Lab 1 - GameEye Product Description

Jacob L. Cook

Old Dominion University

CS 411W: Professional Workforce Development II

Professor Janet Brunelle

May 4, 2020

Draft

Table of Contents

Table of Figures

1. Introduction

1.1 Societal Problem

According to Gough (2019a), the rate at which video games are released worldwide on Steam is increasing exponentially. Gough (2019a) provides data indicating that, worldwide, the number of new video games released on Steam grew from 4207 new games in 2016 to 9050 new games in 2018. The yearly release rate in 2018 is more than double the yearly release rate in 2016. In addition, the increasing video game release rate, the number of video gamers is increasing significantly. Gough (2019b) predicts that, by 2021, the number of video gamers worldwide will surpass 2.7 billion. For gamers (whether new or experienced), the growing pool of thousands of video games is overwhelming.

Coupled with the quantity of new releases, several other factors amplify the problem. For instance, gamers struggle to keep up-to-date with the development of new video games. There is great variation in the number of years from the date a game is announced to the date a game is released. Dietz (2011) illustrates that a prime example of a game with a long development timeline is Team Fortress 2. Team Fortress 2 was announced in 1998 and eventually released in 2007, spanning nine years. While under development, games have distinct statuses, such as “announced”, “delayed”, and “released”. For games with extensive development timeframes, gamers face the difficulty of staying apprised of new information and status updates. Game developers, especially those with small marketing budgets, suffer financially from uninformed video game consumers. In general, gamers use social media platforms, such as Twitter, YouTube, and Reddit, video game news websites, Steam, Google, and game databases (e.g., IGDB) to maintain awareness of video games news. Due to the overwhelming amount of game data, consumers need a centralized platform to maintain awareness of favorite video games.

1.2 Solution Description

GameEye is a platform for consumers to access game news and updates from a central location. Users can search for their favorite games and curate a personal watchlist. Each user can add a game they wish to follow to their watchlist. Once a game is added to the user's watchlist, they will receive notifications when GameEye finds new related content. Rather than bombarding users with all information related to a game, machine learning classifies news articles and Tweets into specific content categories. In addition to content categorization, each news resource is prioritized by a multifactor score computed by machine learning. GameEye informs users of the most popular games currently added to other watchlists. By centralizing game news, providing users with personal watchlists, and ranking information by importance, GameEye is the gamer's eye (Figure 1) into the video game industry.

Figure 1: GameEye Logo

2. Product Description

2.1 Key Product Features and Capabilities

2.1.1 General

GameEye's design leverages progressive web application (PWA) technologies, enabling usage on desktop and mobile devices. PWA architecture uses common web technologies to deliver the same user experience on any device with a modern web browser. Several features ensure a consistent and durable user experience. GameEye offers offline support by locally caching recently viewed game content. Connectivity interruption resiliency features create a durable user experience.

2.1.2 Authentication

GameEye implements several authentication features to protect users while facilitating a user-friendly experience. Users have access to a secure login and registration portal through the application. Login and registration is available through GameEye via Auth0, or through an external provider (e.g., social media accounts). Persistent sessions, which remember users that recently signed, in remove the inconvenience of constant authentication. To increase the security for users, two-factor authentication is a core feature of GameEye. GameEye employs recovery mechanisms to mitigate the risk of users forgetting their passwords.

2.1.3 Account Management

Users can manage several aspects of their account through GameEye's account management portal. For instance, users can change their passwords. Users can also modify their profile information including first name, last name, and email address. Finally, users can delete their GameEye accounts.

2.1.4 Game Tracking

2.1.4.1 Personal Watchlists. Customizable watchlists are the core feature that GameEye provides to users for game tracking. When users identify a game of interest, they add the game to their watchlist. The personal watchlist allows users to narrow in on games of interest and filter out the rest. User watchlists contain a list of game titles that a user is following and a thumbnail image representing the game.

2.1.4.2 New Game Updates. To structure game content, GameEye organizes information into six distinct categories: important updates, news articles, Tweets, Reddit posts, images, and videos. Game content is populated in the user's watchlist for each game they are following. Users can find a link to full content in each category within their personal watchlist along with a thumbnail image for news articles and Tweets.

2.1.4.3 Most-Watched Games List. In order to showcase popular games, the most-watched games list is available for users. GameEye determines game popularity by the total number of followers for a specific game title.

2.1.5 Searching

GameEye allows users to search for video games by title and quickly add a game to their watchlist. As a user types the game title, auto-completion assists the user in locating the desired video game. For users without a specific game title in mind, browsing the most-watched games list serves as another search method.

2.1.6 Web Scraping

GameEye uses web scrapers to retrieve the content that populates a user's watchlist categories. Web scraping gathers news articles from video game news websites, removing the need for users to navigate to dozens of web sources. GameEye locates Tweets about video games from official publisher Twitter feeds via web scraping. Web scrapers find Reddit posts from official game subreddits. GameEye provides users with links to game videos that the web scrapers find on YouTube channels. Web scraping enables GameEye to show users images relevant to their watched video games.

2.1.7 Notifications

Users receive push-notifications for new game updates whether using GameEye on desktop or mobile systems. The user interface provides a count of notifications for each game and category in a user's watchlist. Users receive notifications to suggest games on the most-watched games list in which they may be interested.

2.1.8 Settings

GameEye provides users with a customizable experience managed through content, notification, and general settings. Content preferences allow users to determine if they would like to view archived resources (i.e., resources older than 90 days). Importance scores assigned to news articles and Tweets can be enabled or disabled. Users can fine-tune notification preferences by resources categories, or they can disable notifications entirely. To improve the user experience, users can submit feedback about GameEye.

2.1.9 Machine Learning

Using game data, machine learning algorithms classify game updates into meaningful categories. Game update categories include release date announcement, delay, major game update, and minor game update. GameEye determines the importance of news articles and Tweets by assigning an "importance score". Machine learning enables accurate generation of importance scores for resources. GameEye also employs machine learning to extract important information from news articles and Tweets. Users receive important information and have the option to view importance scores of resources through GameEye's machine learning component.

2.2 Major Components

2.2.1 Hardware

GameEye's hardware components, as depicted in Figure 2 on page 10, include a front-end server, a main back-end server, a web scraping back-end server, a machine learning back-end server, and a main database server. The front-end server allows users to connect directly to GameEye on a desktop or mobile system. Connections between the front-end, push-notifications, main database, web scraping, and authentication service are available through the main back-end server. The web scraping back-end server is responsible for scraping the web for game data and relaying data to the main database server and machine learning back-end server. The machine learning back-end server receives raw data from the web scraping back-end server and produces game updates, important information, and importance scores. Containing user and game data, the main database server provides the main back-end server with game data and user information. In addition, the main database server receives game data from the web scraping back-end server.

2.2.2 Software

2.2.2.1 Frontend. GameEye's front-end will be implemented with several technologies and tools. HTML, SASS/CSS, and TypeScript will be the front-end development languages. SASS (Syntactically awesome style sheets), an extension of CSS, will enable more robust development of style sheets. TypeScript, which is similar to JavaScript, will allow large-scale JavaScript applications for modern web browsers. WebStorm IDE (integrated developer environment) will be used for web development in the outlined programming languages. Angular Framework and Google Workbox (PWA libraries) will be used for development.

2.2.2.2 Backend. Java will be the primary programming language to implement GameEye. IntelliJ IDEA IDE will be used for Java development and Spring Framework will be employed to ensure GameEye is secure and responsive. The jsoup library will be used for web scraping. For database integration, the MongoDB Java Driver will be used to directly interface with the main database from within Java.

2.2.2.3 Testing. Test driven development (TDD) will be followed by using the Junit Java Framework for unit testing. For front-end testing, Jest JavaScript Framework will be employed to test JavaScript code. Unit testing will occur throughout each stage of development and system and stress testing will occur in the final phase of development.

2.2.2.4 Machine Learning. Python, an interpreted, high-level programming language, will be used to implement the machine learning functionality of GameEye. GameEye will rely on two primary machine learning libraries: Keras (deep learning library) and scikit-learn library. Keras is an API for Python that allows easy deployment of machine learning algorithms. The machine learning library, scikit-learn, will allow GameEye to classify game data. TensorFlow Serving is used to interact with the REST API, as shown in Figure 2 on page 10. Machine learning is the core of GameEye's innovation, allowing users to receive important information.

2.2.2.5 Natural Language Processing. Python will also be used to implement the natural language processing (NLP) necessary to determine important information. The spaCy library will be used for GameEye's NLP component.

2.2.2.6 Databases. GameEye's main database will implemented using MongoDB, a non-relational database language and platform. GUI access will be provided through MongoDB Compass. The main database will connect to the video game database (IGDB) via a REST API. MongoDB enables GameEye to have fast and responsive database. In addition, flexible database schema allows for growth in the structure of the types of data that GameEye collects.

2.2.2.7 COTS Software. GameEye will employ two commercial off-the-shelf, or third-party, software packages: Auth0 Single Page App SDK and Firebase Cloud Messaging (using Firebase Admin Java SDK). Auth0 will allow GameEye to authenticate users via auth tokens, which are credential for a login session, off-loading the security risks of storing user passwords or password hashes to a third-party service. Auth0 provides an SDK for development and integration into GameEye's application. GameEye will employ Firebase Cloud Messaging to send push-notifications to users when new content is found.

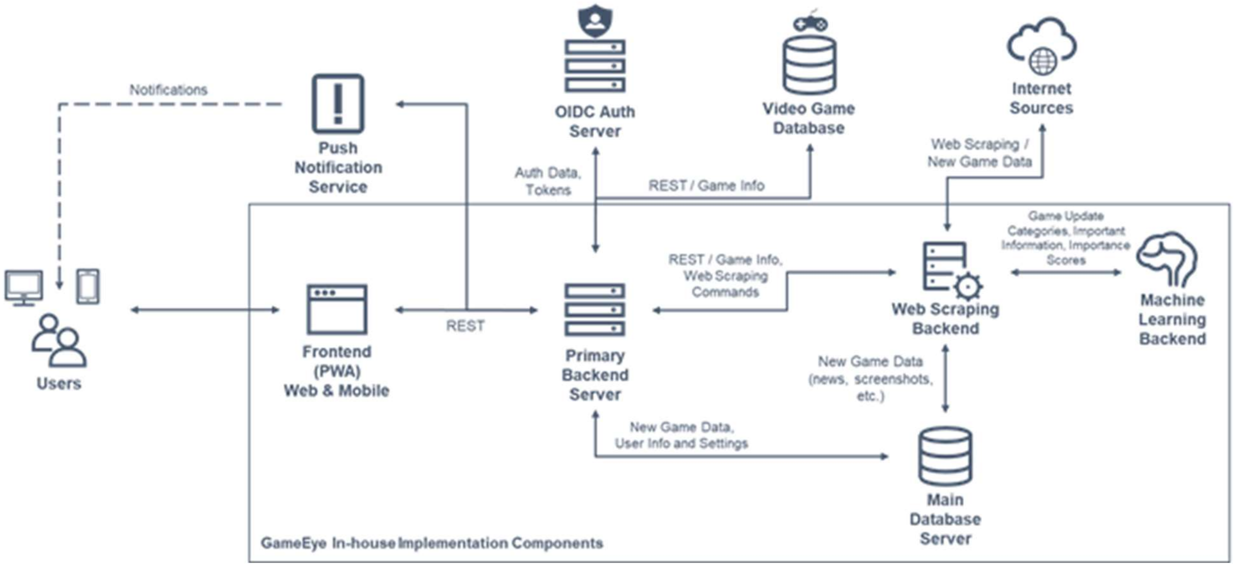
Notifications
Push Notification Service
OIDC Auth Server
Video Game Database
Internet Sources
Web Scraping / New Game Data
Auth Data, Tokens
REST / Game Info
Game Update Categories, Important Information, Importance Scores
REST / Game Info, Web Scraping Commands
Web Scraping Backend
Machine Learning Backend
Users
Frontend (PWA) Web & Mobile
REST
Primary Backend Server
New Game Data (news, screenshots, etc.)
New Game Data, User Info and Settings
Main Database Server
GameEye In-house Implementation Components

Figure 2: Major Functional Component Diagram

3. Identification of Case Study

GameEye is designed for gamers seeking to filter the noise produced by a myriad of video game news sources. This target market includes experienced and novice gamers. GameEye is designed to provide any gamer with notifications for their favorite video games. In addition, users are unburdened from the time-consuming process of finding game news. Any group of gamers would be sufficient for a representative case study. The characteristics of GameEye's target audience is not likely to change in the future.

4. Glossary

Angular Framework: Platform for building mobile and desktop applications.

API: Application Programming Interface; a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other services.

AWS: Amazon® subsidiary that provides on-demand cloud computing platforms and APIs

CSS: Cascading Style Sheets; used to stylize webpages.

Guest: Initial role for users who have not created an account on GameEye.

Hitlist: List of highly watched video games by users.

HTML: Hypertext Markup Language; used as markup for documents meant to be displayed in a web browser.

IGDB: Database of known video games, accessed by REST API to populate GameEye's database

Indie Games: Games developed by individuals or smaller teams of people without the financial support of larger game publishers.

IntelliJ Idea: IDE developed by JetBrains to write Java applications and will be used in the back-end development of GameEye.

JavaScript: Object-oriented language used to create dynamic, interactive effects on webpages.

Jest JavaScript Framework: Testing framework maintained by Facebook Inc.

JSoup Library: Java library for working with real-world HTML.

JUnit Java Framework: A testing framework for Java.

Keras (Python Deep Learning Library): Open-source neural-network library written in Python.

MongoDB: A cross-platform document-oriented database program

Noise Filtering: Information/news articles shown that caters to an individual's content preferences.

OIDC Authentication: Authentication protocol based on the OAuth2.0 family of specifications.

Progressive Web Application (PWA): a type of application software delivered through the web which is built using common web technologies including HTML, CSS, and JavaScript.

Python: Interpreted, high-level, general-purpose programming language.

REST: Software architectural style used in creating web services.

RSS Feed: Really simple syndication; web feed that allows users and applications to access updates to websites in a standardized, computer-readable format.

Scikit-learn Library: Software machine learning library for the Python programming language.

SpaCy Library: Open-source software library for advanced natural language processing.

Spring Framework: Application framework and inversion of control container for the Java platform.

Tester: GameEye beta testers; users of the application in its prototype phase who will provide feedback on their experience.

Web Scraping: Data scraping for extracting data from websites.

WebStorm: IDE developed by JetBrains to write JavaScript code.

5. References

Anderton, K. (2019, June 26). *Video game market share and growth*. [Infographic]. https://www.forbes.com/sites/kevinanderton/2019/06/26/the-business-of-video-games-market-share-for-gaming-platforms-in-2019-infographic/#c1793427b254

Dietz, J. (2011, June 23). 30 Games that emerged from development hell. https://www.metacritic.com/feature/games-that-shed-vaporware-status

Gough, C. (2019, August 9). Number of games released on Steam 2018. https://www.statista.com/statistics/552623/number-games-released-steam/

Gough, C. (2019, August 9). Number of gamers worldwide 2021. https://www.statista.com/statistics/748044/number-video-gamers-world/

Gough, C. (2019, October 9). Google Play: Number of available games by quarter 2019. https://www.statista.com/statistics/780229/number-of-available-gaming-apps-in-the-google-play-store-quarter/

Humphries, M. (2019, September 18). Twitch acquires gaming database website IGDB. https://www.pcmag.com/news/twitch-acquires-gaming-database-website-igdb

Rose, M. (2014, May 15). How the surge of Steam releases will affect game developers. https://www.gamasutra.com/view/news/217583/How_the_surge_of_Steam_releases_will_affect_game_developers.php