MS PlayFab Assignment

Dev_Team_2

Aravind Vudatha

Lakshmi Priya Mutyala

Vamshi gone

Gopi Mandapati

Introduction

PlayFab is a comprehensive backend platform that enables game developers to build, launch, and operate their online games more efficiently. It provides a wide range of services that allow developers to focus on game development, while PlayFab handles the backend infrastructure and services.

PlayFab offers an array of tools for game developers to create and manage their game's online features. Some of these services include:

Player authentication: PlayFab provides authentication services that allow game developers to securely authenticate players, protecting their game's data and services from unauthorized access.

Data storage: PlayFab provides scalable and flexible storage services for game data, allowing developers to store player data, game settings, and other important game-related data.

Matchmaking: PlayFab's matchmaking services enable game developers to easily match players with each other based on different criteria, such as player skill, location, and availability.

Leaderboards: PlayFab provides leaderboard services that allow developers to track and display player rankings based on different game metrics, such as high scores, completion times, or other in-game achievements.

Player messaging: PlayFab's messaging services allow game developers to send targeted messages to players, such as notifications, announcements, or other in-game messages.

Content management: PlayFab provides services to manage game content, such as digital assets, player generated content, or downloadable content.

One of the benefits of PlayFab is its scalability and flexibility. As games grow and evolve, PlayFab can easily adapt to changing player demand and game requirements, ensuring that the backend services are always optimized for the best player experience.

In 2018, Microsoft acquired PlayFab to integrate its services into the Azure cloud platform and to further enhance its gaming capabilities. PlayFab is now a part of Microsoft's gaming division, and its services are used by many game developers worldwide.

Services

Vanguard-Outrider, the most relevant PlayFab services to integrate would likely include:

- Player authentication: This service allows players to securely log in to the game, ensuring that only authorized players can access the game's content and features. This is important for protecting player data and preventing unauthorized access.

- Data storage: This service allows the game to store player data, game settings, and other important game-related data. This could include things like player progress, inventory, and game preferences.
- Leaderboards: This service allows the game to track and display player rankings based on different game metrics, such as high scores, completion times, or other in-game achievements. This can help drive player engagement and competition within the game.
- Matchmaking: This service allows the game to match players with each other based on different criteria, such as player skill, location, and availability. This can help ensure that players have a balanced and enjoyable game experience.
- Player messaging: This service allows the game to send targeted messages to players, such as notifications, announcements, or other in-game messages. This can help keep players engaged and informed about new content or events within the game.

By integrating these PlayFab services into the Vanguard-Outrider demo game, the game developer can focus on creating a compelling gameplay experience, while PlayFab handles the backend infrastructure and services. This can help reduce development time and cost, while also ensuring that the game has reliable and scalable online features.

Implement player authentication using PlayFab's Login API.

Implementing player authentication using PlayFab's Login API involves several steps:

- Set up a PlayFab account: Before you can use PlayFab's Login API, you need to create a PlayFab account and set up your game project in the PlayFab developer dashboard.

- Choose a login method: PlayFab offers several different login methods, including email and password, Facebook, Google, and more. Choose the login method that works best for your game.
- Integrate the PlayFab SDK: You'll need to integrate the PlayFab SDK into your game code. The SDK is available for several different programming languages, including C#, Unity, and more.
- Call the Login API: Once the SDK is integrated, you can call the Login API to authenticate the player. The exact code you'll need to write will depend on your chosen login method. Here's an example of how to authenticate a player using email and password.
- Handle login callbacks: The Login API is asynchronous, meaning that it will return a callback when the authentication process is complete. You'll need to handle these callbacks in your game code to perform the appropriate actions, such as loading the player's data or displaying an error message.
- By following these steps, you can implement player authentication using PlayFab's Login API and ensure that only authorized players can access your game's content and features.

Setting up and managing leaderboards using PlayFab's statistics and leaderboard APIs involves several steps:

- Create a leaderboard: In the PlayFab developer dashboard, go to the Leaderboards section and create a new leaderboard. You can choose the type of leaderboard (e.g. high score, completion time, etc.), the reset interval (e.g. daily, weekly, etc.), and other settings.
- Add statistics: In order to populate the leaderboard, you'll need to define one or more statistics that the leaderboard will track. For example, if you're creating a high score leaderboard, you might create a "Score" statistic.
- Submit statistics: In your game code, you can submit player statistics to PlayFab using the Statistics API. Here's an example of how to submit a player's score:

- Retrieve leaderboard data: You can retrieve leaderboard data using the Leaderboard API. Here's an example of how to retrieve the top 10 scores for a high score leaderboard:
- Display leaderboard data: Once you've retrieved the leaderboard data, you can display it in your game's user interface. PlayFab provides a Leaderboard UI component that you can use to display leaderboard data in Unity games, or you can create your own custom UI.

By following these steps, you can set up and manage leaderboards using PlayFab's statistics and leaderboard APIs, allowing your players to compete with each other and drive engagement within your game.

- Integrating PlayFab's analytics services to track player behavior and game performance involves several steps:
- Set up PlayFab analytics: In the PlayFab developer dashboard, go to the Analytics section and set up your analytics configuration. You can choose the analytics provider (such as Google Analytics or Mixpanel), enable custom events and metrics, and other settings.
- Integrate the PlayFab SDK: You'll need to integrate the PlayFab SDK into your game code. The SDK is available for several different programming languages, including C#, Unity, and more.
- Log custom events: In your game code, you can log custom events using the PlayFab analytics API. These events can include player actions (such as completing a level or making a purchase) or performance metrics (such as FPS or memory usage). Here's an example of how to log a custom event:
- View analytics data: In the PlayFab developer dashboard, you can view analytics data for your game, including custom events and metrics, player retention, and more.
- Analyze and optimize: Use the analytics data to analyze player behavior and game performance, and make informed decisions about how to optimize your game. For example, you might use the data to identify areas where players are getting stuck and adjust the game design to improve the player experience.

By following these steps, you can integrate PlayFab's analytics services to track player behavior and game performance, and use the data to optimize your game and drive engagement.

Testing the integrated PlayFab services in the Vanguard-Outrider demo game is an important step to ensure proper functionality. Here are some steps you can follow to test the integrated services:

1. Test player authentication: Make sure that players can create accounts, log in, and log out of the game using PlayFab's Login API. Test different scenarios, such as logging in with incorrect credentials or logging in with a new account.
2. Test leaderboards: Make sure that player scores are being submitted to the leaderboard correctly using the Statistics API, and that the leaderboard is being updated correctly. Test different scenarios, such as submitting a new score, viewing the leaderboard, and checking if the leaderboard resets correctly.
3. Test analytics: Make sure that custom events and metrics are being logged correctly using PlayFab's analytics API, and that the data is being displayed correctly in the PlayFab developer dashboard. Test different scenarios, such as logging events for different player actions and checking if the data is being tracked correctly.
4. Test data storage: Make sure that player data is being stored correctly using PlayFab's data storage APIs, and that the data can be retrieved and updated correctly. Test

   different scenarios, such as saving and retrieving player preferences, and making sure that the data is persistent across multiple game sessions.
5. Test error handling: Test different error scenarios to ensure that proper error messages are being displayed to the user, and that the game is handling errors correctly. This includes testing scenarios such as network connection errors, server downtime, and invalid user input.

By following these steps, you can test the integrated PlayFab services in the Vanguard-Outrider demo game to ensure proper functionality, and make any necessary adjustments before releasing the game to players.