

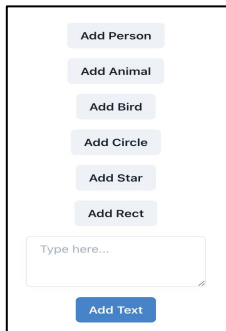
CS4530 Final Project: Stop Motion Studio

Group 603: Leonid Belyaev, Owen Dolan, Ansh Marwa, Cameron Powell

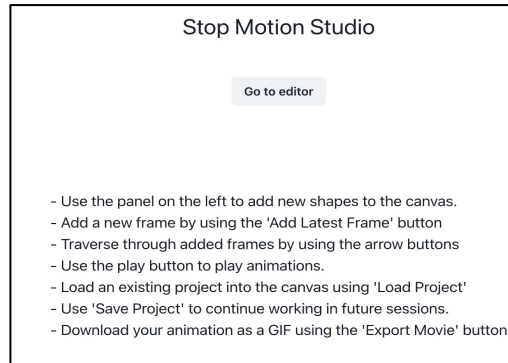
Our Feature: Stop Motion Studio

We thought that it might be interesting to implement some kind of creative software in Covey.Town. This took the form of a stop motion animation studio interactable. Our project allows users to create and edit stop motion animations from scratch using our predefined figures, such as editable text, a human, bird, and an animal, then export and share their animations with friends.

The software includes interactive animation tools, dynamic playback options, and the ability to showcase an animation. Saving is possible both as a .json “project file”, suitable for continued work in a later session, and an animated .gif final export, suitable for sharing out-of-band.



Above is the figure selection panel.



And here is the tutorial for using the animator.

Future Work

Although the current implementation of the software is functional, it depends on out-of-band channels to achieve its social aims, not fully utilizing the power of Covey.Town. In the future, it would be desirable to also implement a backend service to allow in-band sharing of animations. In the same vein, collaborative animation editing, while tricky to implement in a conflict-free way, would significantly enhance usefulness and fun.

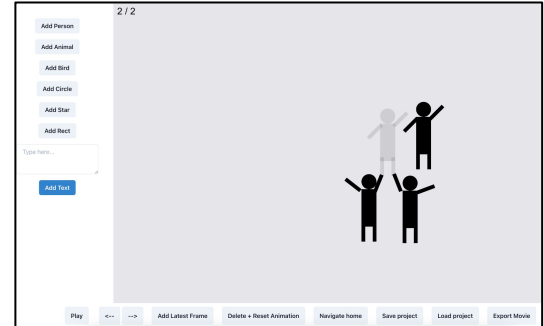
Demo and Source

Demo available at:

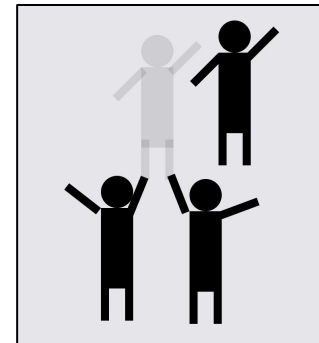
<https://stop-motion-studio-hyvi.onrender.com>

Source available at:

<https://github.com/neu-cs4530/covey-town-project-team-603>



Here is an example animation in progress, with the previous frame displayed in low opacity (closer look below).



Technology Stack and Design

Canvas manipulation is done with React, Chakra UI, and `React Konva` library, a wrapper on top of the `Konva` library to enable React-style declarative programming. In order to implement angular and linear hierarchical figure posing, the default Konva behavior for dragging was largely overwritten with bespoke code. We designed a Frame interface to store a collection of shapes some simple and some complex, and built a list of these frames to serve as the animation sequence.