MONTCLAIR STATE UNIVERSITY

POKÉCATCHER

An Educational game that teaches programming concepts

Master Project in Computer Science

Submitted in Partial Fulfillment

of the Requirement for the

Degree of Master of Science

Advisor Dr. John Jenq

Rishika Devaragatla

Department of computer Science

devaragatlar1@montclair.edu

Acknowledgement

I would like to take this opportunity to express my sincere gratitude to all those who have supported and guided me in the successful completion of my project. This project has been an enriching experience, and I could not have achieved it without the support and guidance of the following individuals:

Firstly, I would like to extend my sincere thanks to my project advisor, Dr. John Jenq. His unwavering support, expertise, and encouragement were instrumental in shaping the direction and scope of this project. His guidance and feedback helped me to overcome many challenges and provided me with invaluable insights and perspectives that helped me to refine and improve the final product.

I would also like to acknowledge the support and resources provided by Montclair State University and the Department of Computer Science. The academic and professional training I have received through the master's program has helped me complete this project.

Furthermore, I would like to thank my friends who have supported me throughout the project. Their feedback, suggestions, and constructive criticism helped in refining the final product.

Finally, I would like to thank my husband Nithin Sai, and my family for their unwavering support, encouragement, and patience. Their support and understanding were essential in allowing me to focus on completing this project.

TABLE OF CONTENTS

1. Abstract

Games for learning are being employed in a variety of fields to motivate students and improve their learning experiences. This novel approach to technology-enhanced learning has caught the interest of educators and researchers in the discipline of programming concepts, which is one of the most cognitively demanding subjects in Computer Science. Teaching beginners or kids any new concepts with the help of games would make the concepts more interactive and easier to understand. This helps in building their creativity in one or the other way and helps in developing problem-solving skills. I have built a game "PokéCatcher" which focuses on teaching basic programming concepts to beginners/kids by building sequential code blocks to perform tasks like catching Pokémon into poké balls.

2. Introduction

It has never been a better moment for beginners/children to learn to code. Coding has become an extremely crucial talent to acquire in the current world, from increasing future work chances to developing technical and social abilities. Indeed, overall employment in computer science sectors is predicted to increase by 15% over the next ten years! With all of this in mind, it is more crucial than ever to discover strategies to inspire beginners/children to learn to code. The traditional teaching techniques would not be appropriate and engaging.

Game-based learning is the best-suited approach for learning new concepts. Combining education through games(Gamification) makes the learning process engaging. This makes the learners more interested and motivated to learn the concept. Through games, users can learn the concept without even knowing that they are learning a new concept.

Educational games use graphical user interfaces, animations, sound, and interesting scenarios to solve and learn a concept. Teaching concepts through games helps the users to easily comprehend and understand the concept and will also potentially benefit them in learning other related concepts. Learning concepts through games helps the users understand the concept deeper and improve their problem-solving skills, creativity, critical thinking, and their interest to learn the concept because games are built-in real-world situations. Gamification has been demonstrated to be an effective teaching technique in a variety of areas, including coding and programming.

The goal of the project is to produce a fun, interactive web-based game that teaches basic coding concepts that are essential and common in most programming languages. This game is intended for beginners or kids who are being introduced to programming and coding concepts for the first time.

Concepts like variables, datatypes, conditional statements, and iterative statements are common in all programming languages. This game focuses on educating the users about these concepts. It also teaches the structure and syntax of these concepts.

This game is best suited when accompanied in an educational environment where the basic concepts are explained, and the same concepts learned are practically learned through this game.

3. Target Audience

This game is designed for beginners or young learners who want to learn the basics of programming language concepts that are common over all the trending programming languages in a fun, interactive, and engaging way. This game is designed for students in primary school, middle school, and high school as well as adults who are interested in learning coding as a new skill.

This game is great for people who want to pursue a career in software engineering, web development, game development, or any other industry that requires coding skills. It is also appropriate for people who want to learn to code as a hobby, as an instructor, or learn more about the technology/engineering industry.

The game is designed to be accessible to beginners, so no prior knowledge of programming knowledge is required. The gameplay mechanics and challenges gradually increase in difficulty, allowing players to progress at their own pace and build a strong foundation of coding concepts.

Overall, the target audience for the game is individuals who are eager to learn programming concepts in a fun and engaging way.

4. Specification

4.1 Concept

The aim of PokéCatcher is to learn programming concepts in a fun and engaging way. Initially, users will learn about data types and variables and use that knowledge to fill up their basic information. Next users have to drag and drop code blocks sequentially to catch the Pokémon in the poké balls. Each level will introduce a new programming concept, gradually increasing the complexity.

4.2 Platform

The game is designed for multiple platforms including desktops and mobile devices. This implies that players will be able to access this game on their personal computers, smartphones, and tablets, making it more feasible to learn programming concepts on the go.

4.3 Game Mechanics

The game includes interactive challenges and puzzles in which players must arrange code blocks to capture Pokémon's. The gameplay is designed to be interesting and instructive, with a focus on teaching playing coding basics like conditional statements, iterative statements, and sequencing.

4.4 Programming Languages

Players will drag and drop code blocks to form instructions and sequences in the game's visual coding language. This will allow gamers to understand coding ideas without having to worry about programming language syntax and organization. Movement, capture, and other gaming features will be represented by code blocks, making it more intuitive and easier to understand.

4.5 User Interface

The user interface will be simple and easy, with a code editor that allows players to drag and drop code blocks, test their code, and capture Pokémon's. The interface is tailored for beginners, with simple and succinct instructions as well as visual clues to help players.

4.6 Graphics

The game is designed with colorful and engaging graphics, with a focus on Pokémon characters. The graphics are visually appealing and impressive, making it more enjoyable for players.

4.7 Progress Tracking

The game includes a progress tracking system that allows players to see their progress, earn achievements, and unlock new levels. This will provide players with a sense of achievement and motivation to capture more Pokémon's while learning concepts.

5. Design and Implementation

PokéCatcher is a browser-based educational 2D game. The design of the game follows the same principles mentioned in the specification section. The game has been developed using JavaScript, HTML, and CSS.

JavaScript is a versatile scripting language that can be used in game development to create highly engaging virtual experiences. JavaScript takes care of all the interactions in the game. JavaScript EventListerners is a key feature of the language that allows the creation of interactive web-based games. Essentially JavaScript EventListener is a function that awaits for a particular event to occur.

Some of the JavaScript EventListerners used in this game are:

Mouse Events: click, dragstart, dragend, dragenter, dragover, dragleave, drop

Window Events: load

To use an event listener in JavaScript, I typically followed these steps:

1. Select the HTML element that the event listener will be attached to using the Document Object Model (DOM) API.
2. Create the event listener function that will be called when the event occurs.
3. Attach the event listener function to the HTML element using the addEventListener() method.

HTML and CSS manage the structure and style of the website. HTML is a markup language that describes the webpages. CSS are style sheets that enhance the website. CSS is all about the colors, fonts, layouts, etc. of the website designed in HTML. For the graphics on the website, I have used the canvas tag. The canvas tag helps to draw graphics, it is a container that when combined with JavaScript will draw the graphics on the website.

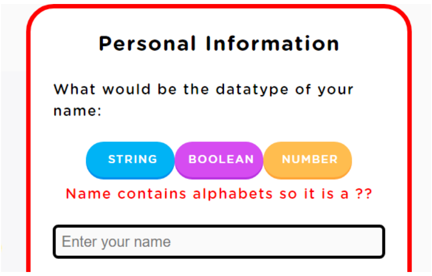
These three combined helped in creating various components of the game and bringing in animations and characters in the game.

The user's name and their progress through the game must be stored. For the data storage, I have used the browser's local storage supported by JavaScript web pages. It stores data in a key-pair format. For my game, I have used local storage of the browser to store the player's name and the player's progress through the game. If the user wants to reset the game only the user's name is stored, and the user's progress is removed from the local storage.

6. Key features

6.1 Login Page

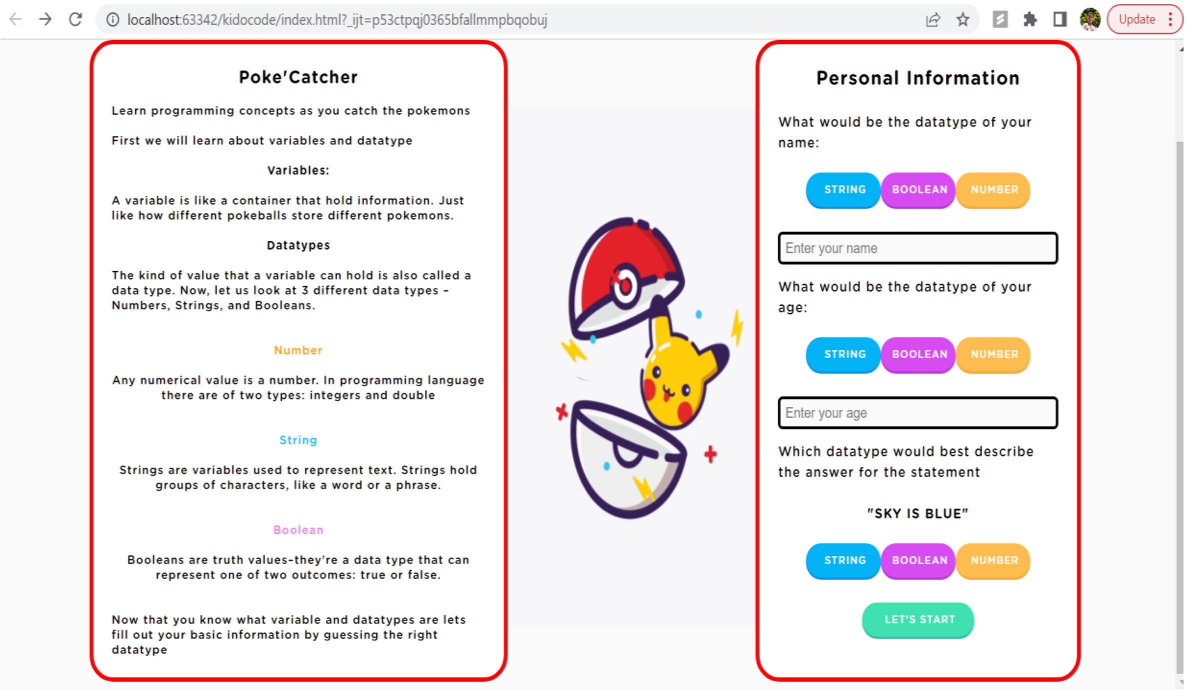
The Login Page is the first page the user land on once he opens the website. The login page collects the basic information of the users. But before that, the user has to learn about datatypes and variables and implement them to fill in their basic information. For example, in the game, if the user wants to fill in their name the user must answer what is the datatype of their name only then they can enter the name. To make sure the user selects the right answer form validations are used. If the user selects a wrong answer, then a prompt appears giving a hint as shown in the second picture.



Personal Information

What would be the datatype of your name:

Name contains alphabets so it is a ??


localhost:63342/kidocode/index.html?_ijt=p53ctpqj0365bfallmmpbqobuj
Update

Poke'Catcher

Learn programming concepts as you catch the pokemons

First we will learn about variables and datatype

Variables:

A variable is like a container that hold information. Just like how different pokeballs store different pokemons.

Datatypes

The kind of value that a variable can hold is also called a data type. Now, let us look at 3 different data types - Numbers, Strings, and Booleans.

Number

Any numerical value is a number. In programming language there are of two types: integers and double

String

Strings are variables used to represent text. Strings hold groups of characters, like a word or a phrase.

Boolean

Booleans are truth values-they're a data type that can represent one of two outcomes: true or false.

Now that you know what variable and datatypes are lets fill out your basic information by guessing the right datatype

Personal Information

What would be the datatype of your name:

Enter your name

What would be the datatype of your age:

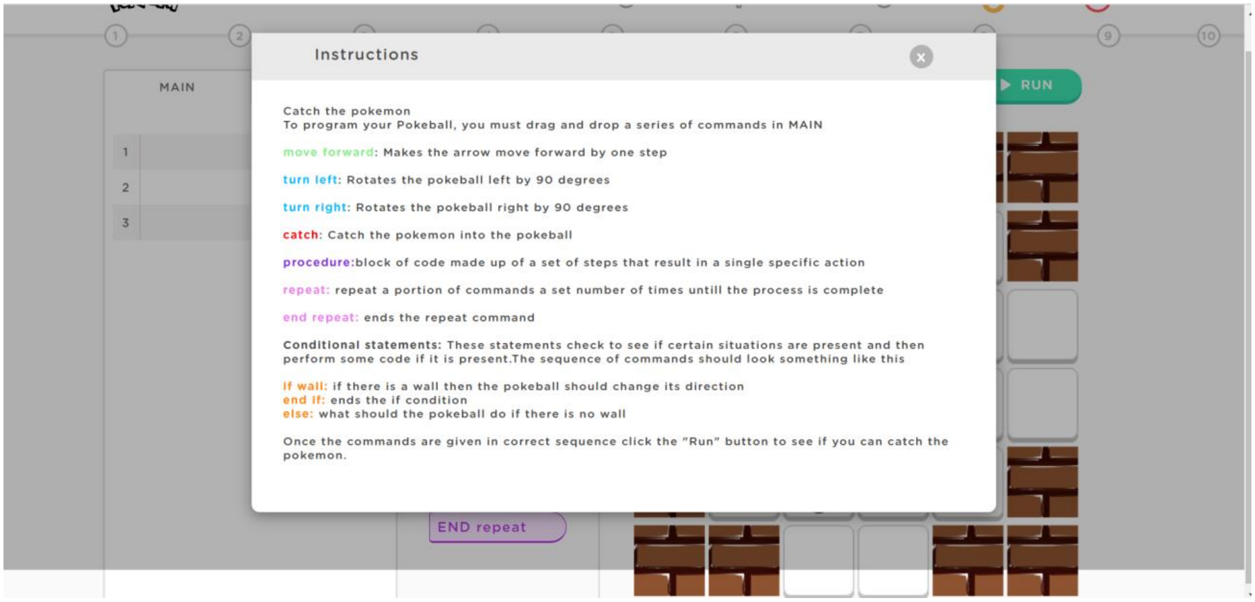
Enter your age

Which datatype would best describe the answer for the statement

"SKY IS BLUE"

6.2 Instructions Window

As soon as the user clicks the "lets start" button after entering their user details the instruction window pops up on the next page. It consists of the entire details and instructions of what each command does in the game. The instructions window is shown below.



Catch the pokemon
To program your Pokeball, you must drag and drop a series of commands in MAIN

move forward: Makes the arrow move forward by one step

turn left: Rotates the pokeball left by 90 degrees

turn right: Rotates the pokeball right by 90 degrees

catch: Catch the pokemon into the pokeball

procedure:block of code made up of a set of steps that result in a single specific action

repeat: repeat a portion of commands a set number of times untill the process is complete

end repeat: ends the repeat command

Conditional statements: These statements check to see if certain situations are present and then perform some code if it is present.The sequence of commands should look something like this

If wall: if there is a wall then the pokeball should change its direction
end if: ends the if condition
else: what should the pokeball do if there is no wall

Once the commands are given in correct sequence click the "Run" button to see if you can catch the pokemon.

6.3 Level Indicator

The level indicator displays the player's progress through the game. It is an important component of the game's user interface. It helps the player track their progress and understand where they are in the game.

The level indicator typically consists of a series of icons indicating the level numbers. As the player completes each level the corresponding level icon will be marked green to show that it is completed.

The level indicator is visible all the time so that the player can easily see their progress. The level indicator also serves as a motivational tool. Seeing their progress visually represented will help keep the player engaged and motivated to continue playing and learning.

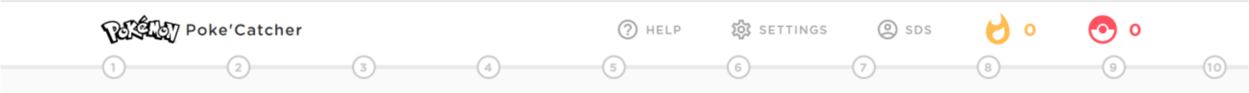


Fig: Level Indicator before the start of the game

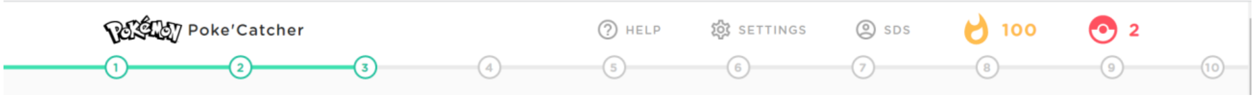
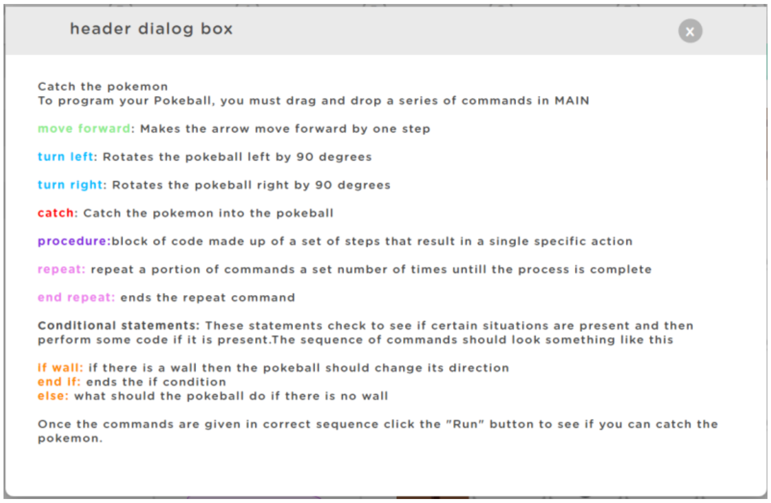


Fig: Level Indicator after 3 levels of the game

6.4 Menu bar

Help: List outs the instructions again



header dialog box

Catch the pokemon
To program your Pokeball, you must drag and drop a series of commands in MAIN

move forward: Makes the arrow move forward by one step

turn left: Rotates the pokeball left by 90 degrees

turn right: Rotates the pokeball right by 90 degrees

catch: Catch the pokemon into the pokeball

procedure:block of code made up of a set of steps that result in a single specific action

repeat: repeat a portion of commands a set number of times untill the process is complete

end repeat: ends the repeat command

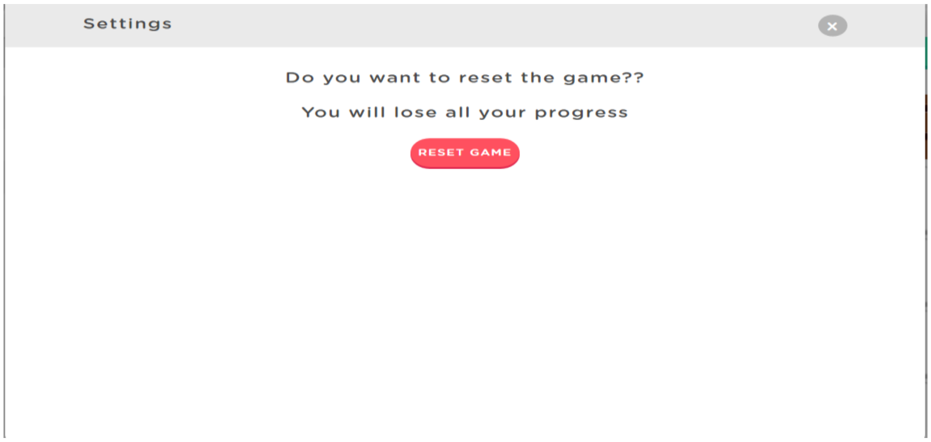
Conditional statements: These statements check to see if certain situations are present and then perform some code if it is present.The sequence of commands should look something like this

if wall: if there is a wall then the pokeball should change its direction
end if: ends the if condition
else: what should the pokeball do if there is no wall

Once the commands are given in correct sequence click the "Run" button to see if you can catch the pokemon.

Fig: Help window

Settings: This option allows the players to reset the game to its initial state. This can be useful if they want to start over or try different strategies.



Settings

Do you want to reset the game??

You will lose all your progress

Fig: Settings Window

Profile, XP indicator, Pokémon count: The profile displays the user's name. XP indicator displays the score the user has acquired until the present level which can help motivate them to continue playing and striving for improvement. Pokémon count displays the number of Pokémon's the player has successfully caught. This is a fun and rewarding way to track progress and provide a sense of achievement.

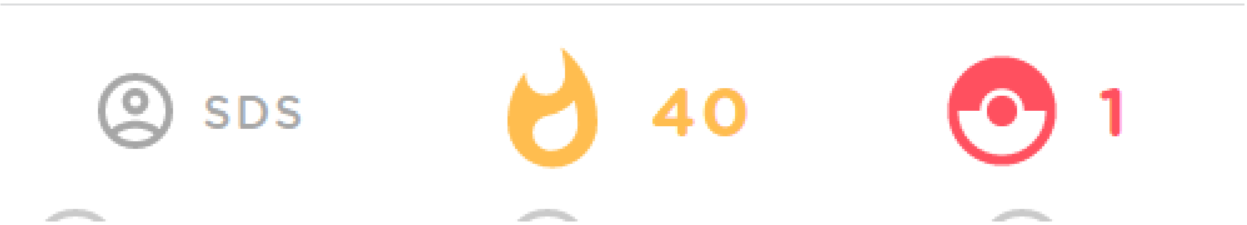
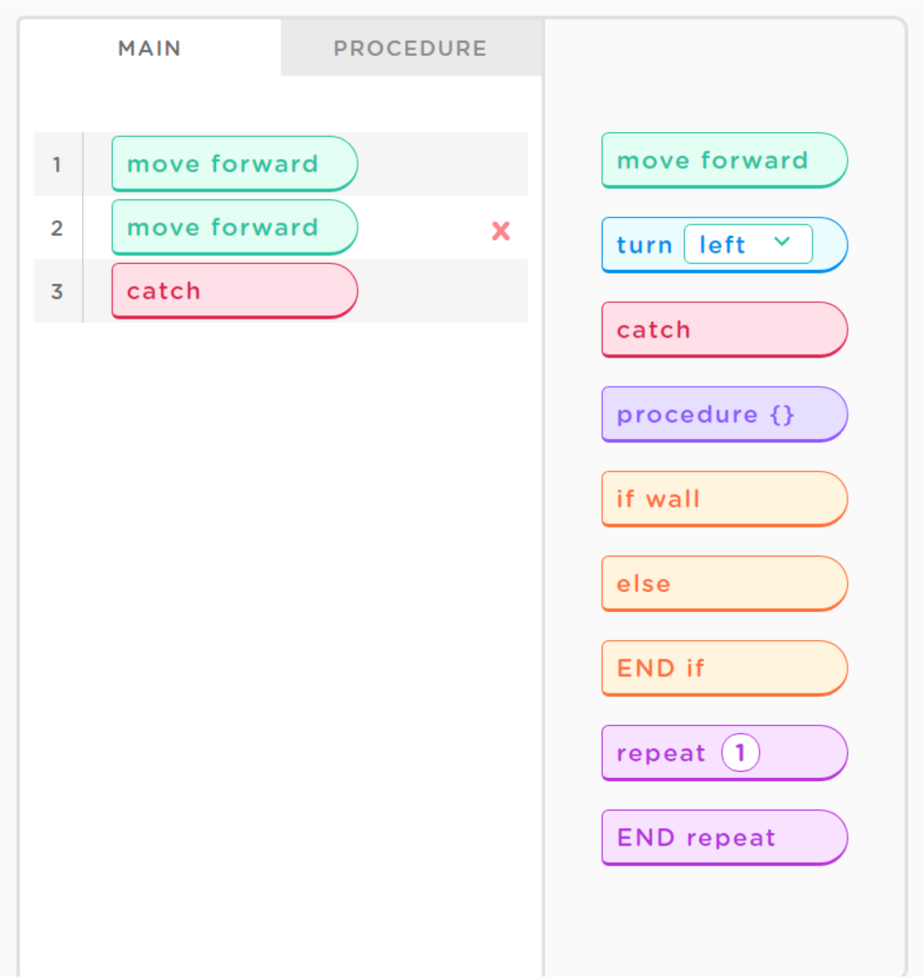


Fig: Profile, XP counter, Pokémon counter

6.5 Code Editor:

The code editor will be the main interface for the players to write and arrange code blocks to catch the Pokémon's. HTML and CSS are used to create the editor with customizable font, syntax highlighting, and line numbers. The code editor will have two sections one is Main section which acts like a programming language main which contains the code and function calls. Next is the procedure to write code blocks for a function. Depending on the level the number of lines for each of the section changes.



```
1 move forward
2 move forward
3 catch
```

Fig: Code editor showing Main, Main calling procedure and Code in Procedure

6.6 Code Blocks

Code blocks are the central feature that players use to create functioning programs. The game mechanics requires the players to drag and drop the code blocks into the code editor in a specific order to control the in-game poké ball movement and actions.

The code blocks in the game are designed to represent specific programming concepts such as conditional statements, iterative statements, and procedures. Each block has a unique color making it easier for players to identify and differentiate between them.

Players start by selecting code blocks from the palette and arranging them in the code editor to create a program. The game's challenges require players to use the code blocks to navigate the poké ball through a maze-like environment, avoiding obstacles and capturing the Pokémon along the way.

As players progress through the game, they encounter more complex challenges that require them to use a wider range of programming concepts and strategies. For example, players might have to use conditional statements to navigate through branching paths or use loops to repeat the action until a condition is met.

The use of code blocks in this game makes it more accessible to players who are new to programming or who may struggle with traditional text-based programming languages. By removing the complexity of syntax and structure, players can focus on the underlying logic and critical thinking skills that are essential to programming. The use of the code blocks is a key feature of this game, providing a visual and tactile way for players to learn programming concepts and develop important problem-solving skills. The code blocks used in this game are shown in the below figure.

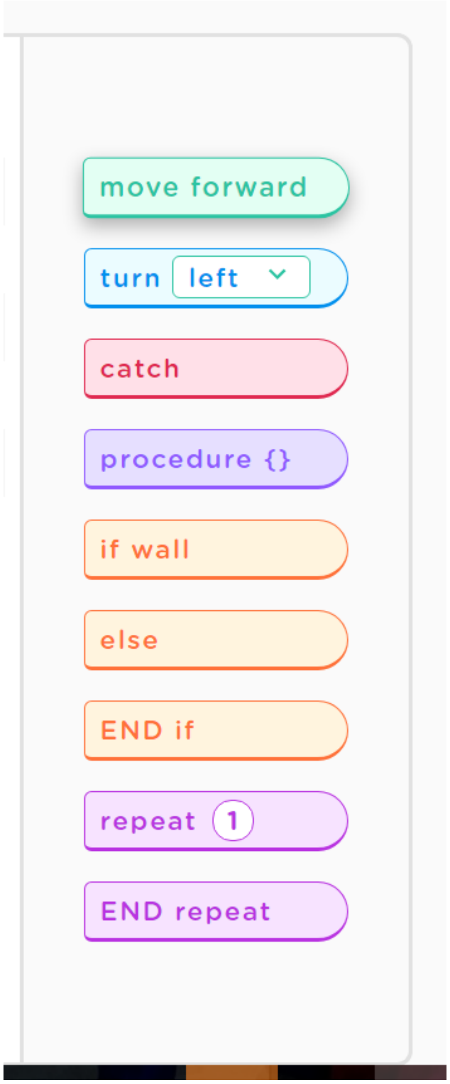


Fig: Code blocks used in this game.

6.7 Maze-like Environment

The maze-like environment is an important aspect of the game that teaches coding concepts using code blocks. It provides players with a fun and engaging way to explore programming concepts and test skills in a challenging setting.

The maze environment is designed to be both visually appealing and functional, with walls as obstacles and other features that require players to think creatively and use their problem-solving skills.

One of the key advantages of the maze environment is that it provides a context for players to apply the programming concepts they are learning in a practical and engaging way. As players navigate the maze, they must use their code blocks to solve the puzzle and overcome the obstacles to reach the Pokémon and catch it in the poké ball. The maze environment provides players with a sense of achievement and progression, as they work their way through increasingly difficult levels and gain new skills and abilities along the way. This sense of progress and achievement is a critical part of the learning process, motivating players to continue playing and developing their programming skills.

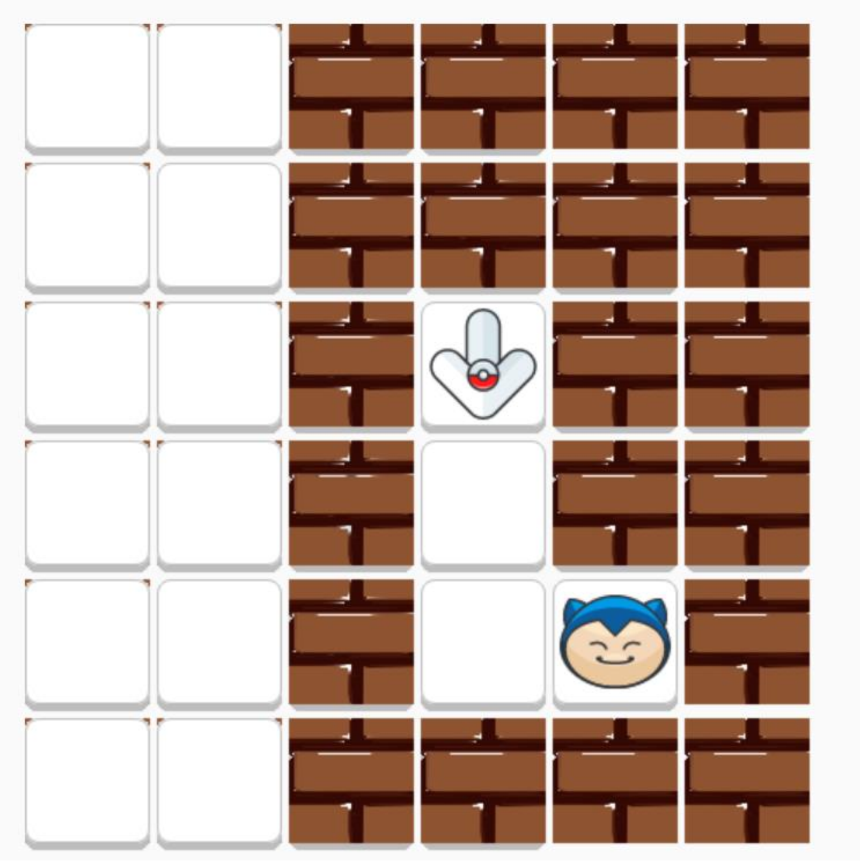


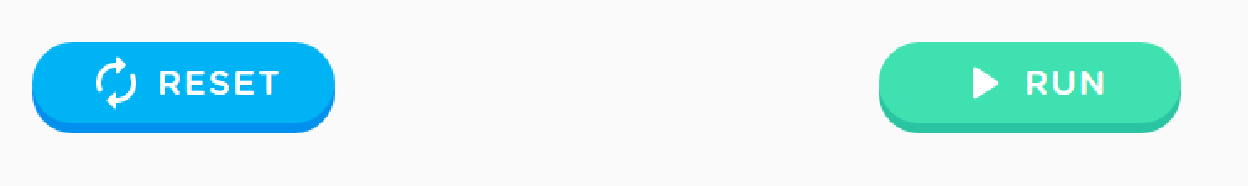
Fig: Maze

6.8 Reset and Run Buttons

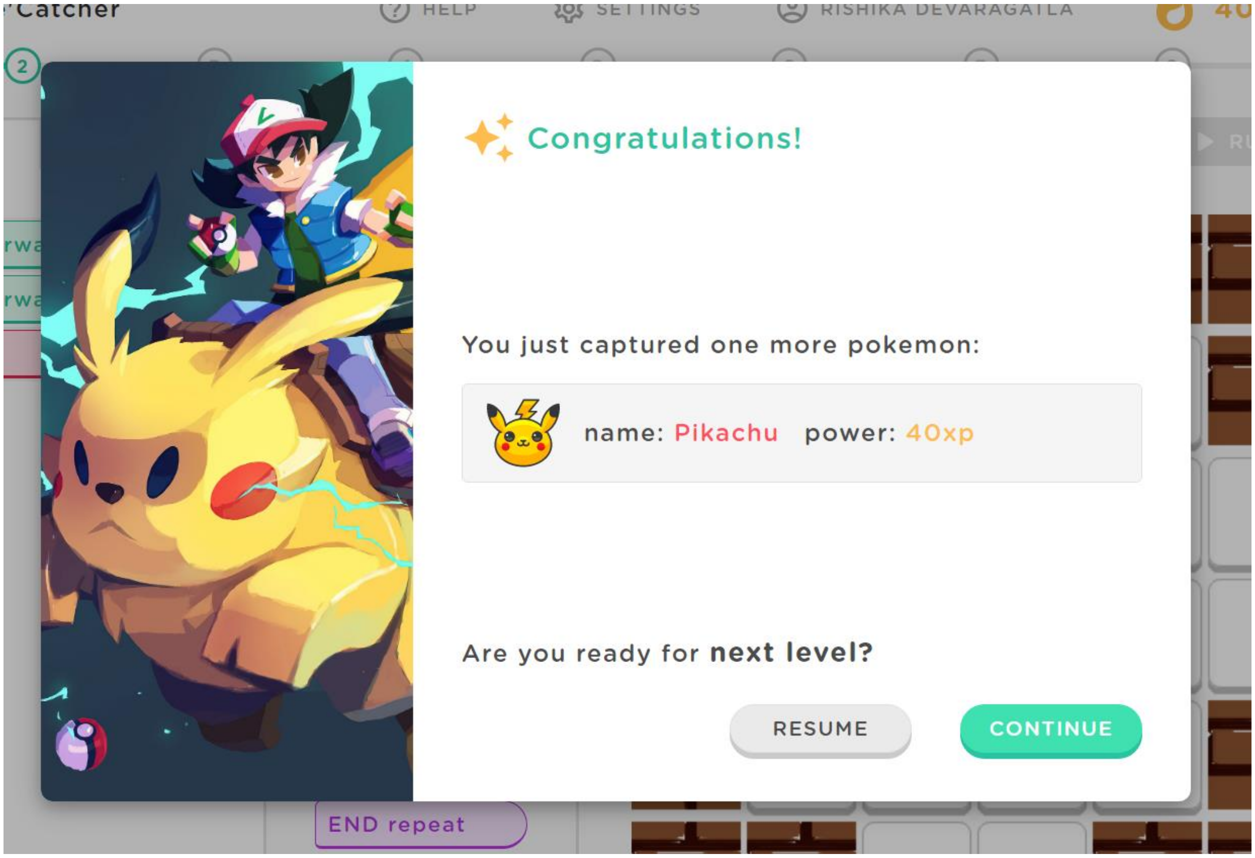
The reset button allows the player to reset the poké ball back to its original state, which is useful if the player makes a mistake or wants to start over.

The run button allows the player to execute the code they have written and see the results in the game environment. It is important that the run button provides immediate feedback to the player, so they can see the effect of their code changes right away.

The reset and run buttons are essential features of the code editor in the game, as they allow the player to experiment with different programming concept and see the results in real-time.



After the player successfully captures the Pokémon after running the program. The player encounters a screen which displays the Pokémon caught, score and two buttons continue and reset.



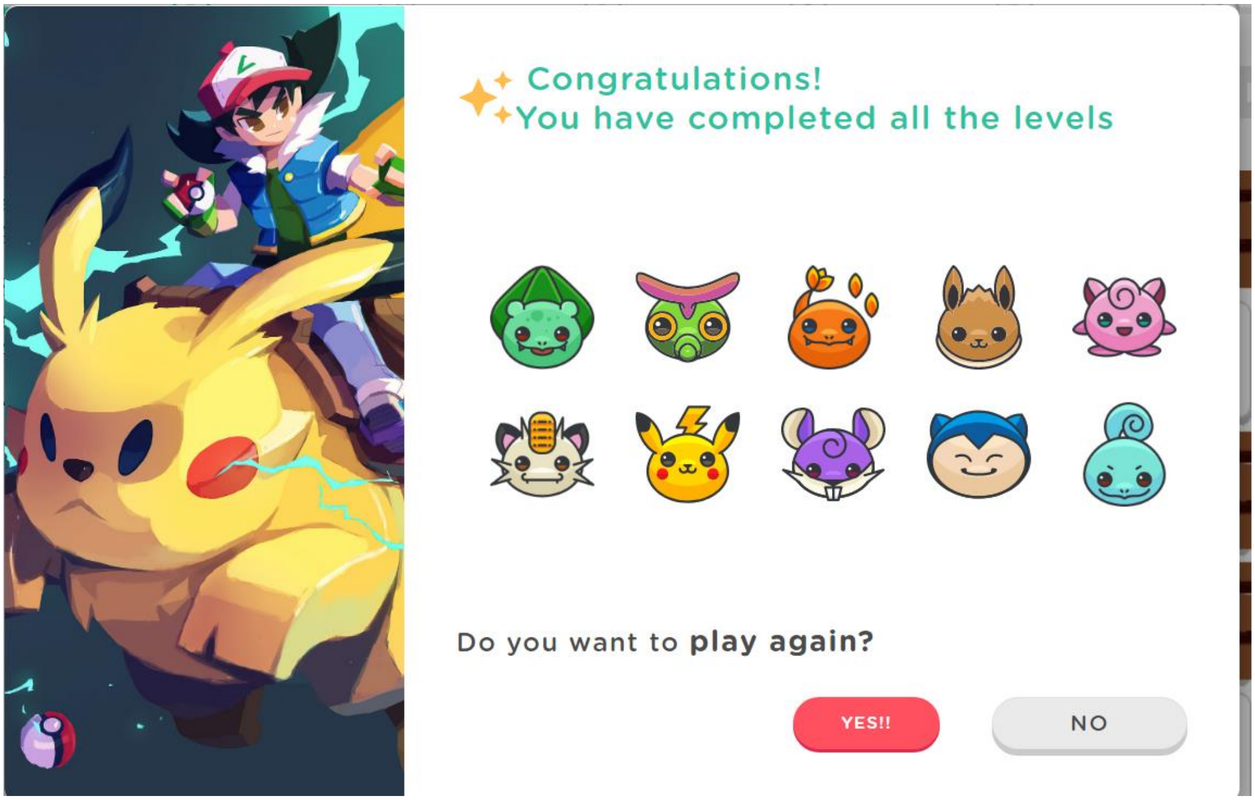
Congratulations!

You just captured one more pokemon:

name: Pikachu power: 40xp

Are you ready for **next level?**

After completing all the 10 levels the player will see an end dialog box. The player can select either to restart the game or to go back to the login page by selecting No.



Congratulations!
You have completed all the levels

Do you want to play again?

6.9 Levels

Levels in the game are the crucial part because they provide players the sense of progress and accomplishment as they advance through the game. Levels are used to gradually introduce reinforce new concepts to the player.

While designing levels of the game I have kept in mind the difficulty curve and pacing the game. The first few levels should be relatively easy to complete, in order to provide players with gentle introduction to the game mechanics and programming concepts. As the players progresses through the levels, the difficulty will gradually increase, with more complex challenges and concepts introduced overtime.

In addition to difficulty, the levels introduce new programming concepts. This will help ensure the players have a thorough understanding of each concept before moving to the next level.

The game is built to have 11 levels

Level 0: Variables and Datatypes

- Introduction to variables and datatypes.
- Exercise to teach the player to identify datatypes by filling out their personal information.

Level 1: Introduction to programming concepts(Part 1)

- Basic introduction to the game mechanics.
- Simple exercise to teach the player how to use code blocks to move the poké ball using the forward and catch.

Level 2: Introduction to programming concepts(Part 2)

- Basic introduction to the game mechanics.
- Simple exercise to teach the player how to use the basic code blocks forward, turn left/right and catch blocks together.

Level 3: Functions (Part 1)

- Introduction to functions.
- Exercise to teach the player how to create and call a function.

Level 4: Loops (Part 1)

- Introduction to loops.
- Exercise to teach the player how to use simple loops to repeat a particular action. E.g., moving the poké ball through a longer maze using a loop instead of repeating the same code block multiple times.

Level 5: Loops (Part 2)

- Continuation of the previous level.
- Exercise to teach player how to use loop iterations.

Level 6: If statements (Part 1)

- Introduction to if-else statements.
- Exercise to teach the player how to use simple if statements.

Level 7: If-else statements combined with Loops (Part 2)

- Exercise to teach the players how to use loops and if-else together in a more complex program.

Level 8: Functions combined with if-else.

- Exercise to teach the players how to use functions and if-else together in a more complex program.

Level 9: Functions combined with loops

- Exercise to teach the players how to use functions and loops together in a more complex program

Level 10: Final Challenge

- Incorporates all the programming concepts learned in previous levels to complete a final challenge.
- Requires the player to solve a complex problem using combination of if-else statements, loops and functions to complete the game.

7. Program Code and files

Main JavaScript code

main.js

Utilities and repeated functions used throughout the program

utilities.js

Login page and the main game environment html code

game.html index.html

CSS Stylesheet

style.css

Various Modules involved are

Drag and drop functionality

drag-and-drop.js

Generate Lines in the Code Editor

generateLines.js

Draw the grid

grid_draw.js

Layout of the grid

grid_layout.js

Code block instruction code

codeBlockInstructions.js

URL of the Pokémon images

pokemonurl.js

8. Future Scope

The game-based learning approach used in this project has potential for future development and expansion.

One possible path for future development is addition of other programming concepts. New concepts are always evolving in the world of programming. The game can be updated to reflect the changes and making it more relevant and effective for future learners.

Another potential area for future development is linking this game with other educational games and programs. The game can be integrated with other online programming courses, textbooks providing a comprehensive learning for young leaners.

This game can also be combined other educational settings like in school activities, after school activities which would help to inspire a new generation of programmers and prepare them for the demands of today’s market.

9. Conclusion

In conclusion, an educational game that teaches programming concepts Cn be a fun and engaging way to introduce beginners to the world of programming. The game will give an immersive experience that keeps players engaged and ready to learn by employing a maze-like environment and adding components from the famous games such as Pokémon.

The levels are carefully designed and distributed to ensure that each one is built upon the previous level and reinforces the concepts learned. Using code blocks, code editor, reset and run buttons, level indicators, and a menu bar with features such profile, settings, and help enhanced the players experience and makes the game more interactive.

Overall, PokéCatcher is an effective tool for introducing beginners to programming in a fun and engaging way.

10. References

- French, J. (2022) *How to write a game design document (with examples), Game Dev Beginner*. Available at: https://gamedevbeginner.com/how-to-write-a-game-design-document-with-examples/ (Accessed: 03 May 2023).
- (No date) *GTA 00 12 - GameDevs.org*. Available at: https://www.gamedevs.org/uploads/grand-theft-auto.pdf (Accessed: 03 May 2023).
- *Game development* (no date) *MDN*. Available at: https://developer.mozilla.org/en-US/docs/Games (Accessed: 03 May 2023).
- *Canvas API - web apis: MDN* (no date) *Web APIs | MDN*. Available at: https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API (Accessed: 03 May 2023).
- *Web apis* (no date) *MDN*. Available at: https://developer.mozilla.org/en-US/docs/Web/API (Accessed: 03 May 2023).