Technical Documentation

Defense Software and Usage:

The defense portion of the game (the actual game play) code sets up a tower defense game environment using a combination of JavaScript and html templates. It initializes variables for enemies, towers, game states, scores, assets paths, and various game parameters like enemy and tower attributes. There are functions to generate game economy, choose game difficulty, set up the game environment, move enemies, activate towers, run game rounds, place towers, handle game contracts, calculate distances, and manage the game's UI elements.

Production Software and Usage:

The HTML structure sets up an HTML document with embedded CSS styles and a JavaScript section for scripting. It includes a canvas element for graphics and a toolbox with buttons like steel source, copper source, storage, constructor, autoshop, conveyor belt, mouse for interaction, and delete for removing elements. The interface elements define two hidden div elements that display information or instructions during gameplay.

In JavaScript we implemented buttons, triggering actions like building placement, conveyor belt creation, tool selection, and displaying help/information. Building placement allows users to place various types of materials (steel source, copper source, storage, constructor, autoshop) on the grid. Conveyor belts can be created to connect buildings and move items between them.

The code manages building inventories, crafting recipes, and resource transactions (buying, selling, crafting). It includes a game loop using setInterval to update the game state, handle building interactions, movement along conveyor belts, and redraw the canvas. Graphics are rendered using the canvas API for drawing grid lines, buildings, conveyor belts, and moving items. There are dropdown menus for selecting recipes or actions, and dynamically updating information such as inventory and money.

Database Software and Usage:

The database portion of our game was implemented using a flask application. It is made up of html,CSS,JavaScript and python files. The html files inside the templates folder are being used by the routes from access.py .Each route is connected to a different template (html) in order to render the visuals and screens of the game. Within each html file there are CSS attributes in order to stylize some of the elements to be visually appealing to theuse.The JavaScript for the database is used in order to manipulate the data and send it to the specified routes in access.py. The functions in access.py are used to manipulate the data inside the database using sql commands mixed with grabbing the data from the JavaScript.

We are using cookies to move our data between the tower defense and production portions of our game. Cookies allow us to transfer data using local storage on our computers. We plan to transfer data pertaining to a user’s crafted towers and their quantity in a string format that can be accessed by the tower defense portion of the game by pulling that information from the cookies.