DoubleDash

Ishwarya Narayana Subramanian
Version #1

Summary of Project

"DoubleDash" is an endless mobile game developed in Unity where players control two cars simultaneously. Navigate both cars through a dynamic environment, collecting points and avoiding obstacles on a four-lane road. Failure to collect points or hitting obstacles ends the game. With intuitive touch controls and addictive gameplay, "DoubleDash" offers a thrilling challenge for players on the go.

Project Analysis

Value Proposition

Players crave exhilarating challenges that push their skills to the limit, but finding games that offer innovative gameplay experiences can be challenging in a saturated market. However, discovering titles that truly push creative boundaries and deliver fresh experiences remains a hurdle. "DoubleDash" is designed for gamers seeking a fresh and thrilling gaming experience that pushes the boundaries of entertainment and challenge. Most gamers express a desire for more challenging gameplay experiences, as player reviews and ratings of popular games often highlight the importance of challenging content and skill-testing mechanics. DoubleDash provides players with a never-ending gaming experience that keeps them on the edge, offering continuous challenges and excitement without a predetermined endpoint. The game's dynamic and relentless nature ensures that players are constantly engaged and immersed in the action, heightening the thrill of gameplay and fostering a desire to achieve higher scores and mastery.

Primary Purpose

The purpose of the DoubleDash project is to provide gamers with an exhilarating and immersive gaming experience that challenges their skills and keeps them engaged indefinitely. By offering continuous excitement and dynamic gameplay, DoubleDash aims to fulfill the desire for challenging and thrilling gaming experiences while fostering a sense of mastery and achievement among players.

Target Audience

Targeting teenagers and young adults aged 18-25, with a secondary focus on adults aged 25-35, DoubleDash aims to cater to individuals seeking distraction amidst their hectic work or study schedules. This demographic is chosen for their familiarity with and affinity for video games and their inclination towards seeking entertainment and relaxation during breaks or downtime. This demographic can be reached by distributing via app stores on smartphones, engaging on Instagram, TikTok, and Twitter, collaborating with influencers, participating in gaming forums and communities, and advertising based on demographics and interests.

Success Criteria

Determining the success of DoubleDash involves assessing various metrics. Financial gain is measured through revenue from in-app purchases, ad revenue, and subscriptions. User satisfaction is gauged by monitoring feedback, app ratings, and reviews. Market share is evaluated by comparing with

competitors and tracking download numbers. Assessing the app's impact on mental well-being through user testimonials and surveys contributes to understanding its public good. Engagement metrics such as Daily Active Users, Monthly Active Users, session duration, and retention rates provide insights into user engagement and loyalty.

Competitor Analysis

In comparison to our competitors, DoubleDash offers unique strengths that differentiate it in the market. While competitors may excel in certain areas, such as graphics or brand recognition, DoubleDash stands out with its focus on providing a truly challenging and engaging gaming experience. Our competitors may lack the dynamic and relentless nature of gameplay that keeps players on the edge in DoubleDash.

Monetization Model

DoubleDash will employ a freemium monetization model, offering the base game for free with optional in-app purchases (IAPs) for additional features, power-ups, and customization options. The game will incorporate non-intrusive advertising, such as rewarded ads, to provide players with the option to earn in-game rewards or continue gameplay uninterrupted. This combination of IAPs and advertising will ensure a sustainable revenue stream while maintaining a positive user experience.

Initial Design

For the initial design of DoubleDash, the Minimum Viable Product (MVP) will focus on delivering the core gameplay experience that aligns with our value proposition of providing an engaging and challenging gaming experience for players. The MVP will include the following key features:

- *Players will be able to control two cars simultaneously, one on the left side of the road and the other on the right.*
- *The game will feature a variety of obstacles, such as barriers, moving obstacles, and collectible points, placed along the road to challenge players' multitasking skills.*
- *Players will be required to switch lanes to avoid obstacles and collect points, with the left cars controlling the left lanes and the right cars controlling the right lanes.*
- *The game will offer an endless gameplay experience, with increasing difficulty over time to keep players engaged and challenged.*
- *Points will be awarded for successfully collecting points. The scoring system will allow players to track their progress and compete for high scores.*

Scope and Expected Limitations

The MVP will focus solely on single-player gameplay, with no multiplayer or social features included initially. The game's visuals and sound effects will be kept simple and functional, with more elaborate graphics and audio enhancements planned for future updates. Additional game modes, power-ups, and customization options may be introduced in later iterations based on player feedback and development resources. The MVP will be developed for mobile platforms initially, with potential expansion to other platforms in the future.

By focusing on these core features and keeping the scope manageable, we aim to deliver a polished and enjoyable gaming experience that aligns with our vision for DoubleDash while also allowing for flexibility and scalability in future updates and enhancements.

UI/UX Design

For the MVP of DoubleDash, the UI/UX design will focus on providing a seamless and intuitive gaming experience that aligns with our audience's preferences and the game's purpose and value proposition. Important UI/UX components to include are:

- *A simple and visually appealing main menu that allows players to start the game, access settings, view high scores, and navigate to other sections of the app.*
- *A clean game interface that prominently displays the two cars and the road ahead with essential elements such as score and a pause button to pause the game.*
- *Intuitive touch controls that allow players to switch lanes smoothly and accurately, ensuring responsive gameplay.*
- *Simple yet engaging graphics that convey the game's fast-paced and dynamic nature. Bright colors, clear visual cues for obstacles and points, and smooth animations enhance the overall gaming experience.*

By prioritizing these UI/UX components, we aim to create an MVP that delivers a fun and immersive gaming experience while laying the foundation for future enhancements and refinements based on player feedback and evolving market trends.

Technical Architecture

To support the MVP of DoubleDash, the technical architecture will require the following components:

- *Utilizing Unity as the game engine to develop and deploy the game across multiple platforms, ensuring cross-platform compatibility and efficient development.*
- *Implementing data structures such as arrays or lists to manage game elements like obstacles, points, and player scores. This allows for efficient storage and retrieval of game data during runtime.*
- *Utilizing local storage on the device to save player progress, high scores, and game settings while also considering cloud storage options for cross-device synchronization and backup.*
- *Implementing game logic to control player movement, obstacle spawning, collision detection, and score calculation. This involves scripting in C# within the Unity environment to create the core gameplay mechanics.*
- *Integrating the UI/UX components mentioned earlier into the game using Unity's UI toolkit, ensuring a seamless and visually appealing user experience.*
- *Success can be measured through various metrics, including player engagement (DAU, MAU), retention rates, in-app purchases, and user satisfaction (feedback, ratings). Analyzing these metrics allows for iterative improvements to the game and informs future development decisions.*

Dependencies on third-party services/APIs:

- *Firebase will be used for backend services such as cloud storage, authentication, and analytics tracking.*

- *Unity Asset Store will be used for accessing third-party assets and plugins to enhance game development efficiency and functionality.*

By implementing these components and considerations, we aim to create a robust technical architecture that supports the MVP of DoubleDash while providing a foundation for future enhancements and scalability.

Challenges and Open Questions

Technical Challenges:

- *One challenge is ensuring that the player progress, settings, and preferences are restored when the game is reinstalled on a device or when switching to a new device. This involves managing local data storage and synchronization across devices while maintaining data integrity and security.*
- *Another challenge is establishing and maintaining communication with web services for features such as analytics tracking, and in-game purchases. This requires handling network requests, managing data exchange between the game client and server, and ensuring reliable connectivity and error handling.*

Proposed Solutions:

Restoration of Data on Reinstall:

- *Implement cloud-based storage solutions such as Firebase Firestore or Google Cloud Storage to securely store player data.*
- *Utilize device identifiers (e.g., device ID, user ID) to associate player data with their account, allowing for seamless restoration of data across devices.*

Interaction with Web Services

- *Use RESTful APIs or SDKs provided by web service providers (e.g., Firebase, PlayFab) to facilitate communication between the game client and server.*
- *Implement asynchronous network requests to avoid blocking the main thread and ensure smooth gameplay experience.*

Feedback Required:

1. *What are the preferred web service providers or APIs for leaderboard integration, analytics tracking, and in-game purchases among the target audience?*
2. *Are there any specific security or privacy concerns related to storing and accessing player data in the cloud?*