REPORT-FutureFix

Game Idea:

API City is an educational game designed to teach players how API requests (GET, POST, PUT, PATCH, DELETE) work in a real-world simulation. Players will explore a virtual city, complete missions using API calls, and learn about authentication, rate limiting, and error handling.

Features:

Interactive City Environment

- Players explore a vibrant, interactive city where every building and system is tied to an API.

Mission-based Learning

- Players are tasked with completing missions using different HTTP methods, like GET, POST, PUT, and DELETE.

Simulation of API Calls

- Players perform API requests to interact with the virtual environment, learning through hands-on experience.

Dialogue Box

- Continuous messages are displayed for player dynamically in each scene.

Progressive Difficulty

- The complexity of the tasks increases as the player progresses through different scenes, simulating real-world use cases of API calls.

Scenes implemented in the game play:

Scene 1: Login (POST Method)

Objective: Players need to use the POST method to authenticate and log in.

Gameplay:

- A pop-up interface appears with the four HTTP methods: **GET**, **POST**, **PUT**, **DELETE**.
- The player selects **POST** to authenticate and log in to the game.
- If the player selects any other method (e.g., **GET**), an error message appears.

Learning Outcome:Understand how to use the **POST** method to create or authenticate a Resource.

Scene 2:Weather Information (GET Method)

Objective: Use the **GET** method to retrieve weather information for the city.
Gameplay:

- A pop-up interface appears with the four HTTP methods: **GET**, **POST**, **PUT**, **DELETE**.
- The player must select **GET** to retrieve weather data.
- If the player selects any other method (e.g., **GET**), an error message appears, guiding the player to choose the appropriate method.

Learning Outcome:Understand how the **GET** method is used to retrieve data from an API.

Scene 3: Get City Map (GET Method)

Objective: Use the **GET** method to retrieve the map of the city
Gameplay:

- The player enters a scene where they are in a city environment. The task is to view the city map, which includes various locations such as buildings, streets, parks, and other notable areas.
- A pop-up interface appears with the four HTTP methods: **GET**, **POST**, **PUT**, **DELETE**.
- The player is tasked with selecting the **GET** method to retrieve the city map.
- If the player selects any other method (e.g., **GET**), an error message appears, guiding the player to choose the appropriate method.

Learning Outcome:Understand how the **GET** method is used to retrieve data from an API.

Scene 4: Building Information (GET Method)

Objective: Retrieve detailed information about a building using the **GET** method.
Gameplay:

- A pop-up interface appears with the four HTTP methods: **GET**, **POST**, **PUT**, **DELETE**.
- The player selects **GET** to get information about a selected building.
- If the player selects any other method (e.g., **GET**), an error message appears.

Learning Outcome:Learn to use the **GET** method to query resources and retrieve information from an API.

Scene 5: Supermarket Inventory (PUT Method)

Objective: Players need to use the POST method to authenticate and log in.
Gameplay:

- A pop-up interface appears with the four HTTP methods: **GET**, **POST**, **PUT**, **DELETE**.
- The player must select **PUT** to replace or update an item in the inventory.
- If the player selects any other method (e.g., **GET**), an error message appears.

Learning Outcome:Understand how the **PUT** method is used to fully update or replace a Resource.

Scene 6: Garbage System (DELETE Method)

Objective: Use the **DELETE** method to remove garbage from a specific location in the city.

Gameplay:

- The player enters a city area where garbage is scattered around.
- A pop-up interface appears with the four HTTP methods: **GET**, **POST**, **PUT**, **DELETE**.
- The player is tasked with selecting the **DELETE** method to remove a specific garbage item from the city.
- If the player selects any other method (e.g., **GET**), an error message appears.
- The player can click on a garbage item, and after selecting **DELETE**, the item will be removed from the map, simulating the process of cleaning the city.

Learning Outcome:Understand how the **PUT** method is used to fully update or replace a Resource.

Scene 7: Accident Reporting (POST Method)

Objective: Use the **POST** method to report an accident to the hospital and request an ambulance.

Gameplay:

- The player comes across an accident scene where a person is lying injured after being hit by a car.
- A pop-up interface with the four HTTP methods (**GET, POST, PUT, DELETE**) appears, prompting the player to select **POST** to send the accident report to the hospital.
- If the player selects the wrong method, an error message guides them to choose the correct method (**POST**).
- Once the player selects POST, the request is processed, and an ambulance arrives to assist the injured person, completing the task.

Learning Outcome:Understand how the **PUT** method is used to fully update or replace a Resource.

Contributions:

CS22B003 - Akshatha RH: Skybox, Traffic System, MeshUtility Script, Build Materials Scene.

CS22B006 - A Preethika: Objects, Scenes, MeshSize Script, Main Page Scene, WayPoint Marker.

CS22B018 - CH Aarya: Roads, Editor, TrafficSystem Script, Ambulance Scene, Map Scene, MiniMap

CS22B029 - K Sanjay Varshith: Waytool, Shaders, CityGenerator Script, Login Scene, DialogueBox.

CS22B032 - K Akhil Solomn: Backgrounds, Water, TrafficCar Script, Garbage Scene, SuperMart Scene

CS22B037 - M Akash: Buildings, Traffic Light, TrafficLight Script, Dynamic weather api and its DialogueBox