Development**

The fourth stage in the game development process cycle is development. In this stage, the game is built using programming languages and software tools. The development team may also create art, animations, and sound effects during this stage.

The main objective of the development stage is to build a working version of the game that is stable and free of bugs. This stage typically involves a significant amount of coding and testing.

**Stage 5: Quality Assurance**

The fifth stage in the game development process cycle is quality assurance (QA). Once the game is built, it is thoroughly tested to ensure that it is free of bugs and glitches. The QA team will play the game multiple times and document any issues they encounter.

The main objective of the QA stage is to ensure that the game is stable, bug-free, and meets the design specifications outlined in the game design document. This stage typically involves a significant amount of testing and bug fixing.

**Stage 6: Deployment**

The final stage in the game development process cycle is deployment. Once the game is complete and has passed QA testing, it is ready to be released to the public. The game may be released on various platforms, such as mobile devices, consoles, and PCs.

The main objective of the deployment stage is to launch the game successfully and generate revenue for the development team. This stage typically involves marketing, distribution, and ongoing support and maintenance.

Using these processes in actual game development, **NINGINBANG: Side-scroller Fast-Paced Action Game with JavaScript!** was developed.

The objective of the game was to learn concepts of ***OOPS, Data Structures in JavaScript*** and the Basics of game development.

**Idea and Game setting:** The game is set in medieval times. A ninja is moving throw a forest from one town to another, and on his way, he encounters many enemies. He takes very few stops, finishes his enemies, and continues to run.

The basic idea revolving around forest and ninjas.

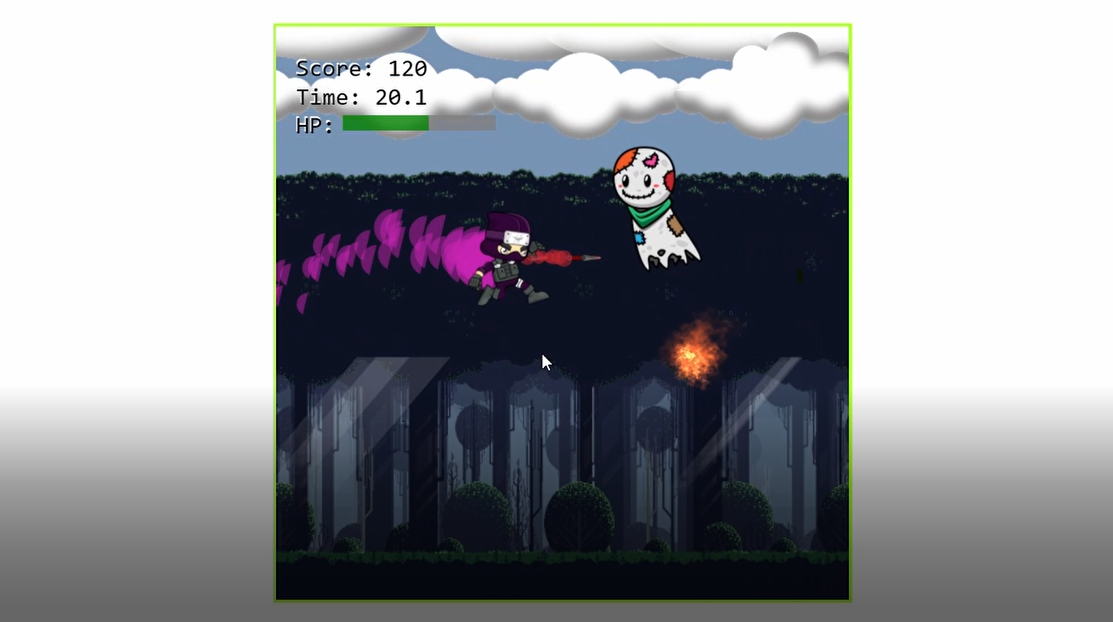


Fig: 1.1

**CHAPTER 2: Tools And Technology Used.**

Game development involves a wide range of tools and technologies, from game engines to programming languages to design software. Here are some of the most commonly used tools and technologies in game development:

**Game engines**: A game engine is a software framework that provides developers with tools for creating and running games. Some popular game engines include Unity, Unreal Engine, and CryEngine.

**Programming languages**: Game developers use a variety of programming languages to create the code that powers their games. Common languages include C++, C#, Java, and Python.

**3D modeling software**: 3D modeling software is used to create the 3D assets that make up the world and characters in a game. Some popular 3D modeling software includes Blender, Maya, and 3ds Max.

**Graphics libraries**: Graphics libraries provide game developers with the tools to create advanced visual effects in their games. Some popular graphics libraries include OpenGL and DirectX.

**Sound and music tools**: Sound and music tools are used to create the audio effects and music in a game. Popular sound and music tools include FMOD and Wwise.

**Integrated Development Environments (IDEs):** An IDE is a software application that provides developers with tools for writing and debugging code. Some popular IDEs for game development include Visual Studio and Eclipse.

**Version control software**: Version control software is used to manage and track changes to the game's code and assets. Some popular version control software includes Git and SVN.

In this project the following technologies and tools were used:-

|  |  |
| --- | --- |
| **Technology and tools** | **Used in this project** |
| *Game Engine* | None |
| *Programming languages* | JavaScript , HTML , CSS |
| *3D-modelling software* | Gimp, Blender |
| *Graphics libraries* | None |
| *Sound and music tools* | Non copyrighted music used. |
| *Integrated Dev Environment(IDEs)* | Visual Studio Code, Sublime text |
| *Version Control Software* | Git |

A screenshot of a computer screen

Description automatically generated

Figure 2.1

A screenshot of a computer

Description automatically generated with medium confidence

Figure 2.2

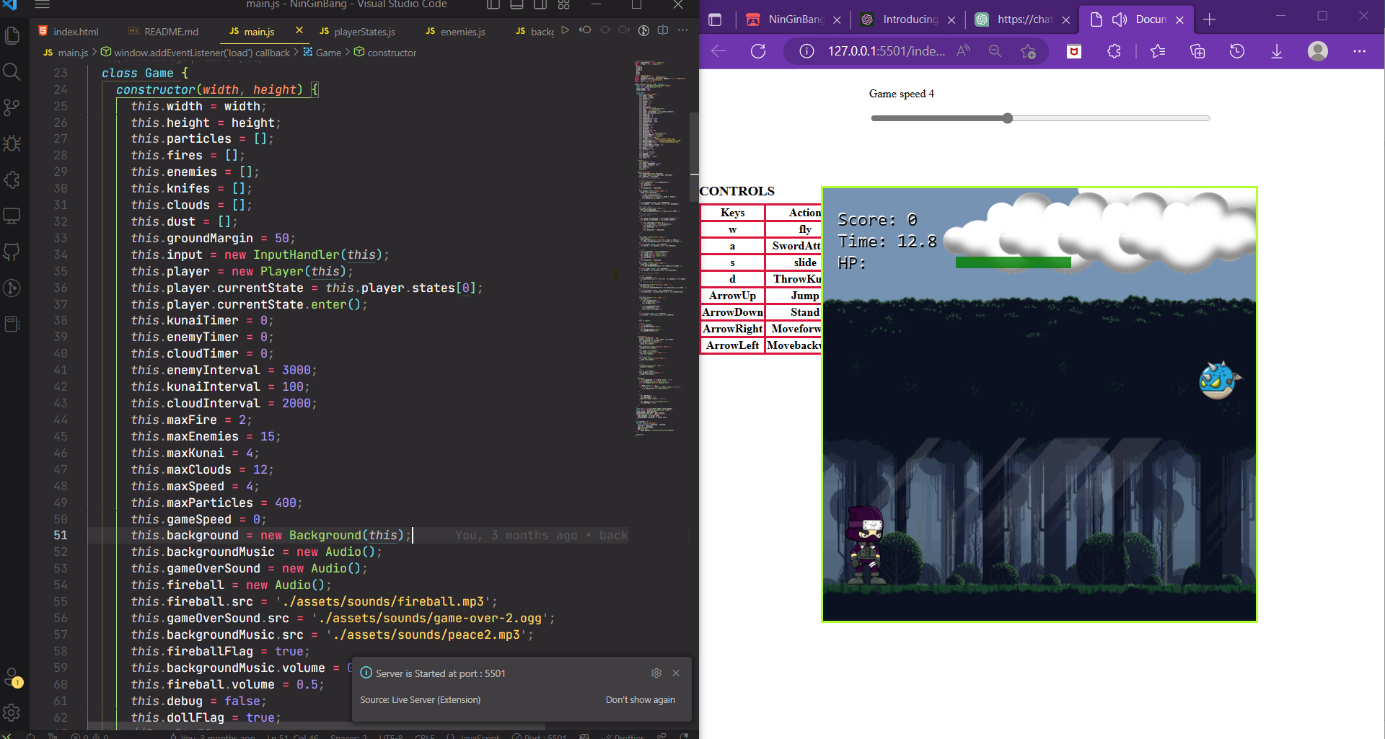


Figure 2.3

**CHAPTER 3: DEVELOPMENT**

**Stage 1: Conceptualization**

**Idea and Game setting:** The game is set in medieval times. A ninja is moving throw a forest from one town to another, and on his way, he encounters many enemies. He takes very few stops, finishes his enemies, and continues to run.

**Stage 2: Design**

Sprite Sheet design for character were used using Gimp software.

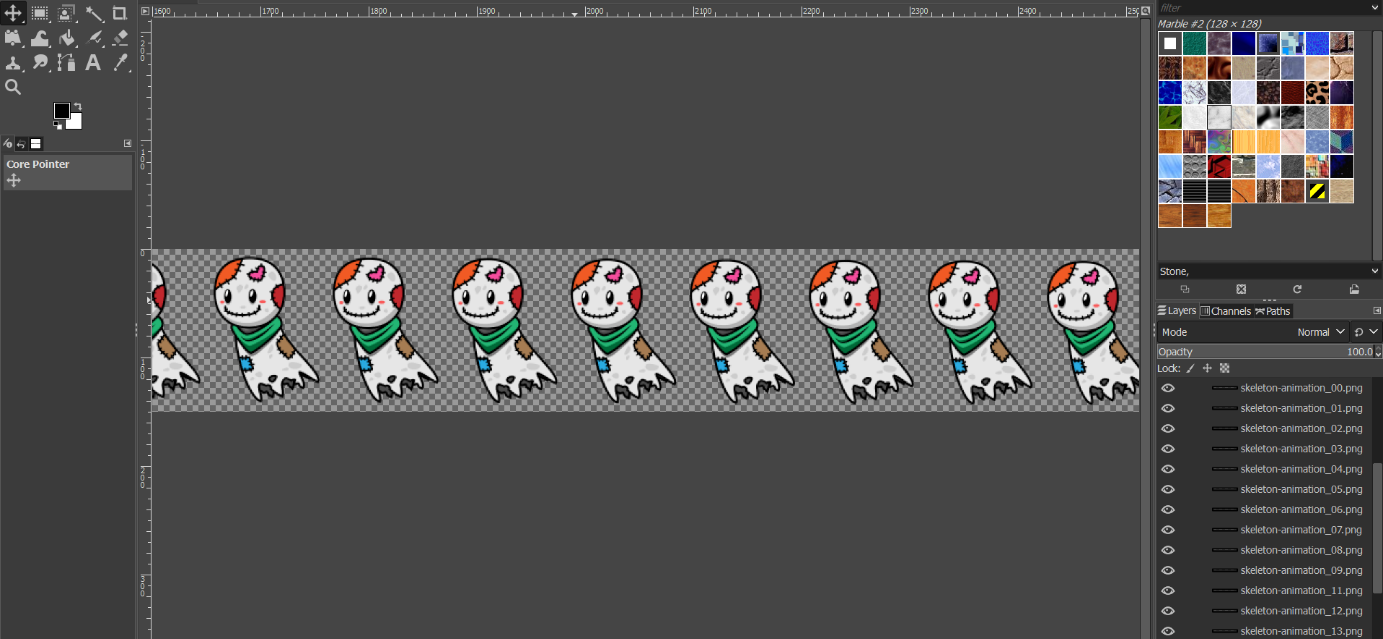
The enemies were designed frame-by-frame and when these frames are in motion it creates animation.

Figure 3.1

**Stage 3: Prototyping and Development**

This was the major part of the project, mostly all of the time went into development.

VS code was used for editing html, CSS and JavaScript files.

The concept of oops were followed.

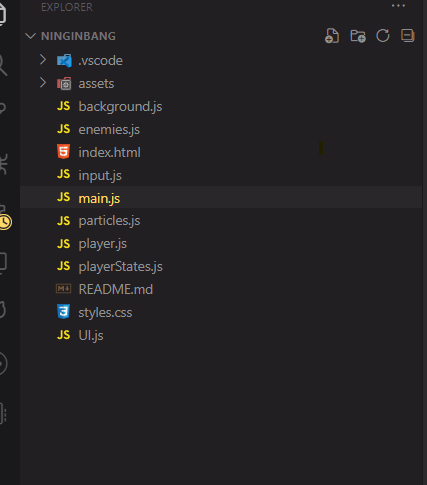


Figure 3.2

Different files were used for different functionality.

The function of each of these files if as follows:

1.**index.html**:

This is the file that contains all homepage Html and all image tags linking the respective images in assets. There is a canvas element in the body tag, on which the whole animation is made is here. Then lastly the scripts with the module linking the main.js are here and the styles.css.

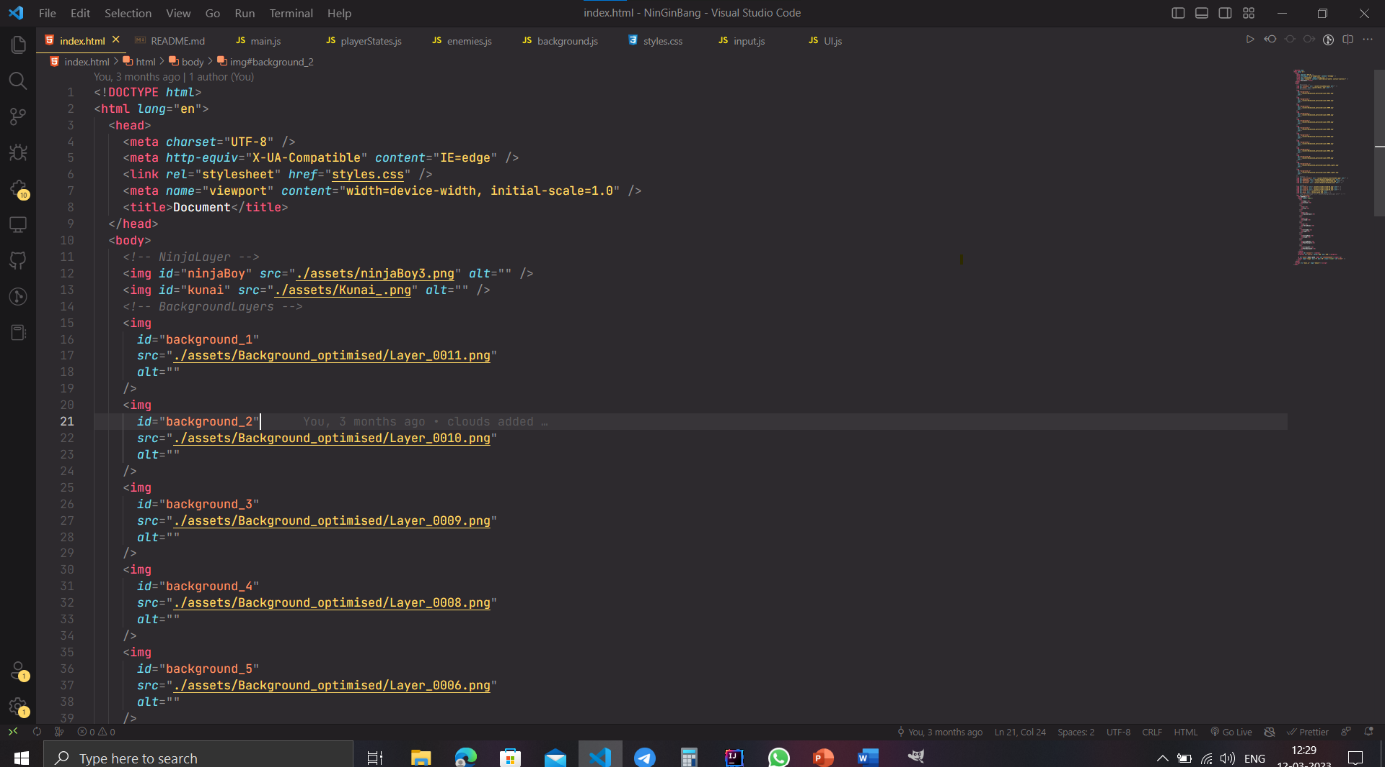


Figure 3.3

2.**styles.css**:

In this file, all the images are given and displayed as hidden to prevent them from appearing on the homepage and appear only when needed. Mostly the id attribute was used for the Html element. This file also contains the CSS for the HTML Canvas.

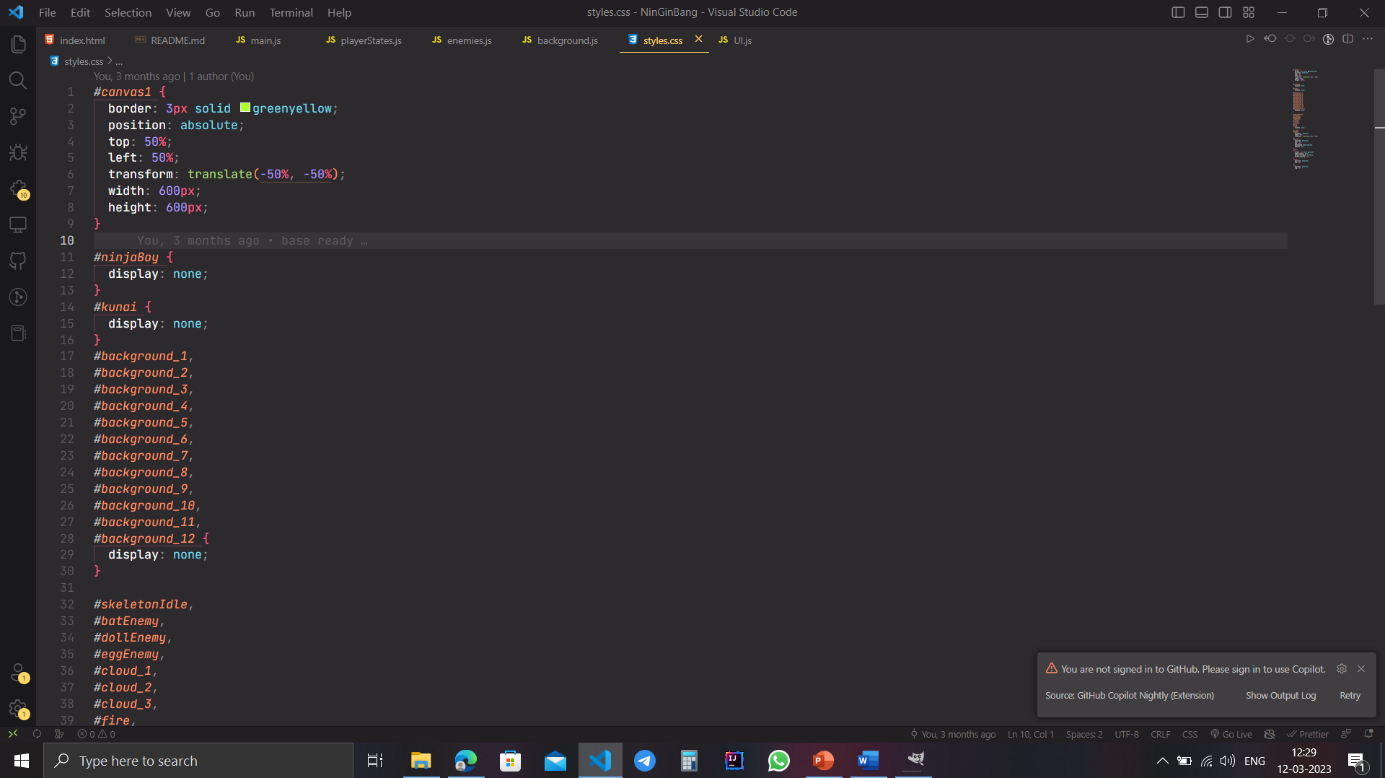


Figure 3.4

3.**background.js**:

This file handles the background layers. It has two classes the Background class which handles all the background layers and the Layer class. The backgroundLayers array stores all the layers. The layer class has a constructor update and draw method, which when called draws full background. Each layer is given a different speed to make a parallax-like effect making the background look more alive.

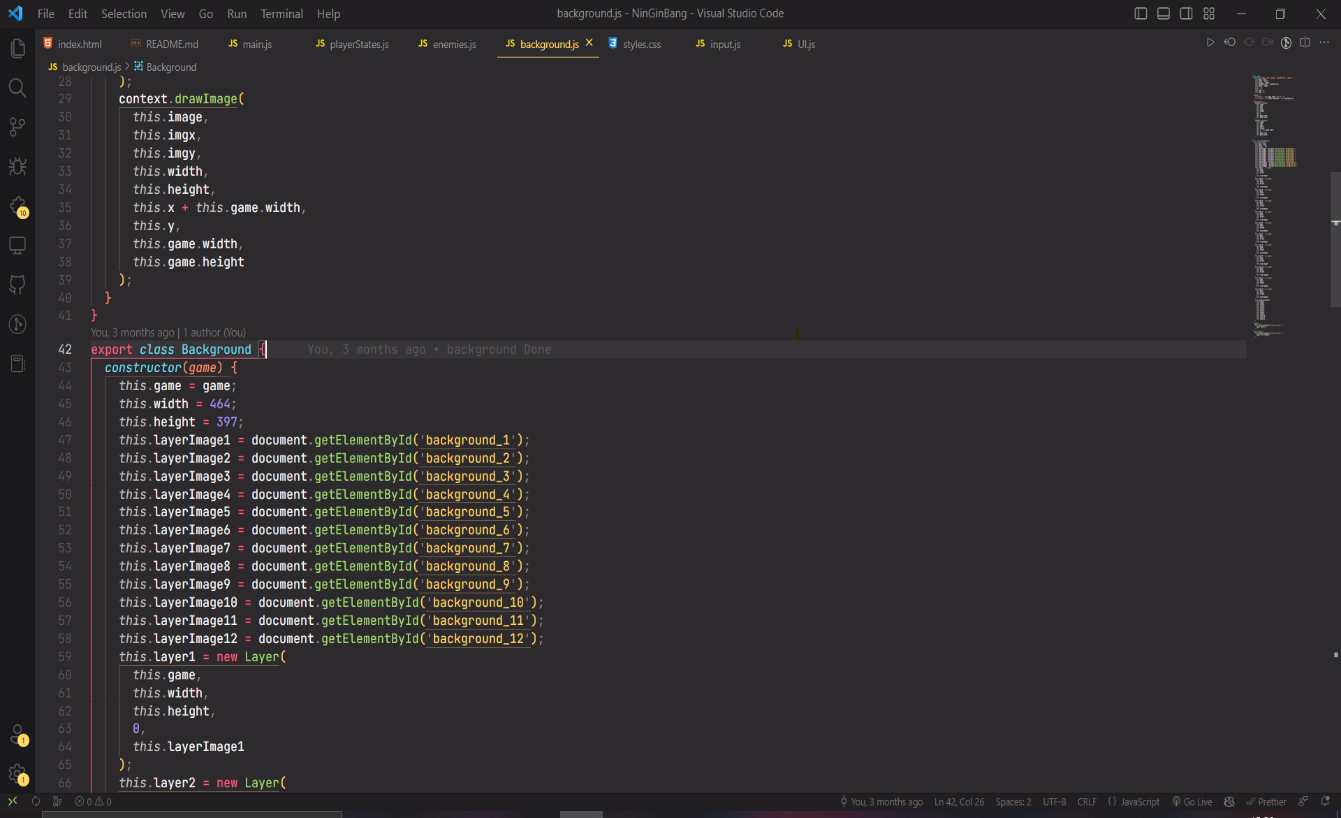


Figure 3.5

4.**enemies.js**:

This file contains the main class Enemy and four subclasses Skeleton,EggEnemy,DollEnemy, and BatEnemy which extend into the main Enemy class.

5.**input.js**:

This file contains the class InputHandler. The constructor takes in the game as parameter.this.keys in an array that stores the keys input from the keyboard. The windows.addEventListener listens for the event key down and the listener arrow function (e)=> checks if the key pressed was the few specific keys such as ArrowUp, ArrowDown, ArrowLeft, ArrowRight, w, a, s, d and pushes them in the keys array, if they are not already present in it. Addition checks for y, Enter keys for debugging, and restart functionality is checked. Similarly, the window.addEventListener for keyup event, executes a function to check for the keys and deletes the from the keys array. This file also contains the restartGame() function which is triggered after the player dies and Enter is pressed. This function resets the various game parameters such as score to restart the game.

6.**main.js**:

The main scripts of the game are where all the other connections and animations are made. The game makes use of an array data structure for storing the objects such as enemies, particles, fires, knives, clouds, and dust. forEach method is called on each of these arrays and they are drawn on canvas, they are updated and new objects are pushed and removed from them as required.

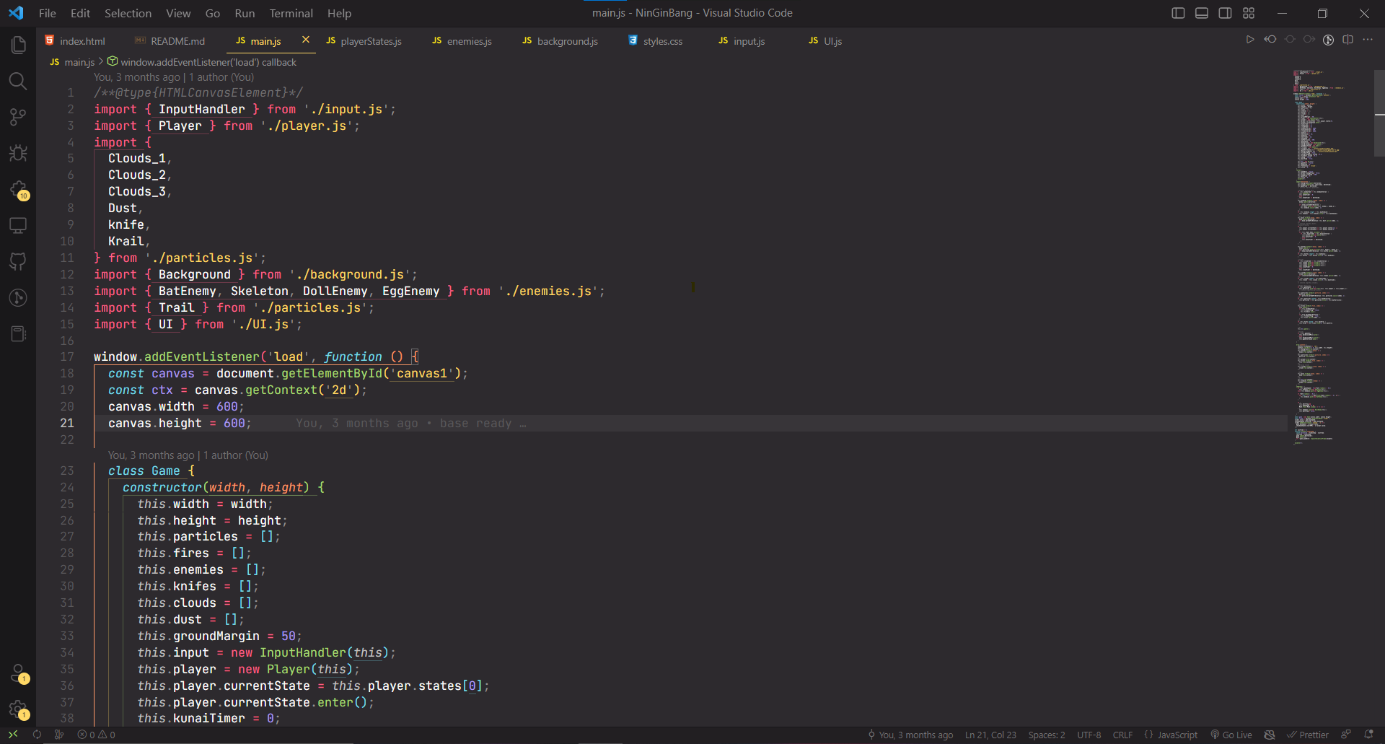


Figure 3.6

7.**particles.js**:

This file contains the Knife class which handles the kunai knives thrown by the player. The working of it is similar to above mention Enemy class. The kunai is marked for deletion once it hit the enemies. Secondly, the Dust class handles the aftereffect of enemy death. A dust cloud is animated at the place where the enemies die. Thirdly, the Clouds class draws clouds over the canvas. It has three subclasses that animate three different types of cloud simultaneously but moving at different speeds at different heights. Lastly, the Particles class draws particles such as player trails, and kunai knife trails. The files also contain another class Fire which draws a fireball throw out of dollEnemy's mouth.

8.**player.js**:

The Player class holds the player's info and draws and updates the player. It imports various player states and uses them on the player sprite to perform different movements.

9.**playerStates.js**:

This file handles the player states which include:

STANDING

RUNNING

JUMPING

ATTACKING

CLIMBING

DYING

FLYING

It contains the State class which has various subclasses namely all the above-mentioned states' classes. These have a constructor with parameter state and game. enter() This method sets the frameY of the player sprite sheet to that specific state action. handleInput() This method is used to transition from one state to another.

10**.UI.js**:

The UI drawn on top of the canvas is handled here. The score, timer, HP, and death screen message all is handled by UI Class.

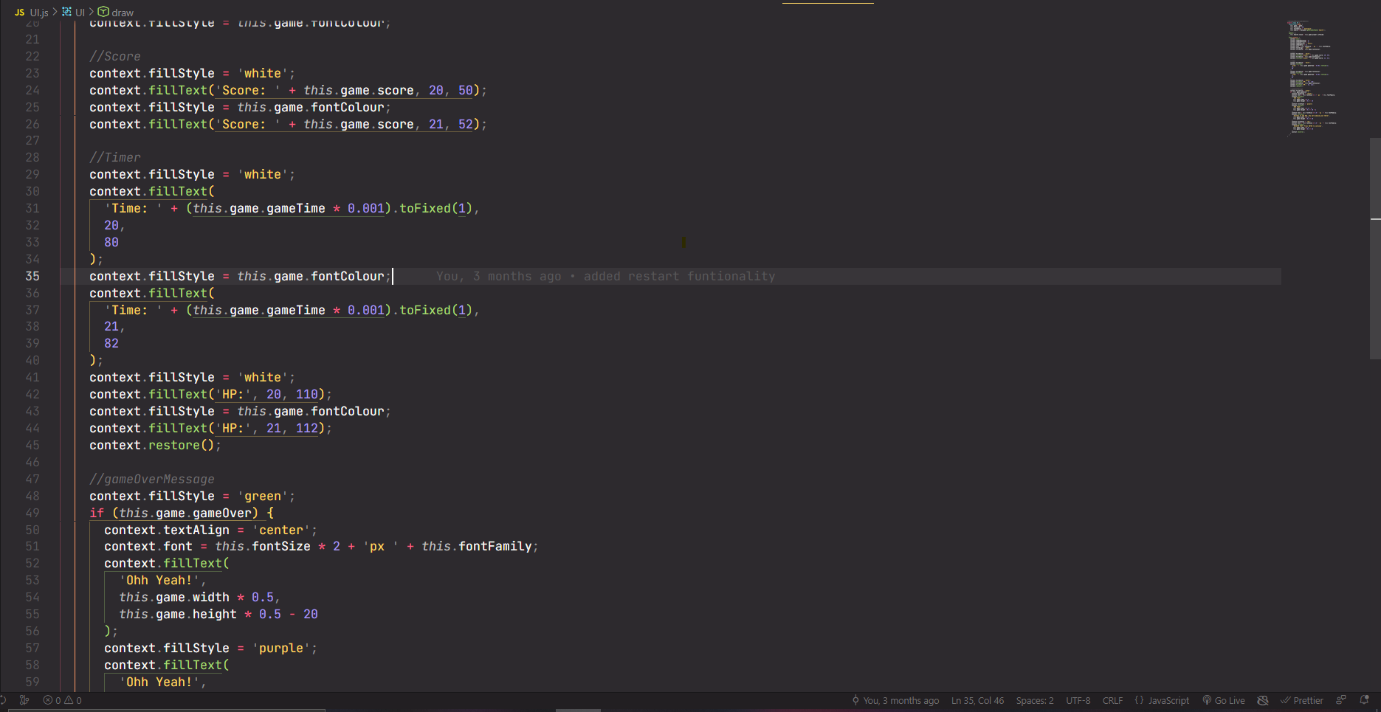


Figure 3.7

**Stage 4 : Quality Assurance and deployment:**

Using the live server extension of VS code, the game can be deployed in browser.

Here changes can be made in game for better visual and graphics.

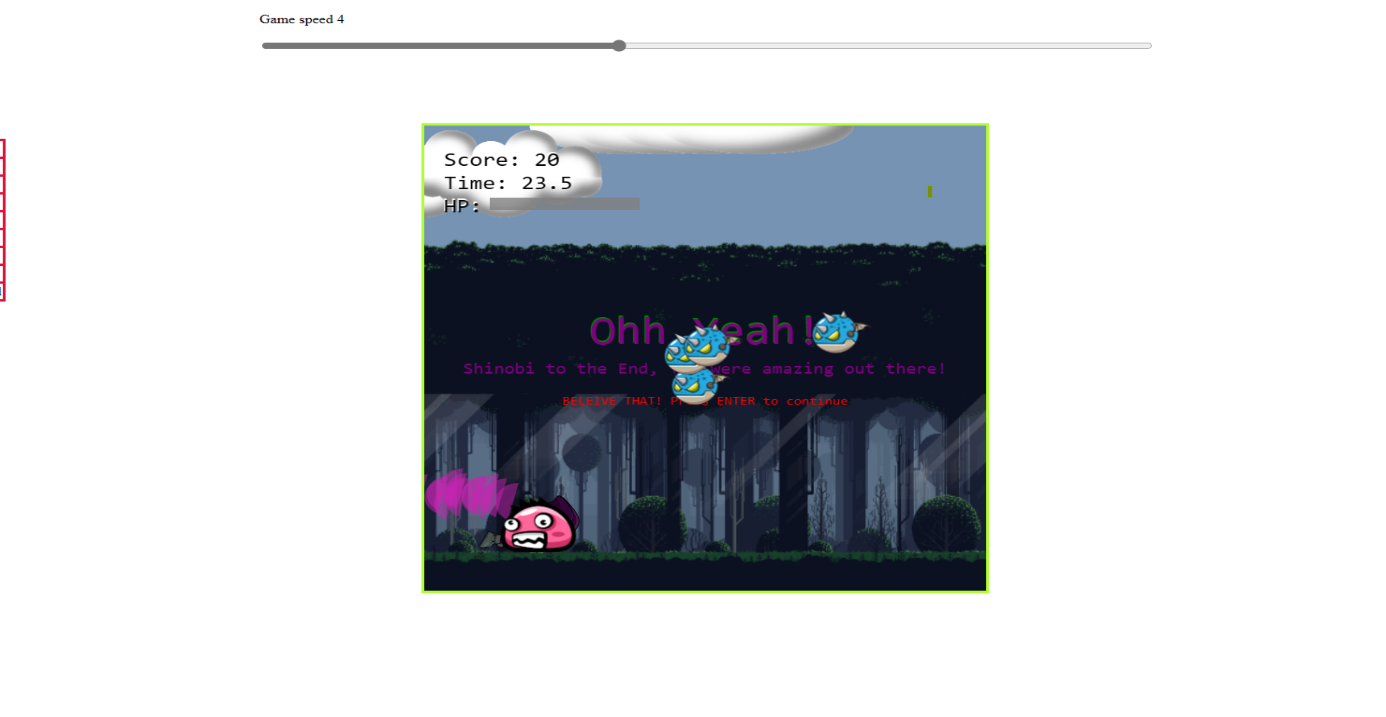
****

Figure 3.8

The game can be published on sites like itch.io for reaching wider audience and monetization.

These sites provides custom url for your game so that it can be assessed from any other device allowing anyone to play.

Some other website such as steam, epic games etc and provide platform for indie game.

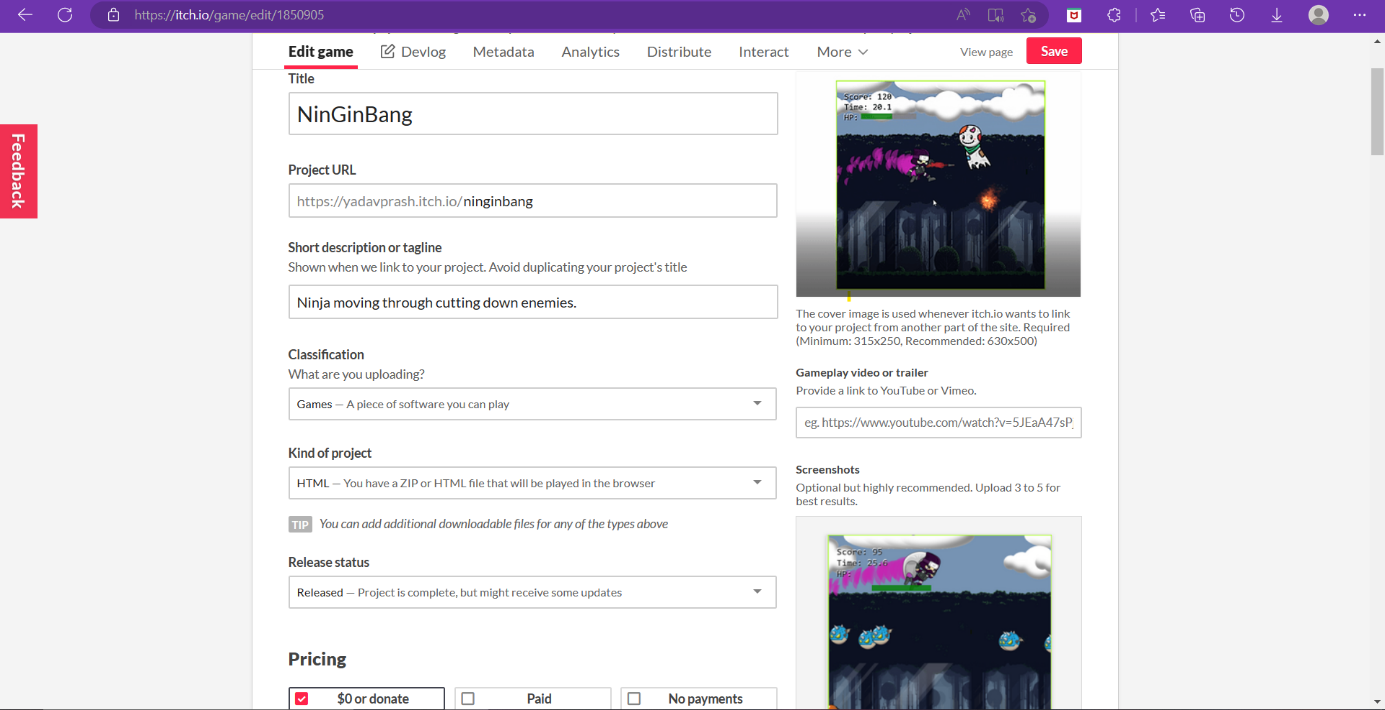


Figure 3.9

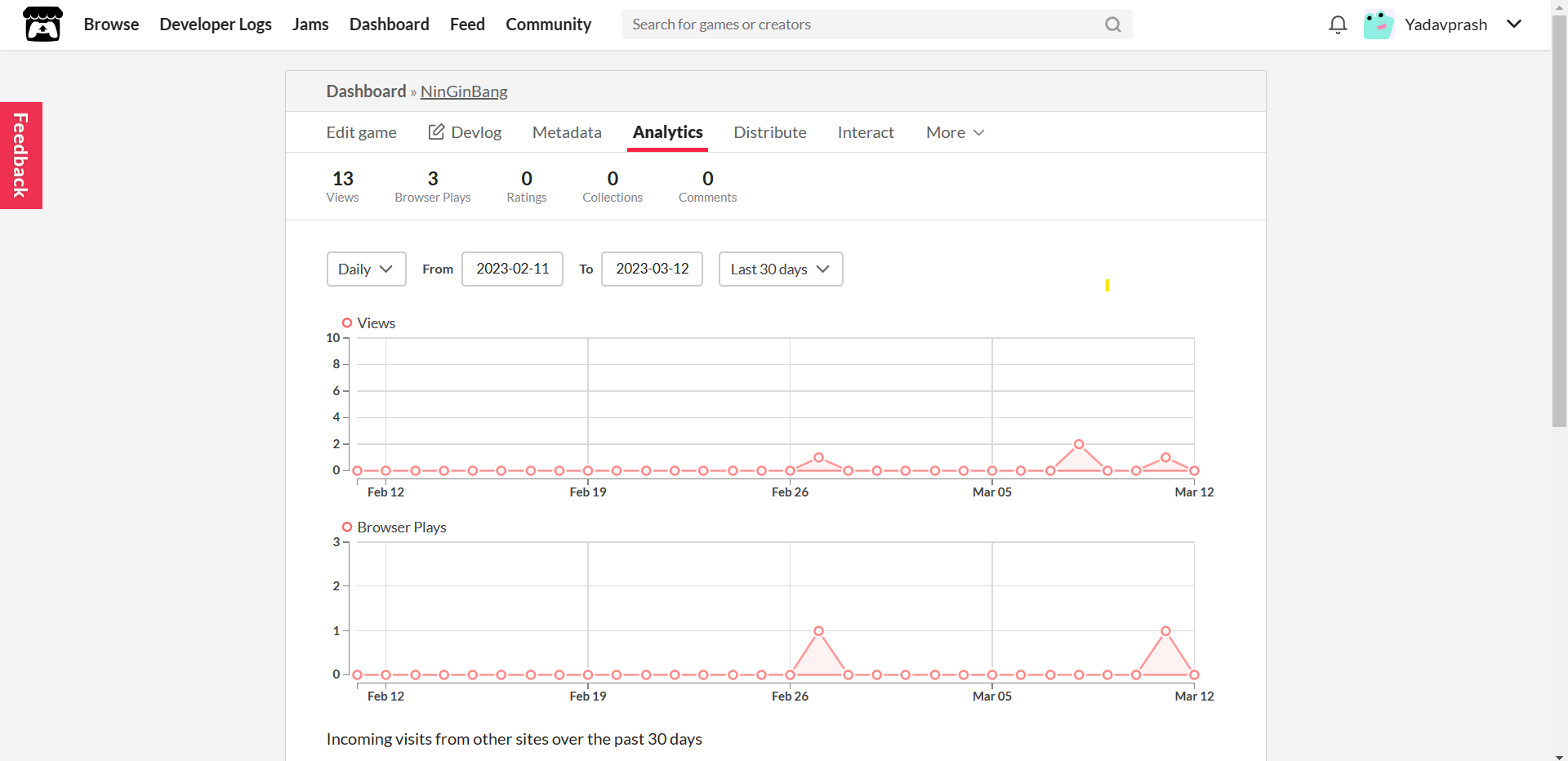


Figure 3.10

**CHAPTER** **4: CONCLUSION**

In conclusion, 2D game development with JavaScript is a popular and exciting way to create interactive games for a wide range of platforms, including web browsers, mobile devices, and desktop computers. The power and flexibility of JavaScript, combined with the ease of use of popular game engines such as Phaser and PixiJS, make it possible for developers of all skill levels to create engaging and entertaining games.

By using JavaScript for game development, developers can take advantage of the vast array of libraries and frameworks available to them, allowing for rapid development and iteration. Additionally, the ability to publish games online and access a global audience provides a unique opportunity for developers to showcase their talents and potentially reach a massive audience.

While 2D game development with JavaScript can be challenging, it is also a rewarding and enjoyable experience. With the right tools, techniques, and mindset, developers can create fun and engaging games that will entertain players for hours on end. As the gaming industry continues to evolve and grow, 2D game development with JavaScript is sure to remain a popular and important area of focus for developers around the world.

**REFERENCES:**

1. "Game Development Essentials: An Introduction" by Jeannie Novak: This book provides a comprehensive overview of the game development process, from concept to design programming and beyond.
2. "The Art of Game Design: A Book of Lenses" by Jesse Schell: This book offers a unique approach to game design, providing a series of "lenses" through which designers can view their games and identify areas for improvement.
3. "Game Programming Patterns" by Robert Nystrom: This book offers practical advice for writing clean, efficient code for games, with a focus on object-oriented programming and design patterns.
4. <https://hackernoon.com/step-by-step-guide-to-push-your-first-project-on-github-fec1dce574f>
5. <https://www.youtube.com/watch?v=CY0HE277IBM&list=PLYElE_rzEw_uryBrrzu2E626MY4zoXvx2>
6. <https://bevouliin.com/>
7. [GitHub - Yadavprash/NinGinBang: Side-scroller Game with Javascript](https://github.com/Yadavprash/NinGinBang)