Design Document for BunBuff Simulation

Application State and Flow

I created the following useState components to manage the game's hooks:

settings: These are the main settings given by the user including grid size, starting population, max lifespan, and aggression details.

monthCount: Tracks the current month

isRunning: Tracks whether the simulation and grid should be running

log: This is a list holding the events of the simulation.

grid: This holds a list of grid cells, each cell is an object. Each of these object contains an x coordinate, y coordinate, and list of buns in that cell.

buns: Stored within grid object contains, id, x coordinate, y coordinate, age, aggression, and litter size. These attributes of the buns each also have their own state

stats: This is an object that holds the stats of the simulation. These include, the current month, total buns, buns alive at the time, average lifespan, average births, and average deaths.

The isRunning state controls the grid and the control panel, this control panel holds inputs that allows the user to give settings to the game. Once the user gives these settings and hits the start button, the start simulation function is run. This function first checks if the settings are within the parameters given for aggression, and I also added in some checks for grid size to keep the game from getting too big. If these parameters are not met there is an alert set to the user to enter valid inputs. The month count is then set to one and the settings are given to their state object. The stats are then initialized with the settings and is running is set to true.

Then still within the start simulation function the initialize grid function is called. This function first creates a temporary grid thats will become the grid. The maximum buns for the grid size is then calculated and a check to get either the initial population or max buns for the grid. The grid is then created with a loop through the dimensions given, at this point no buns are created. Then a counter for buns placed is created and I start to loop through until the initial population is met. I then choose a random index within the the grid and check if there are already two buns there, if there isnt a bun is created with its attributes. Lastly the buns placed counter is increased and once the population is met the temp grid is returned to be created. The Log then alerts the user that the simulation has started and month 1 has began.