

Lab 1: GameEye Product Description

Christopher Diasanta

Old Dominion University

CS 410

Professor Janet Brunelle

Team Yellow

4 May 2020

Version 1.0

### Table of Contents

1.	Introduction .....	3
1.1.	Societal problem .....	3
1.2.	Solution Description .....	3
2.	GameEye Product Description .....	5
2.1.	KeyProduct Features and Capabilities .....	4
2.2.	Major Components .....	7
3.	Identification of Case Study .....	9
4.	GameEye Product Prototype Description .....	11
4.1.	Prototype Features and Capabilities .....	11
4.2.	Prototype Architecture .....	12
4.3.	Prototype Development Challenges .....	12
5.	Glossary .....	13
6.	References .....	15

### Table of Figures

Figure 1: Current Process Flow Diagram for Gamer.....	3
Figure 2: Solution Process Flow Diagram for Gamer.....	5
Figure 3: Major Functional Component Diagram (MFCD).....	8
Figure 4: Software Components Diagram .....	9

## 1. Introduction

### 1.1 Societal Problem

In recent years, there has been a surge in the sheer amount of video games released and with these games involves long development time causing difficulty and time consumption to stay up-to-date with these video games. In 2018 alone, over 9,000 video games were released on the popular video game distribution platform, Steam(Gough, 2019). There is a lack of a convenient way for gamers to keep up-to-date about developments on the vast amount of video games especially since information about video games is very decentralized. Independent game developers have difficulty maintaining public attention as a result. The figure below shows the current process flow a gamer might follow.

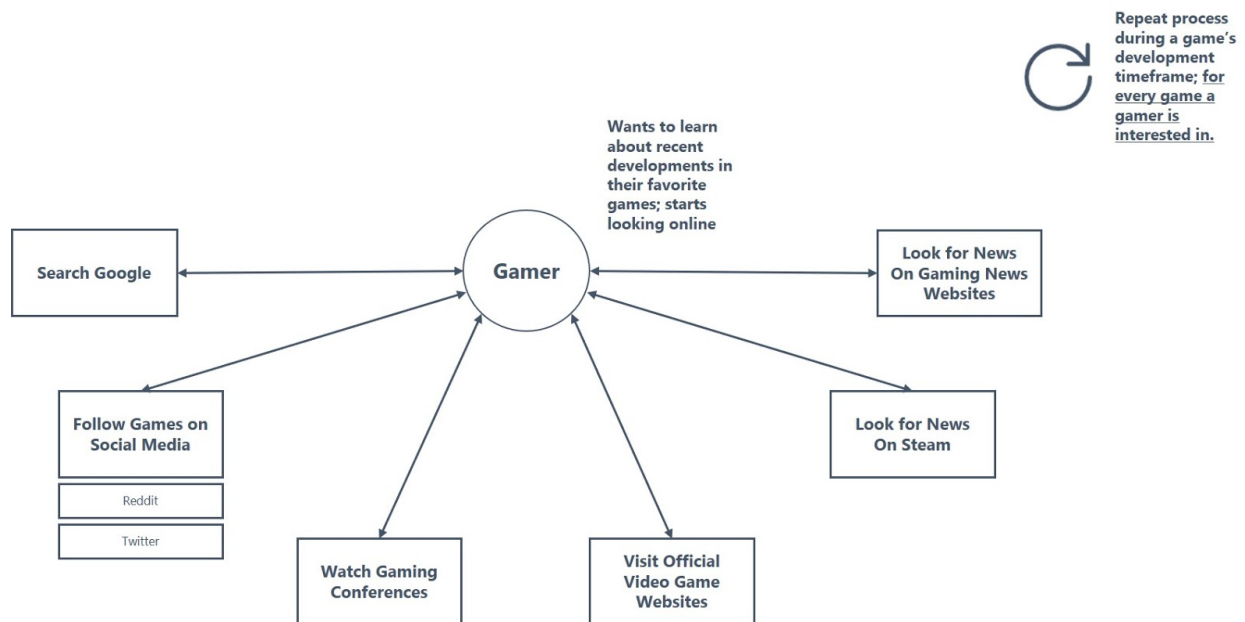


Figure 1: Current process flow diagram describing means of gamer attaining information.

A gamer may already have a means of searching for video game developments. A gamer may want to search about recent developments in one of their favorite video games so they may start looking online. They might search google, follow games on social media, watch

game conferences, visit official video game websites, look for news on steam, or look for news on gaming news websites. This process would be repeated for every game a gamer may be interested in. Some of these news sources like social media may be unverified or spread false information. This also does not account for some independent game developers having difficulty maintaining public attention.

## **1.2 Solution Description**

The solution to this problem would be GameEye. GameEye is a platform in which users can follow/watch games. GamEye Users will be sent notifications of news articles or new content on video games they follow (e.g. screenshots and official videos like trailers or gameplay). Users will have a watch list in which they will be able get updated about the status of current and unreleased games they may be interested in(e.g. Potential release date delays and news updates). The news articles and tweets will be classified depending on what they are about. An importance factor score will be computed to see how important it is for a user to know.

[THIS SPACE INTENTIONALLY LEFT BLANK]

## 2. GameEye Product Description

GameEye is a web application that will aggregate news information about developing video games and users will receive notifications on the new information. Users will be able to go to one place for all their wanted video game development updates.

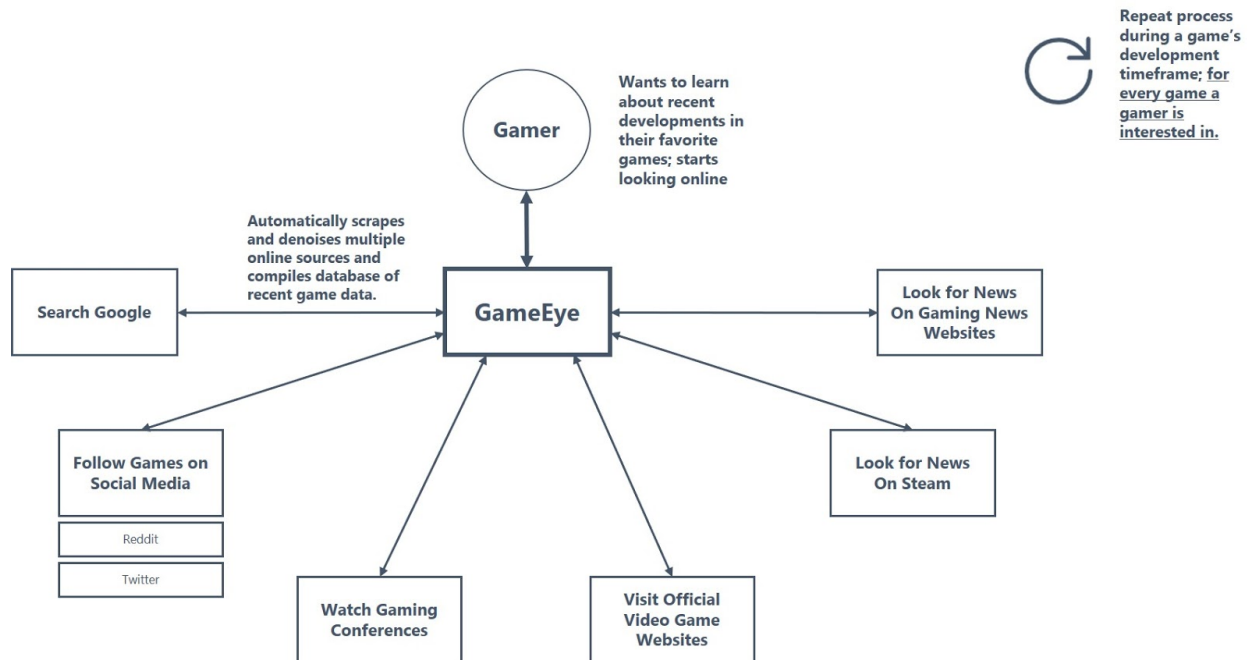


Figure 2: Solution Process Flow Diagram describing aggregation of information to gamer.

### 2.1 Key Product features and Capabilities

#### 2.1.1 General

GameEye will be a cross-platform progressive web application working on both mobile and desktop devices. It features offline support through local caching and connectivity interruption resiliency features.

#### 2.1.2 Authentication

It will also feature authentication like secure user login and registration. An external provider will be for user login and registration (e.g. logging in using social media accounts). There will also be use of persistent sessions allowing the user ease of access as they do not

have login every time. Two-factor authentication may be used as well for increased security.

There will also be a password recovery mechanism in the event a user forgets their password.

### **2.1.3 Account Management**

As for a user's GameEye account they have the ability to change their password, modify their profile information, and the ability to delete their account.

### **2.1.4 Game Tracking**

GameEye will provide a personal watchlist or list of video games an individual wants to stay updated about. In the list there will be a thumbnail displayed next to each game title in the list. The list will show the new game updates and will be organized by important updates, news articles, tweets, reddit posts, images, and videos. Thumbnails will also be included for tweets and news articles. There will also be a separate list of games featuring the most watched games on the platform.

### **2.1.5 Searching**

Users will be able to search for video games based on their title and the search system will feature support for auto completion.

### **2.1.6 Web Scraping**

Multiple sources will be aggregated to provide game development updates. GameEye will utilize web scraping in order to aggregate news articles from video game news websites, tweets from official Twitter feeds, Reddit posts from official game subreddits, videos from official game YouTube channels, as well as images.

### **2.1.7 Notifications**

Cross-platform push-notifications will be used to provide users with new game updates. A UI count of notifications will be displayed for each game and each resource category. The

user will also be notified of suggested video games to follow based on the most-watched games list.

### **2.1.8 Settings**

Users will be provided with options with their GameEye experience. Users may choose if they wish to see archived resources, see the importance scores of news articles and tweets, if they want to receive notifications and which categories to be notified about. Users can also submit feedback

### **2.1.9 Machine Learning**

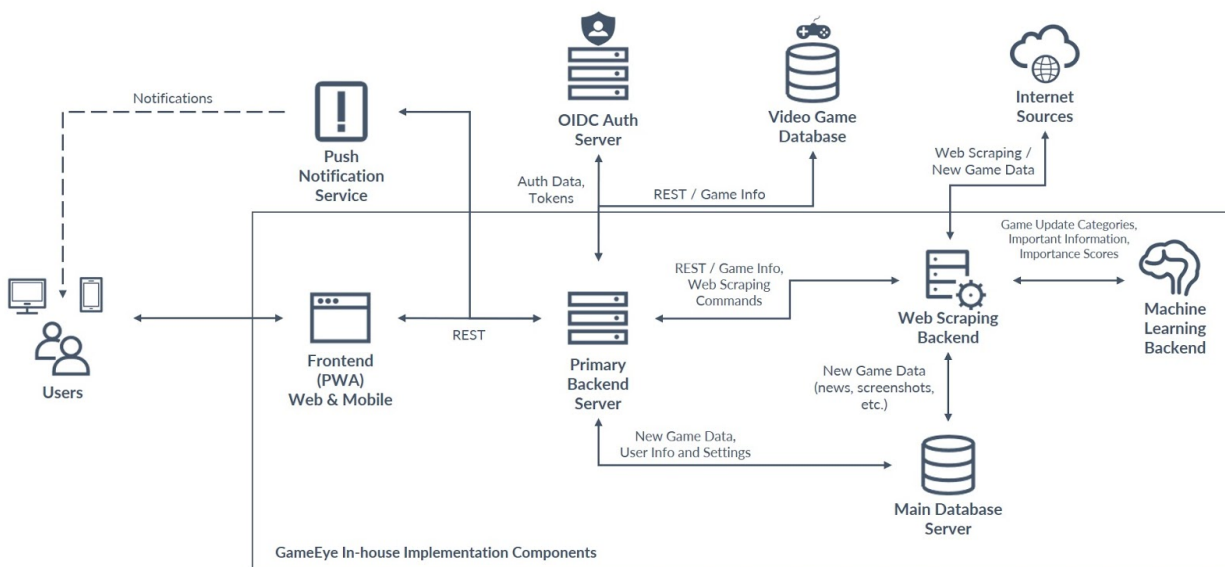
Machine learning will be implemented in order to provide game update classification such as release date announcement, delay, major game update, or minor game update. Importance scoring will be assigned for news articles and tweets utilizing machine learning. Important information will also be extracted from news articles and tweets.

## **2.2 Major Components**

### **2.2.1 Hardware**

GameEye will work by utilizing many components with them being a Frontend server, main backend server, web scraping backend server, machine learning backend server, and a main database server. In figure 2 on the next page, the diagram shows how the components are connected together and the objects within the square shows GameEye's in-house implementation components. Outside the square show the external components such as the push notification service, video game database, internet sources, and OIDC Authorization server

[THIS SPACE INTENTIONALLY LEFT BLANK]



*Figure 3: GameEye's Major Functional Component Diagram*

The front end server will provide the frontend of the web application for the user to see. The main backend server will expose a REST API for communication with the frontend, web scraping backend, databases, and the other third-party services . It also coordinates web scraping backends and powers the dynamic functionality of the frontend. The web scraping backend server is also an external-facing web server and it exposes a REST API for launching web scraping operations. It communicates with the machine learning backend to classify scraped video game data and extract important information before putting it in the database. The machine learning backend server is an internal GPU-powered TensorFlow serving server exposing a REST API for machine learning inference. It communicates with the web scraping backend to classify video game data and extract important information. The main database server is an internal MongoDB database server that stores all data related to the application.

[THIS SPACE INTENTIONALLY LEFT BLANK]



### 2.2.2 Software



*Figure 4: GameEye's Software Components Diagram*

The frontend of GameEye will be developed using WebStorm IDE and Angular Framework to build the UI. Google Workbox will be implemented to provide GameEye with offline support. The frontend of GameEye will be written in HTML, SASS/CSS, and TypeScript.

The backend of GameEye will be developed using IntelliJ IDEA IDE and Spring Framework. The Jsoup library will be used to provide the web scraping capabilities and MongoDB Java Driver will be used to build the database.

Testing of GameEye will be provided by utilizing JUnit Java Framework to test the java code and Jest JavaScript Framework to test the JavaScript.

GameEye will use Python to code the machine learning aspect and will be hosted using the tensorflow serving (model server). The Keras and scikit-learn python libraries will be used to code the machine learning aspects.

Natural Language processing will be a part of GameEye as it will allow the web scraped information to be processed, categorized, and assigned an importance score. Python will once again be used to implement this feature and the spaCy library will be used to do so.

Databases will be constructed using MongoDB and MongoDB Compass(GUI). The database will be populated using the IGDB REST API to retrieve data on known video games.

Some third party software will be used such as Auth0 Single Page App SDK to provide the two-factor authentication. Firebase Cloud Messaging will be utilized to provide the users with push notifications to their desktop or mobile device.

### **3. Case Study**

GameEye is being developed with gamers in mind. There are an estimated 2.47 billion gamers around the world (Gough, 2019), so GameEye would cater to some of those gamers. GameEye would specifically be used by gamers in order to stay-up-to date on games through update notifications and be used for tracking games. The video game industry is a growing industry and will be an estimated 196 billion in 2022(Anderton, 2019), this shows that GameEye's user base will increase as a result of new games and new gamers. Stakeholders of GameEye also include video game news sites (e.g. IGN, Gamespot, etc.) and video game developers. Video game news sites could see an increase in web traffic because GameEye links the articles to the source websites and displays a news sites logo next to each scraped article. Video game developers on the other hand may have an easier time maintaining interest or hype around their developing games seeing as how saturated video game selections have become.

[THIS SPACE INTENTIONALLY LEFT BLANK]

## **4. Product Prototype Description**

### **4.1 Prototype Feature and Capabilities**

Although the real world product is ambitious, the prototype will still provide various features. The prototype will be able to provide a robust user authentication and external provider support for logging in. There will still be necessary account management features for users to change their passwords and modify their profile information. When game searching there will be an auto completion feature. There will still be a personal game watchlists and web scraping for news articles and tweets. GameEye will still have redirection to sources upon clicking scraped resources. There will be a most watched games list to promote the games. Regarding settings there will be the availability to toggle visibility of archive sources, visibility of importance scores, and for notifications. There will be classification and importance scoring of news articles and tweets using machine learning. The GameEye prototype will still feature cross-platform push-notifications and UI count of notifications for new game updates.

The provided features will demonstrate that GameEye can successfully do many tasks. It will be able to scrape data from multiple online sources and for multiple different resource categories. It will successfully provide robust authentication mechanisms for users, robust game searching and fast autocomplete, each user a personal watchlist, a list of most watched games on the platform, customization capabilities regarding content and notifications, and cross-platform support and push-notifications. The prototype will demonstrate that it can use machine learning to classify news articles and tweets as well as score their importance.

There are some risks that can easily be mitigated such as scraping RSS feeds to protect against source website structure changes which would cause web scraping to fail. GameEye would store video game database content on its own database for redundancy. The platform would be made scalable to protect against high load. Content would be cached on a user's

device to prevent blank application in case the main database fails. The platform would use database encryption and 3rd party authentication providers to safeguard user personally identifiable information. A manual will be provided to help users understand the application. Lastly developers will use proper coding practices to prevent No SQL injection, XSS, or data exposure.

#### **4.2 Prototype Architecture**

The prototype will utilize the same architecture as the real-world product with a few exceptions; it will not use replicated servers for load balancing. Some features will not be implemented or will be partially implemented. Web scraping of resource thumbnails tweets and news articles will be partially implemented, but scraping of images, videos, and reddit posts will not be implemented. Users will not have the option for two-factor authorization. There will be only partial functionality for offline support, local caching, and connectivity interruption resiliency. Users will not be able to submit feedback or receive suggested video game notifications. Importance scoring and resource classification will also be partially implemented.

#### **4.3 Prototype Development Challenges**

There will be many hurdles and underlying circumstances to overcome when developing the prototype. The team will be required to learn new technology and frameworks required for implementation. Data will need to be collected and labeled for the machine learning models. There may be complications for properly securing communication between the multiple backends. There is also a challenge in implementing robust caching in the frontend content for connectivity interruption resiliency and performance. Lastly there is also the trouble of implementing robust web scraping with future-proof mechanisms when source websites change.

## 5. Glossary

List of terms and abbreviations:

**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.

**PWA:** Progressive Web Application; 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:** 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.

## 6. References

Anderton, K. (2019, June 26). The Business Of Video Games: Market Share For Gaming Platforms in 2019 [Infographic].

<https://www.forbes.com/sites/kevinanderton/2019/06/26/the-business-of-video-games-market-share-for-gaming-platforms-in-2019-infographic/#c1793427b254>

54

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](https://www.gamasutra.com/view/news/217583/How_the_surge_of_Steam_releases_will_affect_game_developers.php)