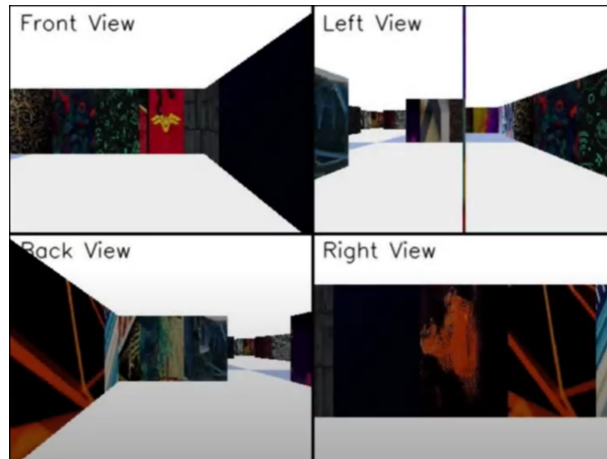
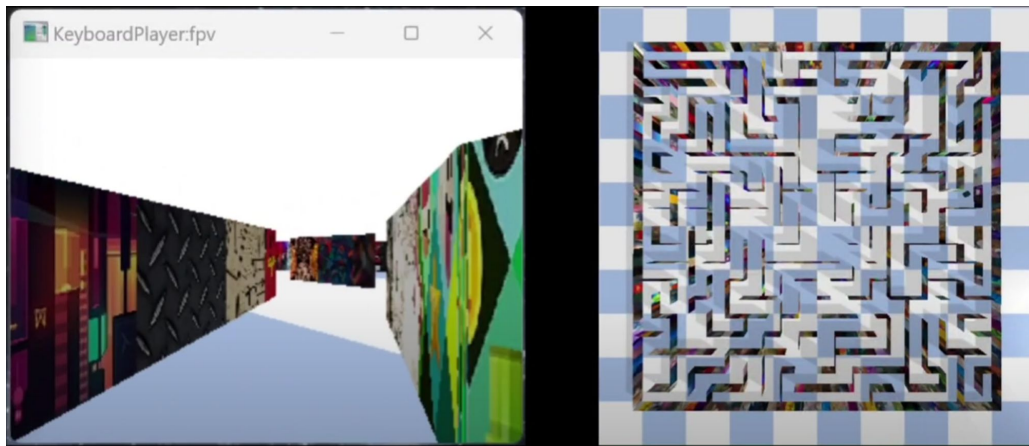


# Problem Statement:

Given access to only a monocular camera, explore a feature rich maze and then navigate to a target view.



# Exploration

- During the exploration phase, the goal is to generate a map of all valid poses and the views associated with those poses. The hope is that the target view and its associated pose will be found during this phase and using the list of valid poses, the pathfinding algorithm can find a path to the target view.
- Dead reckoning via counting the number of frames the player gives directional input is used to keep track of the pose and each frame, an image embedding is stored with the relevant pose information.
- A mini-map is generated by plotting the players history of poses to inform the player what areas of the map are under explored. An orange X gives the player helpful information about their heading relative to the mini-map.

# Navigation: Place Recognition

- Place recognition is calculated by looking at the history of views gathered during exploration and looking at SIFT features.
- The FAISS algorithm is used to train a custom vocabulary which is used to find top similar views to the target images.
- The player manually picks the closest matching image and its associated pose is used for path finding.
- On the off chance that the simulation crashes during navigation, the image embeddings can be loaded from a save directory with the relevant pose information provided in the image title.

# Navigation: Pathfinding

- Given the set of valid poses found during exploration and the selected target image, dijkstra's algorithm is used to find the shortest path to the target.
- A valid path is defined at any two poses which have a difference of only 1 in the x or y direction since these values are calculated discretely via dead reckoning. The cost of every linkage is always 1.
- The generated path is highlighted in blue on the minimap generated in exploration and the player can navigate along that path.

