**Project Description**

A simple web game that simulates a bear trying to gather enough food (honey) allowing it to go into hibernation. Bear’s movement is controlled by the four arrow keys. The bear has to avoid getting stung by the bees after raiding their beehive for honey. To do that the bear needs to create a 3-tile distance between itself and the bees. The bear starts with half of its health bar filled already. To successfully complete the game the bear has to fully fill the health bar. If the bear health goes below 0, its game over.

**How to Run**

* Need to have node.js installed on the computer
* Download the “Source Code and Presentation Video.zip” file
* Inside that file find and unzip the Project.zip file
* Please navigate to the unzipped Project directory and run the following commands in order-

npm install --save three

npm install --save-dev vite

npx vite

**Game Controls**

Up Key – move up

Down Key – move down

Left Key – move left

Right Key – move right

Up + Left Key – move towards top left diagonally

Up + Right Key – move towards top right diagonally

Down + Left Key – move towards bottom left diagonally

Down + Left Key – move towards bottom right diagonally

**Categories and Implementation**

|  |  |  |
| --- | --- | --- |
| **Category** | **Concept Chosen** | **Implementation** |
| Complex Movement Algorithms | Flocking | Once player touches the beehive and runs away from the beehive, the bees will start following the bear and show flocking behavior  Files – Bees.js |
| Pathfinding | Flow Field Pathfinding | The flow field vectors are updated whenever a player moves and the bees use those vectors to find the player.  Files – Player.js (Bear), gameMap.js |
| Decision Making | State Machine | For bees the states are InitialState, ChasingState, and TransitonState  For bear the states are IdleState and MovingState  Files – State.js, Bees.js and Player.js |
| Procedural Content Generation | Depth-First Backtracking Maze Generation | Creating the maze for the game.  Files – MazeGenerator.js and Maprenderer.js |
| Other Topic | Halton Sequence (Random Number Generator) | Used to set location of new beehive after the bear empties the previous beehive. Also used to set location for the initial beehives.  Files – GameMap.js, Main.js, Bees.js |

**Members and Contributions**

**S M Ridwanul Haque (202064317):**

1. Creation of the maze
2. Implementing Game Mechanics for Bear (Player)
3. Updating flow vectors for player movement
4. Beehive and Bee spawning using Halton sequence
5. Using State Machine to handle bee and bear behavior

**Md Golam Mahmud Chowdhury (202164141):**

1. Creating health bar and handling health update
2. Collecting models and assigning appropriate alignment
3. Handling Pop-ups for the game.
4. Integrating the flocking behavior for the bees.
5. Adjusting maze boundaries and other UI modifications