*Department of Computer Science & Engineering*

*University Of Nevada-Reno*

Project Glimmer

Team 15

**Mike DesRoches**

**Christian Garcia**

**Jake Lahr**

**Alex Pasinski**

Instructors

**Dr. David Feil-Seifer**

**Devrin Lee**

External Advisor

(TBD)**\***

October 11, 2019

**To:** CS425

**From:** Team 15 - Mike DesRoches, Christian Garcia, Jake Lahr, Alex Pasinski

**Date:** October 11, 2019

**Subject:** Project Assignment 1

**Abstract**

Group 15 aims to create a software API (codenamed ‘Project Glimmer’) that game developers can use to process comments entered in a chat log on a given web video streaming service (such as Twitch.tv). The intended use of this API between the stream chat and the developers’ code is to allow for the developers design commands that change the state of their games and which can be called from a stream chat. The type of interaction we are attempting to facilitate in video games can be referred to as a one-to-many player game (OTMP). In an OTMP game, a single player streams live footage of their game to an online video streaming service. The many viewers of that stream then interact by sending commands through the stream chat that modify the game state in some way.

**Project Description**

*Main goals, objectives, and significance*

Project Glimmer’s objective is to create an application programming interface (API) that game developers can utilize to create audience interaction between players live streaming their games and viewers watching those lives streams. Live stream viewers would be able to interact with game streamers in a real-time fashion and have an impact on the current game state. Our objective is that, by integrating our API into a game, the life-span of a video game can be increased by the increase in possible interactions from external sources such as live stream viewers. Our main software design goal is to create an API that game developers can easily integrate with their existing software. Our product will have a unique significance in the video game development industry as, while live stream chat integration in video games is not a completely new concept, most solutions need to be developed in house and few resources exist to allow independent developers the ability to easily achieve this type of interaction in their games.

*Main functionality and characteristics*

The team will work using a Scrum-based development cycle to facilitate rapid iteration of the product. The user stories for our product may change many times throughout development and we do not want to accidentally spend months working on features only to find out that those features were a waste of time. We plan to use a weekly meeting structure that is divided into half in person meetings and development time and half individual development time in a member’s preferred space (at home, a park, etc.)

*Intended audience*

The API would be used by game developers to integrate with their video games. So, developers across all platforms would be our target audience to start. However, for the prototype stage of the development cycle, the API will be developed mainly for Linux based operating systems as those are what the team members are most familiar with programming for.

*Key usability goals*

Game developers would be able to add another level of user interaction between players and possible live stream viewers through the use of our API. The API must be easy to integrate with existing code and so should assume as little as possible about the video game that it might be being used in. For the prototype, the current complexity of the commands possible will be limited to single word commands without arguments.

*Potential for further development/product enhancements*

If time allows for it, we plan to add the ability to specify special commands and interaction types that go beyond the simple commands planned for the prototype (such as commands with arguments). We may also look into adding the ability for multiple players in a local or online multiplayer setting to live stream the same game to their own individual streaming audiences. This would then allow for streaming audiences to participate in online multiplayer matches with potentially thousands of individual players without the negative downsides to having to host thousands of game clients.

*Challenges and obstacles*

Marketing is Project Glimmer’s big obstacle. Getting in contact with game developers and selling a functioning prototype will be one of our biggest challenges. Many game developer use in house game engines that we cannot possibly test our API on ahead of time. Attempting to expand our API to several platforms will also be another hurdle. On the team management side, we have a steep learning curve to overcome with regards to the business aspects of creating software.

*Technology description*

We plan to program the API primarily in C++. This may change as we attempt to work with the networking aspects of our project and we may choose to use Python for those parts as it has extensive libraries to help with network programming. For our prototype, we will only be integrating with the Twitch.tv streaming service. This is one of the most popular streaming services for video games and will allow us to reach a large enough audience of game developers interested in our product. We will also either use the Unity Game Engine or build our own small game engine to create a game to demonstrate the features of our prototype.

*Team overview*

Michael DesRoches

11 year Helicopter mechanic/Crew-chief for the Army National Guard. Platoon leader and quality control inspector of high caliber maintenance and inspections. Airframe and powerplant license holder and a new intern at Tesla gigafactory. Have taken CS381, game engine architecture and CS302, Data structures. Expected contribution to the project is to be the main speaker, develop assigned coding tasks, and work side by side with project lead, Alex.

Christian Garcia

I love video games and I hope to work in the gaming industry after graduation. I have some experience in graphics and game development. I’m relatively new to Twitch, but I love the thought of viewer interactive games and I can’t wait to be a part of it. I expect to contribute to all parts of this project.

Jake Lahr

I am an experienced Twitch user and a big fan of video games. I have experience in graphics programming, game development with the Unity game engine, and have been an intern at Scientific Games. I expect to help with all aspects of the project, but focusing on either the graphics side or Twitch chat bot integration.

Alex Pasinski

I have been programming for almost eight years now and tinkering, so far unsuccessfully, with game development for about half of those. I enjoy learning about how complex software systems can be built to work in harmony, or at least an approximation of harmony. I specialize in graphics and game engine programming. I expect my main contribution to be designing how the various systems we will implement will work together. I have no network programming experience but am looking forward to learning as much as I can about the field.

*Advisory overview\**

Currently, we are pursuing options for advisors in the Computer Science and Engineering department. The two professors who are most likely to work well with our team and our project are Dr. Alireza Tavakkoli and Dr. Sushil Louis. Two of our group members have experience working with Dr. Sushil in his CS 381 Game Engines course and we were also referred to Dr. Tavakoli by Dr. Sushil due to his knowledge and expertise on game and game engine development.

*Professional Growth*

Throughout this project, our team members will need to work together to solve difficult problems that none of us have had to work on before. We will have to come up with unique software solutions (for us) and continuously iterate over our designs. The latter challenge being one that most of us have not had to do so far in our college careers as most homework assignments can be completed independently from other ones.

**Market Potential or Open Source Significance**

*Market Analysis*

The market potential for a project like this is big, whether team 15 decides to make this project open source or sell the product. Today’s game developers are realizing the potential for live stream integration based on how big and influential streaming services, such as Twitch, have become for video games. Developers can now reach huge audiences faster than before, and with this integration API, their games will become even more popular with both the streamers and viewers.

*Competitive Analysis*

There are no obvious alternatives already available on the market as most solutions relating to the problem of live stream integration are developed in-house and not open sourced to the larger game development industry. However, products such as ours can provide a large competitive advantage to games that use them. According to an interview with Warwitch, a Twitch streamer, both streamers and viewers are excited about the idea of games that change based on viewer input. The interview revolved around a game called Warhammer: Vermintide 2, where the developers put in options for viewers to vote on what change happens next in the game. The options include things like character buffs/debuffs, bosses, or hordes of enemies. In the interview Warwitch said “When my viewers are integrated into the game, I find it improves every element of my stream. I always try to make my viewers feel that they are a real part of the show (not just spectators), and integration really helps to make that happen.” While the developers were able to integrate these ideas into their game, it is only a part of their game engine and not readily available for other developers.

*Competitive Advantage*

Twitch provides options for this type of integration through reading the chat, but there are not any easy to use libraries or API’s to achieve this functionality. Therefore, developers have to do all of this work themselves. Our product aims to give developers an easy to use option to integrate viewer involvement into their games. While live streaming is not a completely new form of entertainment, our idea for integrating it into games shows promise as a relatively untapped market by which developers can promote their own video games. There are also little to no other competing products out yet. Creating a stable platform while these concepts are still new gives our solution a great competitive advantage.

**Time Worked on Project**

*Michael DesRoches*

Worked a total of 4 hours on this project, completed most of section 2, and wrote a personal overview.

*Jake Lahr*

Worked a total of 4 hours on this project, completed section 0 and 1, and wrote a personal overview.

*Christian Garcia*

Worked a total of 4 hours of this project, completed section 3 and a personal overview.

*Alex Pasinski*

Worked a total of 4 hours and completed the rest of section 2, edited document for each section before creating the final draft, and wrote a personal overview.

**References**

[**https://blog.twitch.tv/en/2019/03/08/a-twitch-integration-lets-the-viewers-call-the-shots-b5eae972d3ef/**](https://blog.twitch.tv/en/2019/03/08/a-twitch-integration-lets-the-viewers-call-the-shots-b5eae972d3ef/)