Lab 1 – Gameye Product Description

Ben Hund

Old Dominion University

CS 111

Janet Brunelle

May 19 2020

Table of Contents

1. Introduction

1.1 The Societal Problem

As the video game industry has grown, its market has become oversaturated with releases[1][2]. This can cause a variety of negative effects. It may cause customers to loose track of a release date as a result of delays.

Video games frequently make use of updates. They may resolves glitches that can cause some serious problems to the viability of the product[7]. Another type of bug is one that causes a story to stop progressing, stopping all progress[7]. A bug related to online games is matchmaking algorithms being unable to allow a customer to play the game they payed for. A more minor bug would be the bugs that break player emersion such as players moving through objects[7]. These could be deal-breakers for a customer, and therefore, something the customer would like to be made aware of should the glitch be resolved.

A game may release more content only for the customers to enjoy the content out of ignorance. Many updates are expansions, which are a form of downloadable content that is purchased. As time goes on, the “games as a service” model, which involves the use of free, continuous content updates with other revenue models become more common. These also tend to have a substantially greater revenue source for publishers, which effectively ensures that the practice will be used more[8]. Another type of content that a customer may miss are events within the game itself.

Customers may get blindsided by a publisher should they not be aware of their reputation. A game may be of high quality, but include predatory practices a customer might want to be informed for. This may not be the focus of reviews, but is very relevant for players if one were to go by a critic score vs a user score.[4] A game may start out user-friendly, but retroactively add in predatory elements that wouldn't be included in reviews.[3]

2. GamEye Description

GamEye is an app that will act as a time-saver for users who wish to monitor information for games. A user will set up their account to monitor games they wish to receive information on. GamEye will then update the user on updates and events related to their chosen games. The user then does not have to manually skim through multiple sources of information for video games.

2.1. Key Product Features and Capabilities

GamEye will follow/watch video games. This feature will allow the user to customize a list of games the user wishes to receive updates on. This gives the user an autonomous update feed. This cuts down on the time required to individually search for game information desired by the user.

GamEye will categorize information about specific video games on the web. This allows the user to not just choose what game the user wishes to follow, but what type of information the user desires to be kept up to date on. GamEye filters out unimportant content. GamEye designates an importance score according to the preferences set by the user. A user can also set an ignore list. For instance, if a user decides that he doesn't want to follow fanart or popular discussion of a game, he can set the GamEye filter to either stress less importance on said content, or hide it completely.

GamEye will notify users in response to news articles. Should a game that's being followed receive articles containing new information, the user will receive a notification. This cuts down on the time of a user browsing through multiple news sites. GamEye will notify users of expansion releases. Many games have paid updates that add new content to a game. This will allow the users to be aware of the release of an expansion in a game that they like even if the game is not marketed well.

GamEye will notify users of content updates. Games often update the content of a video game. Notifying the users of a content update not only potentially saves time for a user, but also gives the user the security of knowing when it's been released and the option to play day 1. GamEye will notify users about date delays. Many games have development troubles that causes delays. These wreaks havoc on a user's perception. Updating users on delays in real time allows them to be firmly aware of these changes. GamEye will notify users about game patches. Many games suffer from glitches that can be anything from minor annoyances to game-killers. Should a game have a glitch that's a deal-breaker for a user, he can return to the game later after receiving a notification that the glitch was fixed.

GamEye will direct users to sources of original content. A user might want to know the citation for the information source. GamEye will give the user access to the original news source. There, they can read about it more in depth. GamEye will search for video games using a search bar with support for auto-completion. This allows for a user to find a game that they wish to follow.

GamEye will display top games watched by users. This allows the user to view the most popular games on the app. This potentially allows the user to discover games that they might devlope an interest in.

GamEye will retrieve game info from a video game database. This will act as a huge source of information for GamEye to operate on so that it can give accurate answers to the users. GamEye will web scrape gaming news sites. This will allow GamEye to determine which news sites contain the information the user desires, and therefore, where to send them.   GamEye will web scrape gaming youtube channels. Youtubers often engage with video games. They, themselves, often cover games, as well as play through early builds. Therefore, it would be of great benefit to the user to search through youtube content for information. GamEye will web scrape relevant twitter feeds. Developers and publishers are amongst the twitter userbase. They use social media to engage with their customer base. Scraping twitter can result in game info to be revealed to the user that only the developers can be a source of.   GamEye will web scrape relevant Reddit posts. Reddit is a blend of social media and forums. This is another way in which developers and publishers interact with customers, and can also be used as a source of information.

2.2 Major Components (Hardware/Software)

This will involve a push notification service, a primary backend server, an authentication server, a backup server, two databases, one being the main database, the other for videogame info, and a backend.

The frontend software will make use of HTML, SASS/CSS, and Typescript. The backend software will involve the IntelliJ IDEA IDE, the Spring Framework, the jsoup library, and the Mongo DP Java Driver. The databases will be from MongoDB, MongoDB compass, and IGDB REST API. The JUNIT framework and the Jest Framework will be used for testing. The machine learning will be utilized with Python, Keras, the scikit-learn library, and TensorFlow Serving.

3. Case Studies

This product is intended for a gamer that desires to keep up to date on developments on his or her favorite games. There are two main scenarios.

Scenario one. A customer knows that content is coming for a popular game, but doesn't want to consistently check specifically for the game. He sets GamEye to give him news about updates on the game.

Scenario two. A customer finds an indie game that doesn't have much of a marketing budget. The game's unique enough that he wants to keep up to date with it. He then has GamEye notify him on the game's development cycle.

4 GamEye Prototype Description

GamEye's prototype will implement much of the features of the final product. It will partially support offline and local features. It will either fully support or partially support authentication features with the exception of two-factor authentication. All account managing features will be implemented except account deletion. The function to search for games will be partially implemented. GamEye's tracking systems will be largely implemented. However, the web scaping will not be utilized for redit posts, images, or videos. The only settings not implemented will be the feedback, as there's no customer in this scenario. Machine learning will be partially implemented. The only notifications that won't be fully implemented are recommended games.

Category	Feature	RWP	Prototype
General			
	Cross-Platform Support (Desktop, Mobile)	Full Functionality	Full Functionality
	Offline Support	Full Functionality	Partial Functionality
	Local Caching	Full Functionality	Partial Functionality
	Connection Interruption Resiliency	Full Functionality	Partial Functionality
Authentication			

	User Login & Registration	Full Functionality	Full Functionality
	External Provider Login & Registration	Full Functionality	Partial Functionality
	Persistent Sessions	Full Functionality	Full Functionality
	Two-Factor Authentication (2FA)	Full Functionality	No Functionality
	Password Recovery	Full Functionality	Full Functionality
Account Management			
	Change Password	Full Functionality	Full Functionality
	Modify Profile Information	Full Functionality	Full Functionality
	Delete Account	Full Functionality	No Functionality
Searching			

	Search for Games	Full Functionality	Partial Functionality
	Search Autocompletion	Full Functionality	Full Functionality
Game Tracking			
	Add Games to Watchlist	Full Functionality	Partial Functionality
Category	Feature	RWP	Prototype
	News Articles (Web Scraping)	Full Functionality	Partial Functionality
	Tweets (Web Scraping)	Full Functionality	Partial Functionality
	Reddit Posts (Web Scraping)	Full Functionality	No Functionality
	Images (Web Scraping)	Full Functionality	No Functionality
	Videos (Web Scraping)	Full Functionality	No Functionality

	Resource Thumbnails (Includes Website Logos)	Full Functionality	Partial Functionality
	Game Thumbnails	Full Functionality	Full Functionality
	Source Website Redirection	Full Functionality	Partial Functionality
	Resource Organization	Full Functionality	Full Functionality
	Archived Resources	Full Functionality	Full Functionality
	Most-Watched Games List	Full Functionality	Full Functionality
Settings			
	Show Archived Resources Option	Full Functionality	Full Functionality

	Show Importance Scores Option	Full Functionality	Full Functionality
	Receive Notifications Option	Full Functionality	Full Functionality
	Receive Notifications Per Category Options	Full Functionality	Full Functionality
	Submit Feedback	Full Functionality	No Functionality
Machine Learning		🟩	🟧
	Importance Scoring	Full Functionality	Partial Functionality
	Resource Classification	Full Functionality	Partial Functionality
	Important Information Extraction	Full Functionality	No Functionality

Category	Feature	RWP	Prototype
Notifications			
	Push-Notifications for New Game Updates	Full Functionality	Full Functionality
	UI Count of Notifications for Each Game	Full Functionality	Full Functionality
	UI Count of Notifications for Each Resource Category	Full Functionality	Full Functionality

	Cross-Platform Notifications	Full Functionality	Full Functionality
	Suggested Video Game Notifications	Full Functionality	No Functionality

4.1. Prototype Architecture (Hardware/Software)

The prototype will utilize a an external-facing web-server as the frontend of the web application. It will also use an external-facing web-server that blocks incoming traffic to utilize web scrapping alongside machine learning. There will be a server dedicated to machine learning itself. There will be a database server. Holding all of these together will be a backend server that coordinates all of the communications between the servers.

The frontend will make use of WebStorm IDE, Angular Framework, and Google Workbox. The languages being utilized will be HTML, SASS/CSS, and Typescript. The backend will utilize IntelliJ IDEA IDE, Spring Framework, the jsoup library, and a MongoDB Java Driver. The databases make use of MongoDB, MongoDB Compass (MongoDB GUI), IGDB REST API. The testing will make use of JUNIT and the Jest JavaScript Framework. The machine learning will be implemented with Python, Keras, a script-learn library, and TensorFlow serving. The natural language processing will be utilized with Python, and spaCy library. Other software used from 3rd parties would be Auth0 Single Page App SDK, and Firebase Cloud Messaging.

4.2 Prototype Features

GamEye's prototype will test many of the basic elements of the real product. These would include a web scrapping algorithm various settings, authentication, and notifications. It will be tested by both fake data on games that we provide, and actual games. We have the capability to test some technical and most security risks.

4.3 Prototype Challenges

We may have issues setting up the various servers depending on cost. Finances are a very real obstacle. This project also requires a variety of skills in order to even attempt. We must learn how to utilize machine learning, web-scrapping, database management, and to an extent, how to hack to test the security.

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 creation of applications that access the features or data of an operating system, application, or other service.

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 financial support of larger game publishers.

IntelliJ Idea: IDE developed by JetBrains for the purpose of writing Java 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: Testing framework for test-driven development.

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 Feeds: 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 for the purpose of writing JavaScript.

6. References

[1]

Gough, C. (2019, August 9). Number of games released on Steam 2018. Retrieved January 30, 2020, from https://www.statista.com/statistics/552623/number-games-released-steam/

[2] Gough, C. (2019, October 9). Google Play: Number of available games by quarter 2019. Retrieved January 30, 2020, from https://www.statista.com/statistics/780229/number-of-available-gaming-apps-in-the-google-play-store-quarter/

[3] Humphries, M. (2019, September 18). Twitch Acquires Gaming Database Website IGDB. Retrieved January 30, 2020, from https://www.pcmag.com/news/twitch-acquires-gaming-database-website-igdb

[4] Birch, N. (2019, February 20). Call of Duty: Black Ops 4 Adds Loot Boxes That Include "Signature Weapons" With XP Boosts. Retrieved from https://wccftech.com/cod-black-ops-4-loot-boxes-weapons/

[5] Star Wars Battlefront II. (2017, November 14). Retrieved from https://www.metacritic.com/game/playstation-4/star-wars-battlefront-ii

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

[7]ronswift9. (2014, February 13). Video Game Bug Types. Retrieved from https://ogcardcaptor.wordpress.com/2014/02/13/video-game-bug-types/

[8]Saed, S. (2017, October 11). The games as a service trend has "tripled" the industry's value - report. Retrieved from https://www.vg247.com/2017/10/11/the-games-as-a-service-trend-has-tripled-the-industrys-value-report/

[9]Smith, C. (2020, April 17). Valorant: Is its anti-cheat client malware? Riot Games respond to accusations on Reddit. Retrieved from https://www.hitc.com/en-gb/2020/04/17/valorant-anti-cheat-client-malware-reddit-riot-games/