UNIVERSIDAD
Panamericana

Alumno:

Yahwthani Morales Gómez

Gabriel Guerra Rosales

Brandon Magaña Ávalos

Gabriel Zaid Gutiérrez González

Trabajo:

Final Project Artificial intelligence on Video Games

Materia:

Artificial Intelligence in Video Games

Profesor:

Alfredo Emmanuel Garcia Falcon

Fecha de entrega:

03/12/2024

FIRE THE MAZE

Game plot and main mission

Fire the maze is a 2D maze game where you play as “Firey” a little fire flame that needs to escape from the maze as soon as possible, Firey is lost and left randomly around the maze some of their flames, he needs to recover their power before escaping while trying to avoid at all cost the enemies that try to turn him off, nevertheless, his not alone, in order to escape he will have to use the power ups randomly spread in the maze. Help fiery to escape the maze following the shortest route.

Description of main characters

The main character is Firey, a little flame that lost all his powers and energy, he is battling versus the water enemies which are in war trying to extinguish all the fire in the world, Firey it's the only hope for the fire of his world, nevertheless this small and weak flame is lost in the maze, the water enemies locked him up in an impossible maze, he is trying to recover all his power and strength searching and recovering all the lost flames which, allow him to escape the maze and running away from the enemies.

Description of enemies

- Water drop: The water drop will have a behaviour of wander around the maze, when the player enters the range and there's no barrier between the water drop and the player the enemy will start to pursue the player until it intercepts it.
- Ice: The ice makes a wander as the water drop and detect when the player is near, in that case the ice changes the behaviour to a pursue as the water drop, but, this enemy has a shorter detection range
- Extinguisher: The extinguisher makes a wander as the water drop and ice detecting when the player is near, in that case the ice changes the behaviour to a pursue as the water drop, but, this enemy has a widder detection range

All the enemies will change their original behaviour if the player grabs the blue fire flame, this power up makes the enemies change their behaviour to a evade, where they are going to escape from the user.

Scene's description

- Menu Scene: Is the first scene displayed in the game, it shows the main options of the game, contains the button to initialize the game and a button where the instructions and objective of the game are descriptive
- Maze Scene: It's the main scene where the game is displayed, in this window you can observe the maze creation with a black background and white roads where the player can walk, in this scene you make the movements of the player, displaying the enemies and main character assets randomly displayed in the maze, The optimal route is displayed in the maze as a orange color, the end of the maze is represented as green and all this maze is divided as a grid.
- Victory/Loss Scene: This scene is displayed once the player achieves the goal or once the main character dies, immediately this window is displayed and the game is paused, also, it has the option to return to the main menu and restart the game

Description of the game mechanics or mechanics

- Player movement: the character moves using W to move forward, D to move to the right S to move backwards and A to move left, this basic controllers are focused on the navigation of the player in the possible movements within the maze
- Fire flames collection: The player only needs to arrive to the position where the flame is and doing that the fire is gonna disappear and it will count as a fire collected
- Fire power ups: The maze has a high complexity, so the player will have power ups, represented as green flames, that help firey, to eliminate the enemies while he grabs the green flame, after grabbing this power, you will have 5 seconds of immunity and the ability to eliminate the enemies
- Enemie Avoiding: The player needs to navigate in the maze and try to leave behind the enemies by escaping from them

- The game ends once all the flames are collected and the player arrives to the end following the optimal route. Also, the game ends when the user is reached by an enemy.

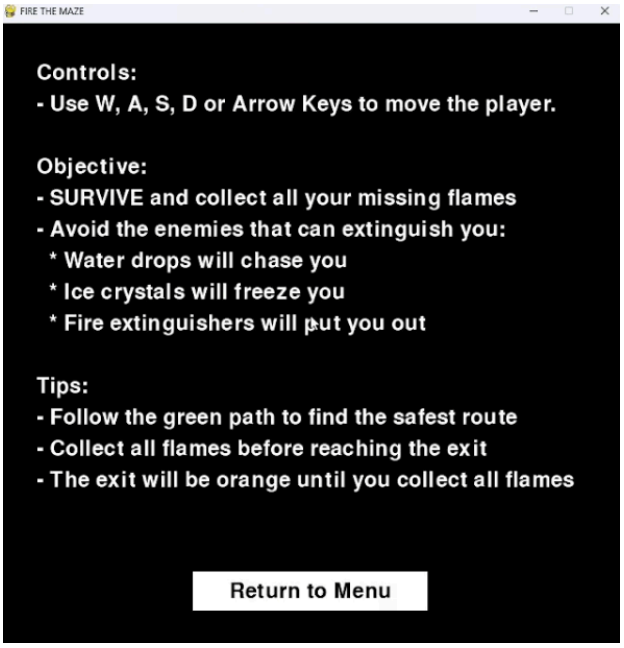
Description of the algorithms used in the game

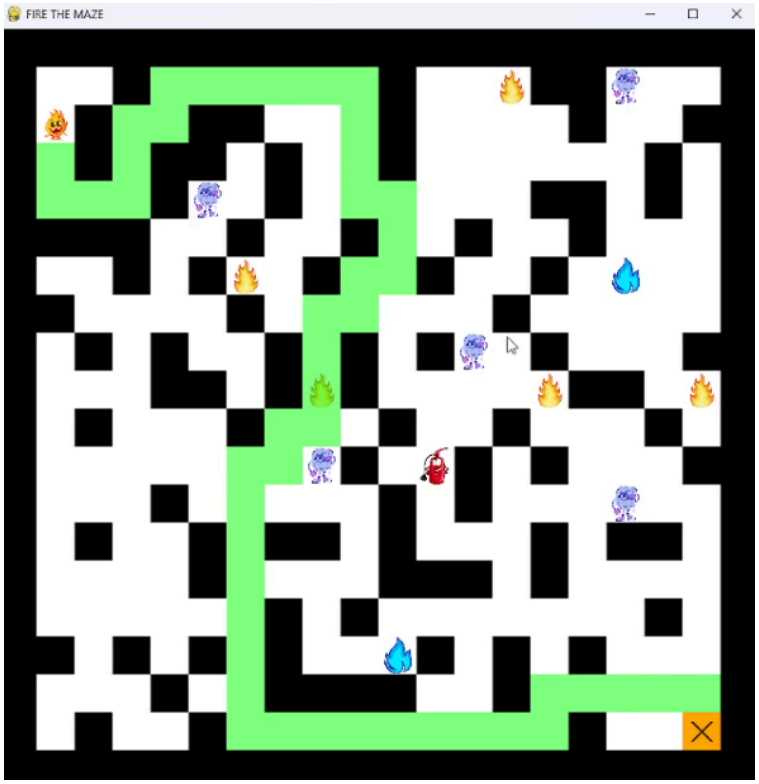
- Maze generation: Creates a initial grid making the subdivision initially as walls, then using a randomize recursive backtracking we start creating the random maze, once is created we open the paths depending on the specified density finally we validate the number of neighbours of the cells to confirm there are valid paths.No
- Dijkstra Algorithm: In this case dijkstra is used to calculate the shortest path from the beginning of the maze to the final node of it. First of all we set the distance to all nodes as infinity, except the start node, which is set to 0 and this node is added to the priority queue initialized. We will store the previous nodes of each visited node to reconstruct the path, then, in our queue we will pop the smallest distance and retrieve its position, if that position is the end node the game ends. Then, for each neighbour we check if is within the grid bounds and not blocked, so, this will add 1 to our current distance, if the new distance is shorter than the current distance we push the neighbor into the priority queue with the new distance, indicate the current node as the last of the neighbor in the reconstruction and we start iterating this process, once the process is over we go back from the end to the start using our priority queue to reconstruct the shortest path.
- Steering Behaviours:
    - Wander: The enemy is going to do a random movement, we project a point to a position and with a random offset we make the enemy move in those random positions inside the offset
    - Pursuit: we predict the future position of a moving target based on its velocity, this algorithm calculates a steering force in direction of the predicted position

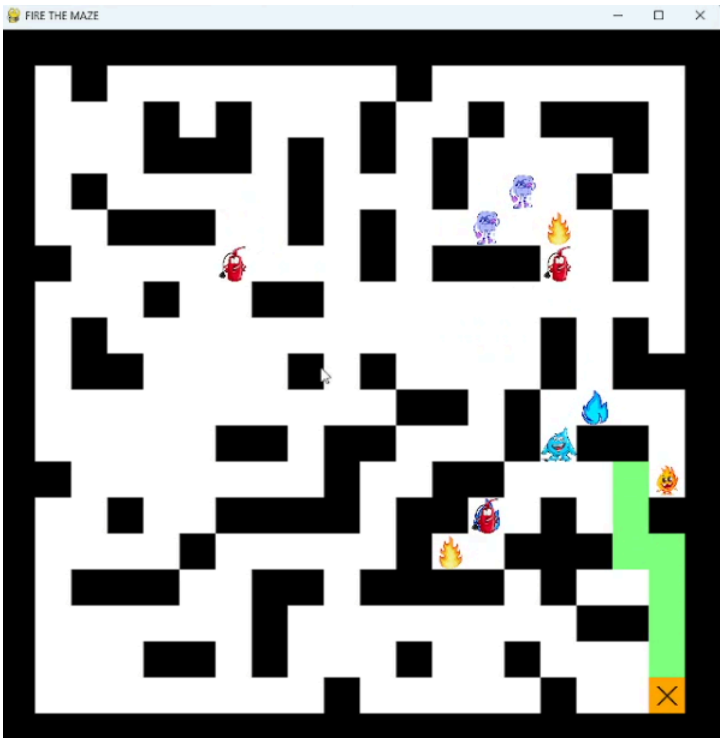
    - Evade: This behaviour predicts the future position of a moving target, similar to pursue. This algorithm calculates a steering force to move away from the predicted position

- Player movement: We update the player's position depending on the key inputs (WASD) and ensures movement only occurs on open cells
- Fire collection: The Fire class displays multiple fire tiles within the maze, and checks if the player is on the till, if that's the case, we add the tile to the array.

In game images








FIRE THE MAZE


FIRE THE MAZE


FIRE THE MAZE
GAME OVER
¡El extintor te ha apagado!
Return to Menu


FIRE THE MAZE
VICTORY!
You saved the fire!
Return to Menu

Conclusion of each of the team members

Gabriel Guerra:

In conclusion, i enjoyed a lot the methodology and final knowledge achieved in class, i think it gave me not only knowledge of steering behaviours, states machines or path finding algorithms, this course gave me tools were i could implement the logic and topics of the class, i really liked the fact that we could chose our project to developed, because also motivates to do something we like and improves creativity, also, the development of the topics was great because as we did most of the algorithms in other projects and exercises in class so it was easier to implement that logic in a final game, that's something i want to congratulate the teacher and just make emphasis on how much i enjoyed this course.

Brandon Magaña:

In the development of “Fire The Maze”, I learned the importance of implementing artificial intelligence algorithms in video games. The procedural generation of the maze and the implementation of Dijkstra's algorithm helped me better understand how to create dynamic and unique experiences for each game. The project also allowed me to understand how steering behaviors can create more realistic and challenging enemies, significantly improving the player experience.

Yahwthani Morales:

The development of this project allowed me to apply in a practical way the concepts learned during the course. The implementation of different enemy behaviors through steering behaviors was an interesting challenge that helped me to better understand how to create more dynamic and realistic AI. It was also rewarding to see how the combination of different algorithms (Dijkstra and steering behaviors) can create a cohesive and entertaining game experience.

Zaid Gutierrez:

This project was an excellent opportunity to explore the practical implementation of AI algorithms in video games. Creating different types of enemies with unique behaviors helped me to better understand how AI can make a game more interesting and challenging. Implementing power-ups and their effects on enemy behavior also showed me how game systems can interact dynamically to create richer and more varied experiences.