
SOUNDS OF
RUNETERRA

Table of Contents

1. INTRODUCTION

Welcome to Sounds of Runeterra, the Legends of Runeterra extension that makes the game more accessible to visually impaired gamers.

We are a group of friends who, for more than 10 years already, have been enjoying the wonderful world of videogames together. Over the years, we have run into people having difficulties when playing. The issues they faced ranged wide, from people that couldn't hear properly, to others who were colour blind or suffered from vision loss.

Due to this, we thought it would be wonderful if we could take this up-and-coming card game and make it more inclusive and approachable for the players that need it.

Ultimately, we want every person to have an equal opportunity and enjoy videogames as much as we do. This project is a first step towards that goal. Our plan is straightforward, create an audio assistant that makes the players' gaming experience as pleasant as possible.

We are hyped by the opportunity you bring us to participate in this exciting project and contribute to making this amazing community, just a little bit better.

2. OBJECTIVES

Our main intention with this project is to create an easily pluggable extension that serves as a companion for the gamers and makes up for their lack of vision.

Sounds of Runeterra will play, on demand, the description of any card that can be found in the match. This could be done in two ways, playing the card that is being hovered by the player or describing the cards in each of the play zones, both when pressing a shortcut.

Enhancing the apex of the match is also important. Whenever a match commences, a brief introduction telling the players' names will be reproduced, as well as the game result when the match is over.

We also want to give the players the ability to adjust the verbosity level of the played audios. A shortcut that switches between a concise and long explanation could be a simple way to solve this.

To successfully develop such an application there are several challenges we must properly tackle:

2.1. External API

Dependencies with external tools usually add a level of complexity to whatever software you are creating.

We must carefully analyse the API to assess what data is available, what types of data we will have to deal with, whether there's throttling or a higher latency than expected, etc.

2.2. Responsiveness

When working with on-demand features and high-latency tasks it's important to give some love to the app responsiveness.

There's a big chain of events that happens between the player's request and the output. They can't be delayed, as that would defeat the purpose of being on-demand, and some of them are not local, which makes this process susceptible of communication latency.

2.3. Player Needs

Each player's needs are different, and even those needs vary on a regular basis. In this case, forcing the software to always play long-winded descriptions of a card when it's not desired could annoy the user and maybe even be the reason of a loss.

Allowing the player to easily switch between the different verbosity levels should be one available feature.

3. DECISION MAKING

Some of the decisions made prior and during the development of the software were based on the hackathon spirit of the challenge.

Those choices moved us forward towards the completion of the MVP and didn't punish the experience of the players. Not refreshing the local game card database whenever the extension started or assuming every player has only one monitor and plays at full screen are examples of that.

Of course, given we worked with an iterative development mindset, those changes just meant setting a realistic deadline for the project. We developed the app core focusing on creating a strong and reliable foundation, that makes the task of building upon it painless.

3.1. Architecture

There are two key concepts that drove the architecture of this project: Asynchronous programming and subscription-based events.

The combination of those two laid a good infrastructure to support the on-demand nature of the app. Additionally, it improved the processor resource usage avoiding downtimes due to blocking calls.

3.2. Program Flow

Once the app is run it stays idle, polling the Legends of Runeterra server to retrieve the game state.

Once it detects the player has started a match it plays an intro audio, and the mouse and keyboard events are subscribed to its defined key.

During the match, whenever the player presses one of the bound keys, the corresponding event is triggered. Those events either play the description of a single card, a group of cards or switch the verbosity level.

Whenever the event of playing a card (or group of cards) description is triggered, there starts a chain of processes that involve detecting the card that is being hovered by the mouse or inside a zone, fetching its data from its card code, synthesising the desired description into an audio buffer and playing it.

Once the match ends, and the player returns to the game menu, the events are unsubscribed to avoid unpredicted behaviour.

Listening to keyboard and mouse events, retrieving the card position and id, fetching its data, synthesising the text and playing it

3.3. Card Detection

We decided there were two main options that trigger the playing of the card descriptions:

- When the player pressed a key while hovering over a card.
- When the player wants to know what cards are in a specific game field. We defined different fields in the game: hand, played cards and battlefield, for each of the players.

To enable the second option, we had to divide the game field into 6 segments and detect which cards are in which segment. Each field is bound to a key from F1 to F6 in order of:

1. Cards in the player hand
2. Cards the player played
3. Cards in the player battlefield
4. Cards in the opponent battlefield
5. Cards the opponent played
6. Cards visible in the opponent's hand

4. ENCOUNTERED OBSTACLES

During the development of this project we came across several circumstances that delayed the development or that made us rethink the architecture and features we had planned.

We believe listing them helps us gather insight into what we could have done to improve the project and our efficiency.

4.1. API

The current API version didn't provide us with all the data and tools we would have desired for our tasks. Anyhow, as we previously mentioned, we understand one of the challenges was to cleverly circumvent the API limitations, and we really enjoyed doing so.

We hope to update our application when future API versions are released.

4.2. Working hours

We didn't realise we had been selected to participate in this challenge until 3 days to the deadline. This made us rethink some of the planning and created some availability issues among some team members.

5. FUTURE PLANS

When we were brainstorming what ideas we could submit to the challenge, we came up with a lot of interesting features. We soon realised attempting to develop all of them would be unfeasible.

Those features are listed below.

5.1. User Interface

Create a user interface would allow our users to easily adjust the volume, edit the shortcuts and even customise the voice the software uses.

5.2. Voice Commands

The next step towards providing a full game experience for visually impaired people, would be to create a voice command system using a speech recognition library.

With such library and controlling the mouse, we could create voice commands that are linked to game actions such as passing a turn, playing a card and even using emojis.

5.3. More Actions

This one's simple, developing the events system further to provide a smoother experience for gamers' needs.

5.4. Expeditions

When the expedition endpoint returns the cards available for drafting in the expedition, we would love to extend the audio assistant to support this gamemode.

6. CONCLUSIONS

First off, we want to sincerely thank you for being so engaging with this community. We appreciate the opportunity you bring us to participate in this game together with you.

There are many reasons why we wanted to join in on this. We are fellow developers and we have been enjoying League for a long time, but most of all we saw the opportunity to endorse and even push forward the ethical values we want this community to be proud of.

Us gamers may be competitive, but we are always there when someone needs it, and there has been countless proof of that over the years.

Thank you again for organising events such as this, we are looking forward for the next one.