Goal This semester saw the first implementation of a web-based GUI for Gamescrafters games.

Members Calvin Wong, Robert Meyer – Responsibilities included learning the current framework for GamesmanUni. Coding the front end GUI and its interactions with the backend data. Keeping the GUI graphically consistent with the rest of the website. Handling user input such as mouse clicks.

Overview For Fox and Geese, the goal is to get the Fox to the side opposite to its starting side, and the Geese work to trap the Fox. Both sides can move only in one direction, the Fox moving towards the Geese’s initial side, and the Geese to the Fox’s initial side. Both pieces are on an 8 x 8 grid, with pieces only able to move diagonally. Essentially the movement is checkers without captures.

As for the GUI aspect, first we drew the grid, then interpreted the backend data to find where each piece’s X and Y should be, filling in squares with a Fox, Goose, or doing nothing respectively. After that, the framework had a “hint” and “clickable” system to first lay down a graphical hint as to what moves were available, and color winning moves, then clickables to actually select the move.

Problems & Bugs The main problem was trying to handle user input. The framework was relatively new, and the “hint” and “clickable” system had only been done for relatively simple games like Tic Tac Toe, where you merely click a position and put a piece at that position. We weren’t quite sure how to retrieve the next available moves from the gamedata, which is what we needed to basically draw arrows from the current geese to the possible moves geese, which was the bottleneck.

Final Product and Future Development The board and piece placement is finished. The only thing left to do is figure out how to get the next moves and draw out all possible hints from the current pieces to the positions they could possibly move to, and then figure out how to make the hints clickable and move to the right positions.

This is pseudocode for the new loop logic that should replace the current loop through all cells. This logic makes more sense for this specific game, as the current logic was
largely transferred from TTT and is not as suitable. Drawing p0 and p1 tokens have been taken care of.

draw empty board

for each cell

if there is p0 token

draw p0 token

if the hint string is not empty

draw each hint diagonally

else if there is p1 token

draw p1 token

if the hint string is not empty

draw each hint diagonally

<use draw board />

<g for each ....>

<g v-if token 0>

<use draw token 0 />

<g v-for each diagonal cell>

if the hint string is...

<use v-if condition is satisfy, draw hint />

</g>

</g>

<g v-else-if ...>
... 
</g>
</g>