CSCE A401/CSCE A601 (Advanced) Software Engineering

TEAM CODE BUSTERS - Brian Fuentes, Gwen Beecher, Nicasio Westlund,
Clint Scholtisek, and Constantine Hinds

Word Buster - Wordle Clone

14 Sep 2024

Introduction:

What's the problem?

While current Wordle apps offer several engaging features such as daily puzzles, universal puzzles, color-coded feedback, social sharing, and statistics tracking, they still have limitations. Furthermore, the gameplay lacks variety that makes the game repetitive and consequently users lose interest over time.

Why it matters:

Varied gameplay may introduce new ways in which users interact with other users and by consequence promote social interaction. Where not only it stimulates brain interaction by introducing new elements and challenges, but also sparks the excitement that allows player retention over time. In a business model, this will matter to generate continued income to maintain the servers in which the game is hosted and pay back to its developers.

How Can/Will We Solve the Problem?

To address these issues, we propose a unique spin on Wordle gameplay. This includes introducing varying word lengths and implementing timed word gauntlets, where the timer increases by a few seconds for each correct guess. These features will add diversity to the gameplay, making it more dynamic and engaging for players.

Roles and responsibilities:

Team Lead: Brian

- Oversee and maintain the production timeline: Ensure that the project stays on schedule and meets all deadlines.
- Act as the Solutions Architect: Design the overall system architecture and ensure all components work together seamlessly.
- Manage personnel and team dynamics: Facilitate communication and collaboration among team members, resolve conflicts, and ensure a productive work environment.

Git Repository Manager: Gwen

- Establish and manage the Git repository: Set up the repository, manage branches, and ensure proper version control practices are followed.
- Organize and maintain project files: Keep the repository organized with clear documentation and structure.

Team Secretary: Constantine

- Record and document weekly Scrum meeting minutes: Take detailed notes during meetings and distribute them to the team.
- Serve as the primary point of communication with the instructor: Relay information between the team and the instructor, ensuring all queries and updates are communicated effectively.

Core Unity Developers: Clint and Constantine

- Responsible for core game structure and functionality: Develop the main game mechanics, user interface, and overall gameplay experience.

Backend Developers: Gwen and Nicasio

- Handle database integration, networking, and AI development: Develop and maintain the backend systems, including databases, server-side logic, and any AI components.

Front End Developers: Brian

- Develop and integrate the Unity front end with the backend: Create the user interface and ensure it communicates effectively with the backend systems.

Risk Analysis:

Potential Risks

1. Gameplay
    - The game might not stand out or be as engaging as other Wordle clones available in the market.
2. Unity Database Integration
    - Unity may present challenges when integrating with web services or SQL databases, potentially causing delays or technical issues.
3. Unity Network Integration
    - Integrating Unity's networking capabilities can be complex and may lead to connectivity issues or performance bottlenecks.
4. Unity Monetization Policies
    - Unity's monetization policies might impose restrictions or additional costs for online software, affecting the project's budget and future revenue model (if applicable).

Mitigation Plan

1. Gameplay
    - Multiple rounds of QA testing: Conduct extensive quality assurance testing to ensure the game is both fun and original. Gather feedback from a diverse group of testers to identify areas for improvement.
    - Add engaging social features: Incorporate new game mechanics or social features
        - e.g.: custom word sharing among friends and a hint system to enhance the gameplay experience and encourage social interaction.
2. Unity Database Integration
    - Use third-party database software: Implement reliable third-party database solutions like MySQL to handle data storage and retrieval efficiently.
    - Write scripts in C#: Develop robust C# scripts to ensure smooth and efficient interaction between Unity and the database, minimizing potential integration issues.
3. Unity Network Integration
    - Use third-party networking tools: Explore and utilize third-party networking tools such as: Netcode for Game Objects (NGO), Photon Unity Networking (PUN), FishNet, or Mirror Networking to facilitate seamless network integration and improve connectivity.
4. Unity Monetization Policies
    - Transition to alternative technologies: If Unity's monetization policies become restrictive, consider transitioning to alternative technologies like HTML/JavaScript or other Unity conversion-friendly tools to maintain flexibility and compliance.

Hardware and software requirements:

Software:

- Unity - WebGL (still deciding version)
- GitHub
- VS Code / VS Studio Community
- Firebase (tentative)

Hardware:

- Stable Internet connection
- 4GB RAM DDR4 or more [to be tested]
- System with Integrated Graphics and
- 32MB of Disk Space (for cache) [to be tested]
- Dual Core Processor with 1GHz or better [to be tested]

Work Breakdown and Milestones:

Development Process: Incremental Development (AGILE)

Initial Version - Base Game

- Ensure the game functions correctly.
- Identify and address any issues or bugs.

Intermediate Version

- Resolve identified issues and bugs.
- Integrate additional features to distinguish our game, including:
    - Social sharing capabilities
    - Various game modes
    - A statistics tracker
- Continuously iterate on different versions based on user feedback.

Final Version

- Release the final version of the game with all issues and bugs resolved.
- Incorporate all favorable features and attributes.

Key dates: (*subject to change*)

- 04 Sep 2024 - Scrum 1
    - Game architecture
    - Assign Roles
- 18 Sep 2024 - Project plan / Game Design Flow Diagram
- 30 Sep 2024 - System requirements due
- *2 Oct 2024 - Core Wordle developed in Unity - sprint milestone*
- *9 Oct 2024 - Core API Design Flow chart - sprint milestone*
- *16 Oct 2024 - Core UI design*
- 21 Oct 2024 - System architecture due
- 28 Oct 2024 - Revised project plan due
- *30 Oct 2024 - Core Network infrastructure -sprint milestone*
- *13 Nov 2024 - Initialize project alpha test and instructor Feedback*
- 18 Nov 2024 - Interactive development plan due
- *20 Nov 2024 - Bug fixes and Finish documentation*
- *27 Nov 2024 - Bug fixes initialize project closed beta test & network test*
- 2 Dec 2024 - Software Deployment Plan due
- *XX Dec 2024 - initial release*

Reporting Practices:

GitHub Repository Management

- Regular commits: Ensure team members commit changes frequently.
- Feature branches: Use branches for different features and merge after testing.
- Backups: Maintain backups of previous iterations.

Weekly SCRUM Meetings (Wednesdays at 6:30pm)

- Discord meetings: Conduct weekly SCRUM meetings on Discord.
    - SWE_A401 - General Voice Chat (subject to change)
- Progress reports: Report on individual progress and challenges.
- Task coordination: Set goals and coordinate tasks for the upcoming week.