CS4530 Final Project: "Mini-Map"

Group 2H: Rivindu Wijedoru, Amy Min, Ian McLaughlin, Julia Martinez

Mini-Map Feature:

The smaller "mini-map" is always present in the top-right of the normal town perspective, which allows the player to understand where they're located in the town and where they are relative to other users.

Three details were adjusted to make it easier to read and understand: rendering players as dots with the user blue-green and other players orange-brown, filtering text displayed on the town perspective to de-clutter the mini-map, and changing conversation area colors based on their active state.

Left: Full view of the game screen with the mini-map in the top left corner. Foyer 6 is currently active with Amy and Rivindu discussing final project deliverables.
Right: Enlarged view of the mini-map, displaying players as dots.

Large Map Feature:

The large map was included to provide a bigger view of the town, making additional features, such as teleportation, easier to use. Again, labels and instructions were ignored in this camera view.

The large map can be toggled using the button on the bottom-left or by pressing the "M" key. A user teleports by hovering the cursor over rooms, where a purple button appears for the user to click. The player is then transported into the middle of the chosen room and is brought back into the normal town perspective.

Large map view. The user is currently hovering their cursor over the game room to teleport.

Design and Implementation:

Three major modifications to the Covey.Town codebase were made. The first consisted of adding a React button component to allow for an immediately noticeable way to toggle between perspectives, or to switch between the normal town and large-map views. This modification consisted of making additional React Context/Provider, state, and hook to help provide access to and keep track of toggle values and updating App.tsx.

The second consisted of adding two map classes to represent each of the maps we added. Both map classes extended the same Phaser Camera class, but each map had unique dimensions and zoom levels for the maps. The map cameras were then instantiated in WorldMap.tsx's CoveyGameScene, which allowed us to render the maps and ignore and simplify cluttering information from the maps.

The third consisted of adding teleport buttons that would be displayed on the larger map upon mouse hover, which would teleport the user to a set location in the selected room. The buttons were added to the CoveyGameScene, which allowed us to ignore the buttons for all cameras except that of the large map.

Future Work:

1. To improve our features, we'd *further simplify each of our maps*. The maps would contain only information essential to helping players navigate the town, ignoring further information on the town in the maps. For example, the mini-map would be a solid-fillled shape of the town containing just dynamic components (players, convo. areas) and simplified renderings of town landmarks helpful for distinguishing rooms from each other.

2. To make it easier to modify WorldMap.tsx in future tasks, we would *refactor the CoveyGameScene*, breaking it down into multiple files to shorten the file.

3. Since the maps are intended to help players get a better sense of what's going on in a town, we would *add information labels that would appear upon hover* on the larger map to tell the player, for example, who each player on the map is, and which what people are talking about in active conversation areas.

4. *Abstract the teleport function to any town*. When a player switches to the large-map view, they should be able to select any room in the town and teleport to the clicked location as long as it's an empty tile.

Demo: https://vigorous-rosalind-641d14.netlify.app/
Source: https://github.com/neu-cs4530-s22/team-project-group-2h.git