Bahria University

Discovering Knowledge

LEAP FROG

Artificial Intelligence

Abstract

Presenting a simple pathfinding game using the concepts of Artificial intelligence & Data Structures, with the help of Java's Swing GUI.

Group Members

Bilal Ahmed
02-136221-028

Dawood Shahzad
02-136221-034

Hassam Azam
02-136221-035

Muhammad Usman Abubakar
02-136221-031

Table of Contents

Table of Figures

1. Acknowledgment

We would like to express our gratitude to the following people:

Dr. Samabia for her excellect teaching and Miss Komal for looking over us in the Lab, instructing and enabling us to be able to work on this project. And also the team behind the image generation model Stable Diffusion, that we used to generate assets for different game related items, and lastly other classmates such us Ali Hamza for indirectly helping us with the project.

2. Abstract

This project is an AI based platformer game named “Leap Frog”. It was developed using Java’s Swing GUI.

The game features a main menu with options to start the game, view credits and exit the game. Using the WASD and arrow keys, players can control the character to move left, right, up and down. The game includes sound effects for movement and reaching goal. The game environment, including the main menu, buttons, credits screen and level backgrounds, were created by first generating the desired images using Stable Diffusion, and then modifying them as needed.

3. Introduction

The purpose of this project is to develop an engaging and immersive indie game that incorporates pathfinding and optimization algorithms. The game revolves around a frog character that navigates through a series of obstacles and reach the goal within the shortest number of moves. It features a grid-based environment where the character can move in four directions.

It features an 8-bit pixel art environment, sound effects, and animated elements, all designed to enhance the player's experience.

This report will provide a look at the design and implementation of the game, as well as the challenges and successes we encountered throughout the development process.

4. Problems and Solutions

We had to develop a game that challenges players to navigate a frog character through a grid-based environment, overcoming obstacles to reach the goal. The game required implemention of some kind of AI pathfinding algorithm that helps reach the goal in the fewest moves possible. It also needed feature controls, visuals, and sound effects.

Additionally, the game needed animations and visual feedback to enhance the player's experience.

We faced some challenges while developing the game.

One of the challenges we faced was that we wanted to implement a transition when moving from one screen to another. But we didn't have a lot of time to implement it properly so we just created an image in photoshop which was going from black to transparent (gradient), then we exported the image and added it to the game, we implemented it as the transition by moving the image from one area to another using threads for smoother animation, this made it look like the screen was actually fading out to a different scene.
These bits of animation really put life in a game.

Another challenge was setting up the procedural generation of lilypads on the walk-able area in each map, we were confused on how to implement it, and it was easy if we just went for the easy way and hardcoded the positions where the sprites should spawn. But we wanted to handle it professionally instead. A group member received the code of a TicTacToe game which had to be fixed, while fixing the code, he saw something similar being implemented in the code, which could've been the solution to our problem. So we switched back and tried the implementation a second time, which worked very nicely!

The solution was to make a 2D array of lilypads, then use the collision system to determine the safe places in a map and then generate lilypads on those areas, this way, it wasn't hardcoded and we were able to use the lilypad array later for other mechanics in the game like the hint system etc.

There were lots of other small bugs and issues which were fixed by trial and error.

5. Overview of the Project

The project involves the development of an indie game that combines elements of puzzle-solving, pathfinding, and optimization. The game revolves around a frog character navigating through a grid-based environment, aiming to reach the goal with the fewest moves. The game features a nice HUD showing the score and total moves, it also has engaging visuals in an 8-bit pixel art style, and immersive sound effects. The main character is a cybernetic frog that is on its way to escape from evil robots that have invaded his home planet. The frog progress through levels, finding its way out.

The game utilizes AI pathfinding algorithms, such as A* search, to calculate the optimal path for the frog character. A 2D array is employed to represent the game board and track the player's position.

> The A* algorithm is a pathfinding algorithm used in the game to calculate the optimal path for the frog character. It is a heuristic search algorithm that efficiently finds the shortest path between two points on a grid-based map.
>
> The A* algorithm combines the advantages of Dijkstra's algorithm, which guarantees finding the shortest path, and greedy best-first search, which explores promising paths first. It uses heuristics, in our case the Manhattan distance, to estimate the cost from each visited node to the goal.
>
> The algorithm maintains a priority queue, known as the open set, which contains the nodes to be evaluated. It also keeps track of the visited nodes in a closed set. The A* algorithm evaluates the neighboring nodes, calculates their costs using the heuristic function and the actual cost from the start, and selects the node with the lowest total cost as the next step.

The game also incorporates a Constraint Satisfaction Problem (CSP).
This is used for the preocedural generation of lilypads in maps as well as the actual collision system, to move to the next block, certain constraints are evaluated before the action is performed.

The game incorporates a visual D-PAD that dynamically indicates available movement directions. A scoring system is also implemented, rewarding players for completing levels with fewer moves. The game environment features animation effects, created using Java Threads, to enhance visual appeal and immersion.

6. Project Screenshots

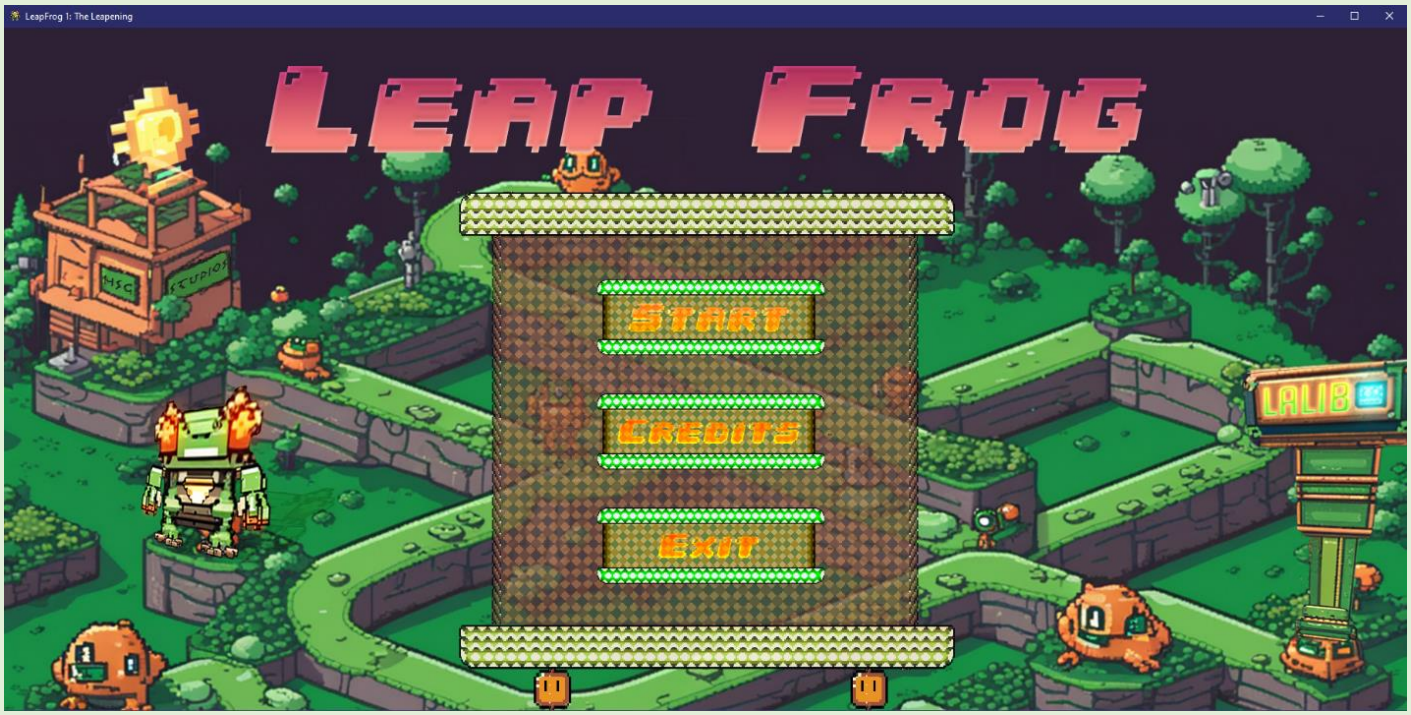

Figure 1: Main Menu screen

Figure 2: Credits screen

Figure 3: Level Select Screen

Figure 4:,Training Level

Figure 5: Level 1, Start area

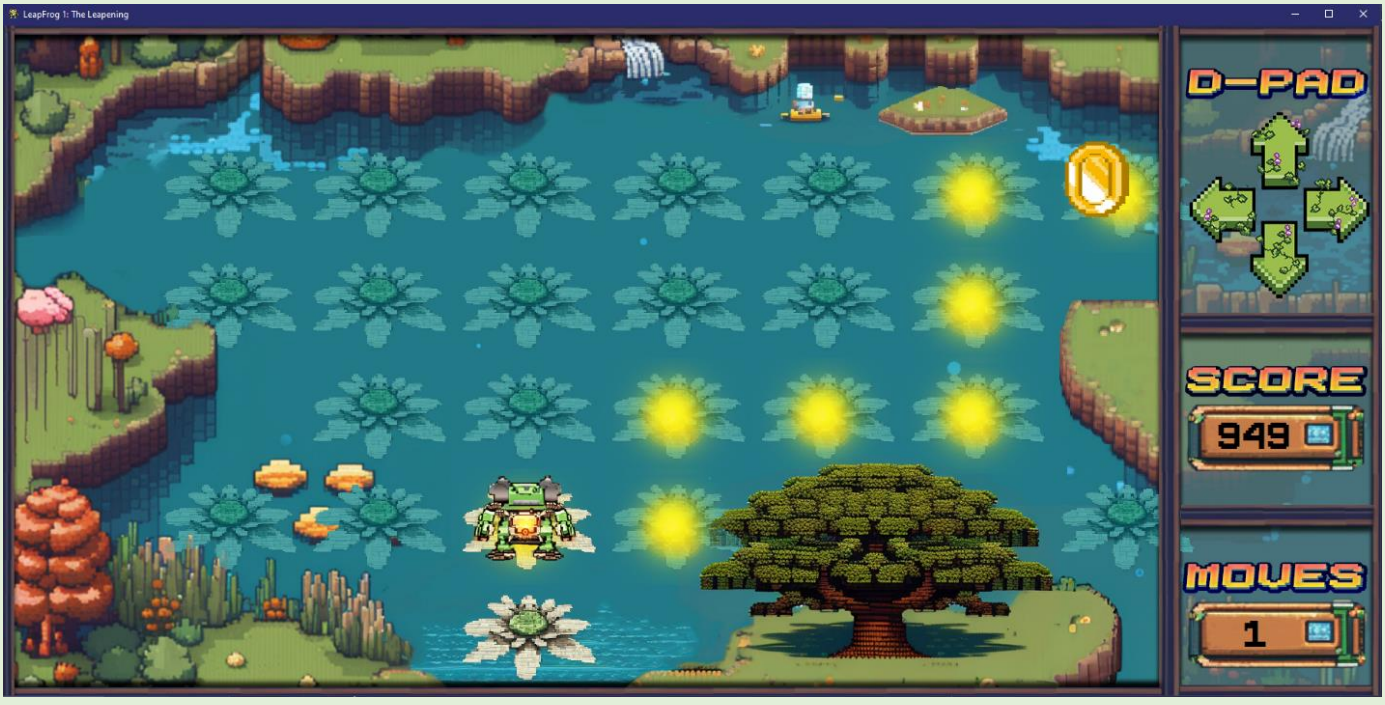

Figure 6: Lervel 1, Pathfinding enabled

Figure 7: Level 2, Starting area

Figure 8: Level 2, Pathfinding enabled

7. Conclusion

Moving forward, there are several areas of potential improvement for the project. This includes, adding more levels, obstacles, enemies, improving graphics and animations. These future developments aim to optimize gameplay, enhance the visuals, and to improve our skills along the way.

In Conclusion, this project improved our programming, problem solving and designing skills and increased our knowledge about game development. The process may have be challenging and difficult, but it was very fun :)

8. References

KaarinGaming | Youtube

Bro Code | Youtube

Stable Diffusion | Discord