**Project Phase One Report**

**CS 3354 Fall 2024**

**Due by:** 10/06/2024 (Sunday) 11:59 PM (EOD) via eLearning

**Project Title: Emgato**

**Team Members:**

* A

**Individual Contributions Breakdown**

All team members contributed equally to this report, participating in discussions, research, and writing of the various sections.

**Table of Contents**

1. **Customer Problem Statement**
   * a. Problem Statement
   * b. Decomposition into Sub-problems
   * c. Glossary of Terms
2. **Goals, Requirements, and Analysis**
   * a. Business Goals
   * b. Enumerated Functional Requirements
   * c. Enumerated Nonfunctional Requirements
   * d. User Interface Requirements
3. **Use Cases**
   * a. Stakeholders
   * b. Actors and Goals
   * c. Use Cases
     + i. Casual Description
     + ii. Use Case Diagram
     + iii. Traceability Matrix
     + iv. Fully-Dressed Description
   * d. System Sequence Diagrams
4. **User Interface Specification**
   * a. Preliminary Design
   * b. User Effort Estimation
5. **System Architecture**
   * a. Identifying Subsystems
   * b. Mapping Subsystems to Hardware
   * c. Connectors and Network Protocols
   * d. Global Control Flow
   * e. Hardware Requirements
6. **Plan of Work**
7. **References**

**1. Customer Problem Statement**

**a. Problem Statement**

In today's increasingly isolated world, especially in a post-COVID environment, students often find it challenging to connect with others and engage in campus life. Many are experiencing living alone for the first time and may struggle to find a sense of community, particularly at universities like UTD where social spaces are limited.

To address this issue, we propose "Emgato," a 2D side-scrolling game that encourages students to explore the UTD campus and interact with others. The game features a cat navigating through levels, collecting trinkets, and discovering fun facts about campus locations. By integrating geolocation services, the game unlocks new levels when players visit specific areas on campus. Periodic event levels will further motivate students to get out of their dorms, participate in campus activities, and meet new people with shared interests.

**b. Decomposition into Sub-problems**

1. **Game Development**
   * Implement core gameplay mechanics (movement, jumping, collecting items).
   * Design engaging levels with varying difficulty and aesthetics.
2. **Geolocation Integration**
   * Utilize geolocation services to determine player location.
   * Unlock location-specific content based on proximity to campus landmarks.
3. **Multiplayer Functionality**
   * Develop real-time multiplayer capabilities for collaborative or competitive play.
   * Implement a chat feature for in-game communication.
4. **Community Features**
   * Create leaderboards to display top players.
   * Allow players to add friends and view their progress.
5. **User Interface and Experience**
   * Design an intuitive UI with pixel art and dynamic color schemes.
   * Ensure seamless navigation through menus and game screens.
6. **Backend Services**
   * Set up servers to handle game logic, user data, and real-time interactions.
   * Establish databases for storing user accounts, inventories, and scores.
7. **Security and Privacy**
   * Implement robust security measures to protect user data and geolocation information.
   * Ensure compliance with relevant data protection regulations.
8. **Testing and Deployment**
   * Conduct thorough testing to identify and fix bugs.
   * Optimize performance for a smooth user experience.

**c. Glossary of Terms**

* **Pixel**: In this context, a pixel refers to a unit of space in the game's visual environment, representing a portion of the game world.
* **Color Palette**: A limited set of colors used consistently in game art to maintain aesthetic cohesion.
* **Layer**: Different visual depths in the game scene (background, midground, foreground) that create a parallax effect.
* **Scene**: The visual area displayed to the player at any given moment.
* **Level**: A distinct stage or environment within the game with specific objectives.
* **Geolocation Services**: Technology that uses GPS data to determine a device's physical location.
* **Trinket**: Collectible items within the game that players can add to their inventory.
* **Leaderboard**: A ranking system displaying players' standings based on certain criteria.
* **Multiplayer Mode**: A game mode where multiple players can interact in the same game environment simultaneously.
* **Parallax Effect**: A visual effect where background layers move slower than foreground layers, creating an illusion of depth.

**2. Goals, Requirements, and Analysis**

**a. Business Goals**

1. **BG-1**: Encourage students to explore the UTD campus.
2. **BG-2**: Foster community engagement through interactive gameplay.
3. **BG-3**: Provide an enjoyable and motivating gaming experience.
4. **BG-4**: Educate players about campus landmarks with fun facts.
5. **BG-5**: Ensure user data privacy and security.
6. **BG-6**: Achieve scalability to support multiple concurrent users.

**Hierarchy of Goals**

* **BG-1** and **BG-2** support the overarching goal of enhancing student engagement and socialization.
* **BG-3** and **BG-4** aim to maintain user interest and provide educational value.
* **BG-5** and **BG-6** are essential for building trust and ensuring the platform's reliability.

**b. Enumerated Functional Requirements**

| **Requirement ID** | **Priority** | **Description** |
| --- | --- | --- |
| **REQ-1** | High | Implement cat character control in a 2D side-scrolling environment. |
| **REQ-2** | High | Integrate geolocation to unlock location-specific levels on campus. |
| **REQ-3** | Medium | Provide base levels accessible from any location. |
| **REQ-4** | High | Enable real-time multiplayer with up to X players. |
| **REQ-5** | Medium | Implement in-game chat for player communication. |
| **REQ-6** | High | Allow players to collect trinkets and add them to their inventory. |
| **REQ-7** | Medium | Display leaderboards showing top-ranking players. |
| **REQ-8** | Medium | Implement user accounts with email verification for password resets. |
| **REQ-9** | Low | Allow players to add friends and view their progress. |
| **REQ-10** | Medium | Include optional multiplayer modes (racing, collaborative). |

**c. Enumerated Nonfunctional Requirements**

| **Requirement ID** | **Priority** | **Description** |
| --- | --- | --- |
| **REQ-11** | High | Secure user data and geolocation information per privacy regulations. |
| **REQ-12** | Medium | Provide an intuitive UI with pixel art and dynamic color schemes. |
| **REQ-13** | High | Ensure scalability for multiple concurrent users without performance loss. |
| **REQ-14** | Medium | Maintain accurate geolocation within reasonable tolerance levels. |
| **REQ-15** | Low | Offer cross-platform compatibility via web browsers. |

**d. User Interface Requirements**

| **Requirement ID** | **Priority** | **Description** |
| --- | --- | --- |
| **REQ-16** | High | Display game environments with layers for parallax depth perception. |
| **REQ-17** | Medium | Include a campus map with event and location markers. |
| **REQ-18** | Medium | Provide an inventory screen for collected trinkets. |
| **REQ-19** | Low | Show notifications for event-style maps and special levels. |
| **REQ-20** | Medium | Allow interaction with leaderboards and friend lists. |

**3. Use Cases**

**a. Stakeholders**

* **Students**: Primary users seeking exploration and social interaction.
* **University Administration**: Interested in promoting campus engagement.
* **Game Developers**: Responsible for development and maintenance.
* **Event Organizers**: Use the game to promote events.
* **Privacy Regulators**: Ensure compliance with data protection laws.

**b. Actors and Goals**

* **Player (Initiating Actor)**: Wants to play, explore, collect trinkets, and interact with others.
* **Server System (Participating Actor)**: Manages game logic and data.
* **Event Coordinator (Initiating Actor)**: Creates and manages event-specific levels.

**c. Use Cases**

**i. Casual Description**

* **UC-1: Play Base Level**
  + Players can select and play base levels from any location.
  + Responds to **REQ-1**, **REQ-3**.
* **UC-2: Unlock Location-Specific Level**
  + Players unlock levels by visiting specific campus locations.
  + Responds to **REQ-2**, **REQ-6**.
* **UC-3: Collect Trinket**
  + Players collect trinkets within levels to add to their inventory.
  + Responds to **REQ-6**.
* **UC-4: View Leaderboard**
  + Players view leaderboards to see top rankings.
  + Responds to **REQ-7**.
* **UC-5: Multiplayer Interaction**
  + Players interact and communicate in real-time within the same map.
  + Responds to **REQ-4**, **REQ-5**.
* **UC-6: Account Management**
  + Players create accounts and log in to access features.
  + Responds to **REQ-8**.
* **UC-7: Add Friend**
  + Players add friends to view their progress.
  + Responds to **REQ-9**.

**ii. Use Case Diagram**

*(Diagram not included in this text format.)*

**iii. Traceability Matrix**

| **Requirement ID** | **Use Case ID(s)** | **Priority** |
| --- | --- | --- |
| **REQ-1** | UC-1 | High |
| **REQ-2** | UC-2 | High |
| **REQ-3** | UC-1 | Medium |
| **REQ-4** | UC-5 | High |
| **REQ-5** | UC-5 | Medium |
| **REQ-6** | UC-2, UC-3 | High |
| **REQ-7** | UC-4 | Medium |
| **REQ-8** | UC-6 | Medium |
| **REQ-9** | UC-7 | Low |
| **REQ-10** | UC-5 | Medium |

**iv. Fully-Dressed Description**

**Use Case UC-2: Unlock Location-Specific Level**

* **Primary Actor**: Player
* **Goal**: Unlock and play levels specific to the player's current campus location.
* **Preconditions**:
  + Player is logged in.
  + Geolocation services are enabled.
* **Postconditions**:
  + Location-specific level is unlocked and accessible.
* **Main Success Scenario**:
  + Player selects "Location Levels" from the menu.
  + System requests geolocation access.
  + Player grants permission.
  + System identifies current location.
  + System checks for levels associated with location.
  + Level is unlocked and displayed.
  + Player selects and plays the level.
* **Extensions**:
  + *3a. Player denies geolocation access*:
    - System informs the player that geolocation is required.
    - Use case ends.
* **Special Requirements**:
  + Accurate geolocation within reasonable tolerance.
  + Compliance with privacy regulations.

**Use Case UC-3: Collect Trinket**

* **Primary Actor**: Player
* **Goal**: Collect trinkets in-game to add to inventory.
* **Preconditions**:
  + Player is in a game level.
* **Postconditions**:
  + Trinket is added to inventory.
* **Main Success Scenario**:
  + Player navigates through the level.
  + Player encounters a trinket.
  + Player collects the trinket.
  + System updates the inventory.
  + Fun fact about the trinket is displayed.
* **Extensions**:
  + *4a. Inventory is full*:
    - System notifies the player.
    - Player cannot collect more until space is freed.
* **Special Requirements**:
  + Immediate feedback upon collection.
  + Relevant and informative fun facts.

**d. System Sequence Diagrams**

*(Diagrams not included in this text format.)*

**4. User Interface Specification**

**a. Preliminary Design**

**UC-2: Unlock Location-Specific Level**

* **Step 1**: Main menu with options like "Play Base Level," "Location Levels," etc.
* **Step 2**: Player selects "Location Levels."
* **Step 3**: System prompts for geolocation access.
* **Step 4**: Upon permission, system displays available levels.
* **Step 5**: Player selects a level to play.
* **Step 6**: Game loads with the cat character in the new level.

**UC-3: Collect Trinket**

* **Step 1**: In-game interface shows the game environment.
* **Step 2**: Trinkets are visibly placed in the level.
* **Step 3**: Player moves towards and collects the trinket.
* **Step 4**: Notification appears confirming collection.
* **Step 5**: Inventory updates with the new trinket.
* **Step 6**: A fun fact is displayed.

*(Mock-up screens would be included in the full report.)*

**b. User Effort Estimation**

**Scenario**: Unlocking a location-specific level and collecting a trinket.

* **Menu Navigation**:
  + Click "Location Levels": 1 click
  + Grant geolocation access: 1 click
  + Select level: 1 click
* **In-Game Actions**:
  + Movement keys pressed: ~20 keystrokes
  + Jump actions: ~5 keystrokes
  + Collect trinket: 1 keystroke
* **Total Actions**: ~28 actions
  + **UI Navigation**: 3 clicks (approx. 11%)
  + **Gameplay Actions**: 25 keystrokes (approx. 89%)

**5. System Architecture**

**a. Identifying Subsystems**

* **Frontend Subsystem**
  + Handles UI, game rendering, user input.
* **Backend Subsystem**
  + Manages game logic, databases, authentication.
* **Geolocation Subsystem**
  + Integrates device GPS data.
* **Multiplayer Subsystem**
  + Manages real-time interactions.
* **Security Subsystem**
  + Ensures data protection and compliance.
* **Database Subsystem**
  + Stores user data, inventories, scores.

*(UML package diagram would be included.)*

**b. Mapping Subsystems to Hardware**

* **Client Devices (User's device)**
  + Frontend and Geolocation Subsystems.
* **Server Machines (Cloud servers)**
  + Backend, Multiplayer, Security, Database Subsystems.

**c. Connectors and Network Protocols**

* **Client-Server Communication**
  + **HTTPS**: For secure data transmission.
  + **WebSockets (Socket.io)**: For real-time multiplayer interactions.
* **Database Communication**
  + **MongoDB/PostgreSQL Protocols**: For database interactions.
* **Geolocation Data**
  + **HTML5 Geolocation API**: Obtains location data.

**Rationale**

* **HTTPS** ensures secure communication.
* **WebSockets** provide low-latency interactions essential for multiplayer.
* **Databases** chosen for scalability and flexibility.

**d. Global Control Flow**

* **Execution Orderness**
  + Event-driven system responding to user actions.
  + Non-linear gameplay allows actions in various orders.
* **Time Dependency**
  + Real-time data for multiplayer and geolocation.
  + Timers for events or challenges may be implemented.

**e. Hardware Requirements**

* **Client Devices**
  + **Display**: Minimum resolution of 640 × 480 pixels.
  + **Browser**: Latest versions of major browsers.
  + **Network**: Minimum 56 Kbps bandwidth.
* **Server Infrastructure**
  + **Scalability**: To handle concurrent users.
  + **Storage**: Adequate for user data and assets.
  + **Network**: High bandwidth for data transmission.

**6. Plan of Work**

**Timeline and Milestones**

* **Weeks 1-2**
  + Finalize requirements and design.
  + Set up development environments.
* **Weeks 3-4**
  + Develop core game mechanics.
  + Create initial level designs.
* **Weeks 5-6**
  + Integrate geolocation services.
  + Implement location-based unlocking.
* **Weeks 7-8**
  + Set up user authentication and databases.
  + Develop account management features.
* **Weeks 9-10**
  + Implement multiplayer functionalities.
  + Add chat feature.
* **Weeks 11-12**
  + Design and refine UI elements.
  + Conduct UI testing.
* **Weeks 13-14**
  + Enhance security measures.
  + Perform load and performance testing.
* **Week 15**
  + Final testing and bug fixes.
  + Prepare for presentation/demo.

**Responsibility Allocation**

* **Team Member A**
  + Frontend development and UI design.
* **Team Member B**
  + Backend development and database management.
* **Team Member C**
  + Geolocation integration and security protocols.
* **Team Member D**
  + Multiplayer features and real-time communication.
* **Team Member E**
  + Testing, QA, and project coordination.

**Management and Coordination**

* Weekly meetings for progress updates.
* Use project management tools for task tracking.
* Utilize version control systems for collaboration.

**7. References**

* **HTML5 Geolocation API**: <https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API>
* **Socket.io Documentation**: https://socket.io/docs/v4/
* **MongoDB**: <https://www.mongodb.com/>
* **PostgreSQL**: <https://www.postgresql.org/>
* **Color Palette Tools**:
  + ColorHexa: <https://www.colorhexa.com/>
  + Coolors: <https://coolors.co/>
* **Parallax Scrolling Tutorial**: https://gamedevelopment.tutsplus.com/tutorials/parallax-scrolling-a-simple-effective-way-to-add-depth-to-a-2d-game--cms-21586
* **Firebase Authentication**: https://firebase.google.com/docs/auth
* **AWS Cloud Services**: <https://aws.amazon.com/>
* **Privacy Regulations**:
  + GDPR: <https://gdpr.eu/>
  + CCPA: https://oag.ca.gov/privacy/ccpa