

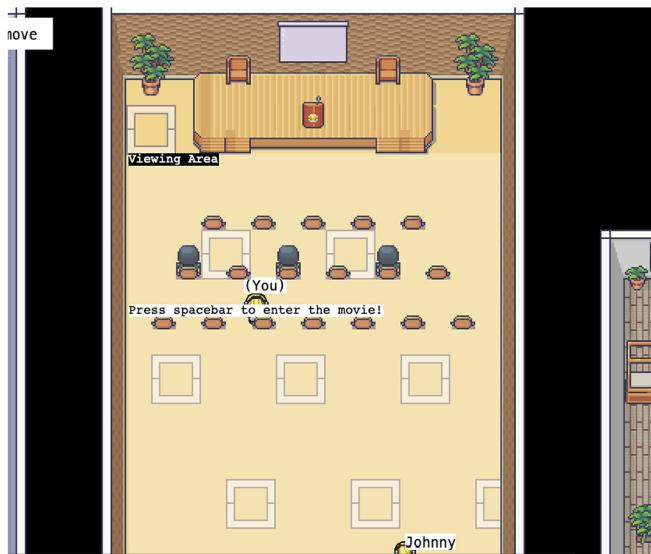
CS 4530 Final Project: “Viewing Area”

Group 2M: Aamir Islam, Jonathan Maduro, Jonathan Ju, Petros Papadopoulos

Our Feature: Viewing Area

In a real life communal space, people often congregate not only to chat, but also to watch movies, TV shows, and other videos together for the purpose of entertainment, sharing learning content, or showing off a video created by someone in the space. Since Covey.Town is a place for people to commune virtually and converse, it follows naturally that Covey.Town should support communal viewing parties as another way to interact and share.

It is for these reasons that we developed the Viewing Area feature for our term project. The Viewing Area is a part of the map that users can enter and watch a YouTube video of their choosing together, in synchrony (at the same time). Previously, if a covey.town user wanted to share a video with other users in their server and watch it together, they would have to tell the other viewers to open another tab in their browser and sync up the videos manually, or use another platform which supports synchronous video watching. Now, users can enter a covey.town, walk over to the conference room labeled Viewing Area, type and submit a YouTube video url, and enjoy watching the video synchronously with the other users in the Viewing Area on the same covey.town server. If the video is paused, changed, or fast forwarded or rewinded this change will be reflected in real time to everyone.



Users enter a Viewing Area just as they would a Conversation Area. If they click the space bar, it opens a modal

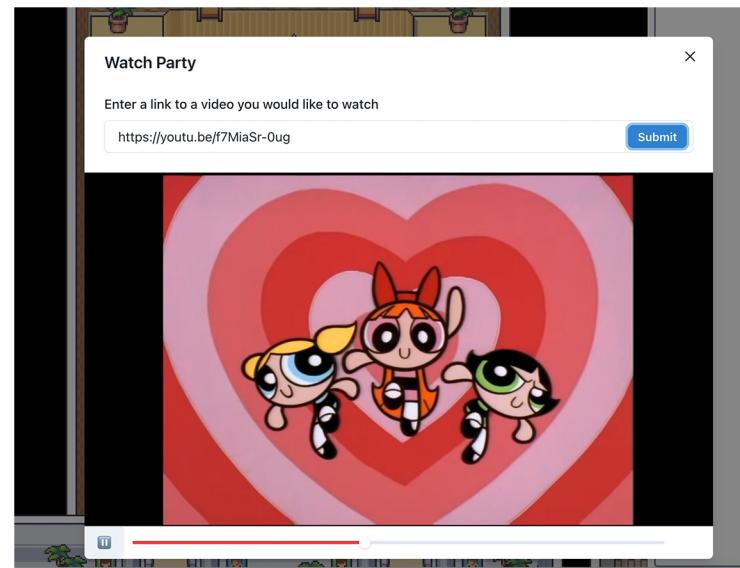
Technology Stack & Design

We implemented the viewing area feature in the existing Covey.Town codebase. There exists a single viewing area represented as an “object” in the tilemap. The object is dynamically constructed when the map is loaded, and rendered on the screen by Phaser.

When a player enters a “new” viewing area, the message “Press spacebar to enter the movie!” is displayed via a Phaser text game object. A React/Chakra modal then appears, with a React/Chakra form component with a submit button embedded within it, inviting the user to enter a YouTube video URL. Once the user submits a valid URL(our MVP supports YouTube videos only), a video appears with a pause/play button built using a Chakra button component and video progress bar built using a Chakra slider component. The embedded video is built using a ReactPlayer component, a react component which renders a video embed.

These viewing area components are defined in the ViewingArea directory in the frontend, instantiated within WorldMap, and each have hooks linking user interaction to the video status in the backend. The video status is tracked by the CoveyTownController, represented as a VideoStatus type, and synced to each client using socket.io. Any interaction with the viewing area that involves a request to change the status of the video (new url, pause status, elapsed time of the video) will be validated, updated in the backend, and then the change is propagated to every client in the same server. The onVideoStatusUpdated listener is called within updateVideoStatus by the CoveyTownController to propagate any changes made to the video status.

Once a VideoStatus is set in the CoveyTownController, it will automatically increment the video’s elapsed time by one second so long as the VideoStatus is not in a paused state and the elapsed time is not equal to the length of the video.

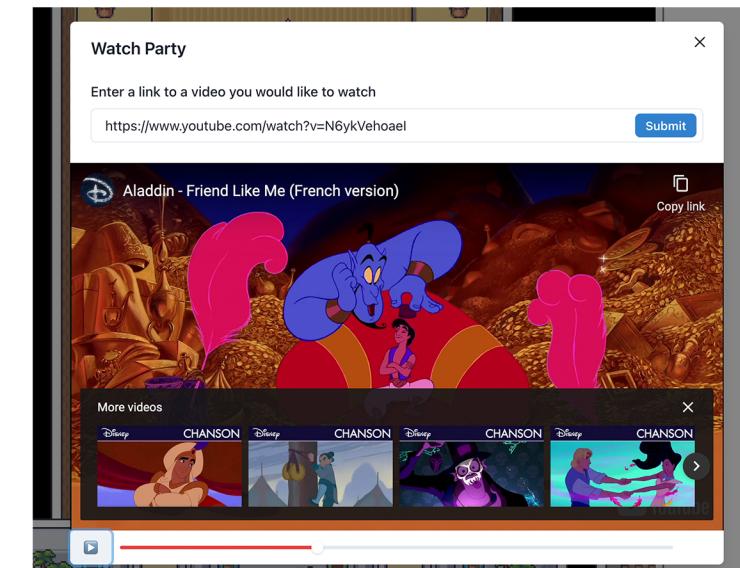


The modal open and playing a video. The video and timestamp are the same for everyone

Future Work

When brainstorming different ways to implement a viewing area, we initially envisioned support for any kind of free, publicly accessible video url. To ensure completion of an MVP by the project deadline, we focused on solely supporting YouTube video urls. One decision made along the way was choosing a video embed component that supported only YouTube videos. This among a select few other places was where there was YouTube specific hard-coding in our design. We learned late in the process of a more widely used video embed component that supported any video url, and was easier to work with, and we decided to switch out the YouTube video component with this more general component. Future work could include extending this abstraction to other parts of our design that currently are specific to YouTube, and generalizing them to any video url.

In the future we’d also like to include a full search functionality for videos rather than making a user supply a link to get the video to start playing. This can use the YouTube API, or the API of other video websites, to fetch video suggestions based upon a user defined string. This would allow the user to stay in covey.town fully without ever having to have another window open to actually fetch the links to the videos they want.



The modal open with a paused video. It is paused for everyone else in the town too

Try it out: viewing-area.netlify.app

Demo: <https://youtu.be/mGJbldbISh0>