Societal Problem Presentation: Rogue-Like Algebra

Societal Problem

Algebra, while being a very important skill that is the backbone of mathematics, is extremely difficult for the average person to understand.

Background

According to PIAAC, in 2017 the average numeracy score for the US was 249, which can indicate difficulty solving problems with unknown variables such as those which require algebra. It is trending downwards from tests conducted in 2012 through 2023. Algebra is used across disciplines to identify relationships and calculate any kind of unknown values from known ones. This is key for STEM, obviously, but also for trades and arts. Anyone would benefit from an easy way to learn algebra. It's used to some degree across almost every field, even outside of STEM.

Customers and End Users

Customers (who will buy the game)

- Schools and tutoring centers
- Parents who want to help their kids learn math
- Self-learners improving their algebra skills
- Libraries and Community learning programs
- Companies for employee training

End Users (who play the game)

- Middle and high school students
- College students needing extra math practice
- Gamers who enjoy strategy games
- STEM learners and aspiring programmers
- Adults wanting to improve their math skills

Proposed Solution

A solo turn-based rogue-like game where the player adventures through dungeons, tackling enemies and increasing in power as you adventure along. The turn-based genre is particularly well-suited for incorporating combat into a math game as it gives the player time to think. Each turn could incentivize the player to think carefully about their moves by dealing damage for each turn the player takes. This means the player will want to defeat the enemies (solve the problems) in as few moves as possible. This both allows for fun optimization challenges while also incentivizing the players to learn and memorize algebraic techniques. Additional penalties would be given for incorrect answers. With regards to the rogue-like mechanics, these games are fun and popular, so that particular part isn't connected to algebra but instead focused on making the game fun and addictive. Enhancements and upgrades could be found to give hints or allow for more moves and tries when solving problems. Gamification is a powerful tool in education as it keeps engagement high while also knocking down the barrier to entry for complex topics.

Software

Web Application

The web-based version of the game will be accessible from any browser, facilitating seamless gameplay without the need for downloads. The game will support mouse and keyboard input. It will be compatible with desktops, Chromebooks, and tablets, enabling students and self-learners to access the game on the go. Players can conveniently log in to save their progress and seamlessly switch between devices. Progress will be seamlessly synced across devices through cloud storage, allowing players to continue their journey regardless of the device they are using.

The difficulty level of the game will dynamically adjust based on the player's proficiency, maintaining a challenging yet rewarding experience. Daily challenges and streak rewards will incentivize regular practice, while push notifications will serve as reminders to engage with the game. Additionally, leaderboards and challenges will foster a competitive environment, while potential integration with Learning Management Systems (LMS) will facilitate the tracking of student progress by educators.

Furthermore, the game will offer optional in-app purchases for hints or cosmetic enhancements, with an ad-free version available for users seeking a more immersive experience. A school-friendly version, devoid of advertisements and in-app purchases, will be made available for educational purposes.

Mobile Application (Optional)

The mobile version of the game will feature an intuitive touch-based interface, enabling players to effortlessly solve algebraic equations by tapping or dragging. Notably, the mobile version will be able to operate offline, ensuring uninterrupted gameplay for users.

Tools and Technologies

The game will be constructed utilizing Rust, ensuring optimal performance, memory safety, and cross-platform compatibility. Rust will handle all game logic, providing a performant and reliable experience. For the game engine, we have two main choices: Godot or Bevy.

Godot benefits from a wide ecosystem of third-party plugins, meaning many complex things won't need to be developed from scratch. Additionally, the onboarding is much easier and development will be faster. The downside to Godot is that it doesn't have complete support for Rust, so a few things will need to be written in Godot's native language, GDScript. This is also a benefit though, as Rust and GDScript are only two of the many supported languages.

The second choice will be Bevy, a Rust-based framework specifically designed for efficient and flexible game development. The primary benefit of Bevy is that it is pure Rust. However, this does also present a problem should we need to pivot to another language for some reason, as Bevy only supports Rust. A second reason to use Bevy is the ECS paradigm. Bevy uses an "entity component system," or ECS, which allows for extremely efficient and parallelizable code.

For mobile development, Rust can be compiled for Android (utilizing JNI bindings) and iOS (utilizing FFI with Swift). If additional platform-specific requirements arise, Android Studio (Kotlin) and Xcode (Swift) may be employed.

Firebase will manage cloud saves, authentication, and real-time data synchronization, while Google Play Services and Apple Game Center will provide leaderboards and achievements.

The web version will be developed using WebAssembly (WASM), enabling Rust to execute efficiently within a browser. The game will utilize wgpu for rendering graphics and HTML5, CSS3, and JavaScript for user interface elements. This combination ensures seamless browser-based gameplay with optimized performance.

For backend and storage, Actix Web or Axum (Rust web frameworks) will be utilized for server-side logic. PostgreSQL or MongoDB will manage user progress,

leaderboards, and game data. These selections guarantee scalability and security for data handling.

Version control will be managed using Git and GitHub, and testing will be conducted utilizing Rust's built-in testing framework with cargo test. These tools and technologies will ensure the game's rapidity, scalability, seamless cross-platform compatibility, and leverage Rust's safety and efficiency.

References

"Highlights of the 2023 U.S. PIAAC Results." *National Center For Education Statistics*, 2023, nces.ed.gov/surveys/piaac/2023/national_results.asp. Accessed 31 Jan. 2025.