

Pseudo Code for point and click *feeding* game. Geese Feed Game.

Main goal: Targets appear then disappear after time interval or clicked TICK 😊

Secondary goal: Introduce scores and 10 rounds TICK 😊

Tertiary goal: Introduce moving targets TICK 😊

Quaternary goal: mode settings – round based or endless mode

28/01 – Quinary goal: star rating on level select – to do after full page css

The game will be a point and click game with the objective of *feeding* geese.

There will be a nav bar for a title card and resetting the game.

There will be a target (goose) gallery where the targets (geese) appear and will need to be clicked (*fed*) to disappear or will disappear (flee) after a certain amount of time.

Once all geese have been *fed* or fled the game will progress to the next round if more than 50% of the geese have been *fed*.

When less than 50% of the geese have been *fed* Round count, Score total, and Reward is displayed

Main Goal: Psuedo

Html:

- Shooting range box
- Nav-Bar header: (re)Start button, Title

SCSS:

- Shooting range Box size and position
- Targets shape – all look the same

JS:

- Populate HTML with target divs.
- Each targetDiv has a unique ID
- Each targetDiv has unique position details in target box
- Each targetDiv has addEventListener
- The populate HTML must only last certain time for each DIV
- Look towards reset button for rounds counter

### Main JS pseudo code:

Call for nav bar (re)Start button (could have clicking the main box to begin)

Call for main box div

Function that on start button click populates galleryDiv-html with targetDiv elements

Positions will be decided using an array of available positions. Once used loaded out of available positions

Html of targetDiv will need a unique id corresponding to its page position as well as a class of targetDiv

Target div will be made of a button and image for the button. Event listener will be put on button with value of targetDiv id

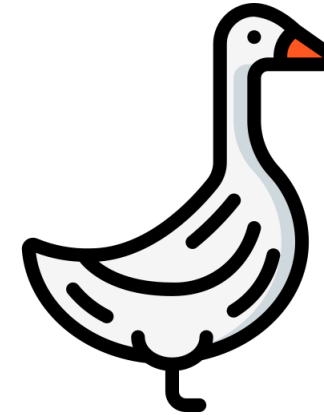
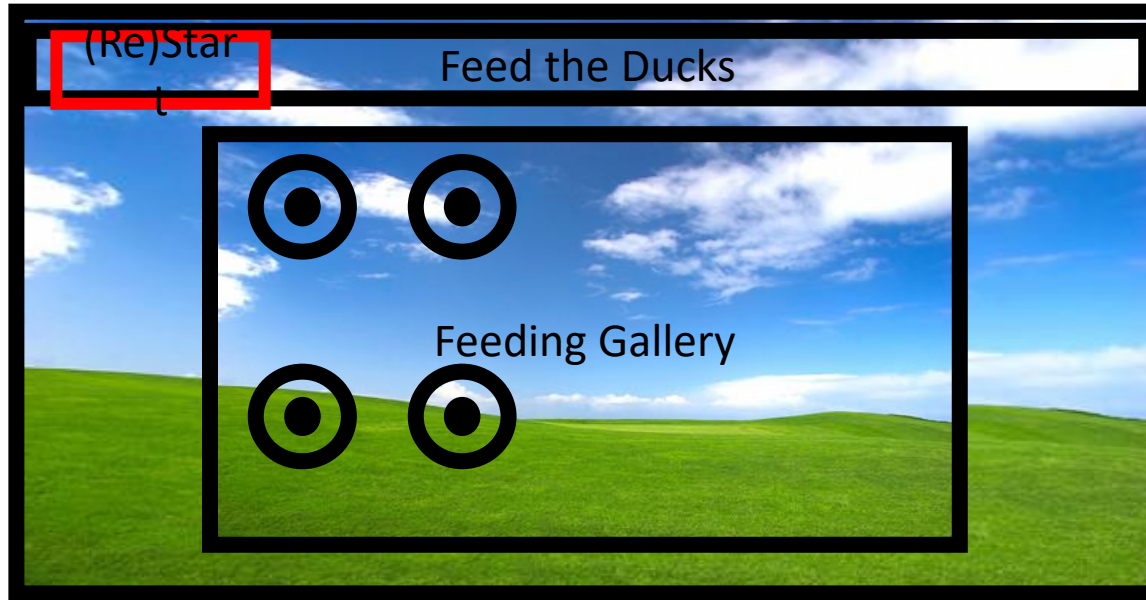
Function that handles div elements being clicked and unloads them from the page.

Loads position it was in back to the available position array.

Event listener for nav bar

Event listener for each targetDiv

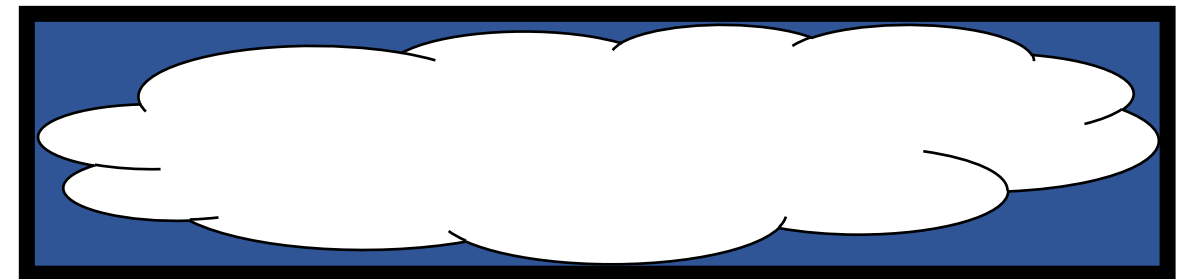
## SCSS: Generic Display layout



Goose Icon

headerDiv with background image as a cloud and information strategically placed on it to house buttons and appear as a div shaped like a cloud

Blue and border removed in realisation



Not implemented in final game.

27/01 - Functional frame of game works.

- To finish main JS goal geese need to appear and disappear after a time interval

`setinterval(function, time)` function will happen every time seconds

Function is defined once and then repeatedly ran.

`clearinterval(intervalID)`

write a function that hatches a goose from the array every x seconds then uses `setTimeout` to clear it from display

`Set timeout(function, time)` function will happen after given time

Had problem with declaring the function as it can only be called as *function* not *function()*. Fixed this issue and code works and rounds can be run with press of a button. Game can be reset with `reload()`

28/01 completion of main JS goal!

CSS needs to be tidied up but aesthetics aren't a priority until tertiary JS goal is complete

Introduction of rounds will be introduced using a while condition or an if condition and round load will be made into a function that runs for each condition.

*For round=0, round<10, round ++*

*When round = 0 consts = consts*

**if (lostGeese < number of geese in round 0 )**

**Run round[0]**

**Else if (lostGeese < number of geese in round 1**

**Can use switch case**

**Can have rounds or level select??? Level select is easier for me to code rn.**

**Level select: level available to click when condition met in a round**

**Have an option of level select or endless mode.**

**Rounds don't occur anymore, simply select next level and then**

## Secondary goal: Introduce scores and rounds

This will be implemented via a score counter on the geese

Can use a while, while score < x so when score = x clears screen.

Displays round next. Round score set to zero and total score is kept track.

Round counter changed when no geese are on screen

Stage two Objectives:

Introduce a level select bar on the lefthand side. Next Div will become available after achieving a score of > 50%.

## **Pseudo code for level select:**

Click level (event.target.value = current round number)

If event.target.value <= highest level available{

Run handle round +

setTimeout timed to end of round that checks score >= (totalgeese in round)/2

If true and If event.target.value = highest level available

highest level available = event.target.value

Else do nothing as then the level must be played again

(can change the else statement another time)

} Else {


Alert player of level progression requirements.

}

## Tertiary goal: Introduce moving targets

Using some relation equations between width of box, speed of animation, and translation angle of a goose I will have a moving goose mode available with 10 levels of moving targets originating from an array of positions on the screen. Initially the bottom, with prospect of movement from the sides.

W3 school code using set interval to be utilised and improved for varying vectors.



```
<script>
var id = null;
function myMove() {
  var elem = document.getElementById("myAnimation");
  var pos = 0; //initial position
  clearInterval(id);
  id = setInterval(frame, 10);
  function frame() {
    if (pos == 350) { //width of box
      clearInterval(id);
    } else {
      pos++;
      elem.style.top = pos + 'px'; //moves one pixel down every frame
      elem.style.left = pos + 'px'; // moves one pixel right every frame
    }
  }
}
</script>
```

Left and top are x and y positions. To change angle of trajectory is the relation between them.

If the angle is changed so must final pov value (divided by largest multiplier)



01/02 Tertiary JS goal almost complete.

Known bugs that need fixing:

- Mode select needs to clear intervals
  - can't clear intervals on level select so will lock players to level TICK 😊
- Selecting a round should reset all intervals
  - can't clear intervals on level select so will lock players to level TICK 😊
- Resetting the game should reset all intervals and variables.
- Mode select should save progress on each mode. TICK 😊
  - Note rounds available are currently saved but icons need to match those available TICK 😊

Next Goal after mode select complete.

Taking into consideration timeframe achievable goals are:

Endless mode

Pretty SCSS

Pop up overlay Instructions menu. TICK 😊

Endless mode pseudo

Endless = Set interval(

if(miss 5 geese in a row endless ends and  
highscore is displayed) check is if score < total  
geese-5

Clear interval (endless)

Else {

Setinterval(()=>{increase round},100)

Create a goose html, += html into div, ensure it  
has its own id.

Decide to random place it or launch it

When random place it attach lifetime

When random launch it attach velocity  
,10)

Each one will have its own attributes

it starts a new “round” on each  
interval

Attach round number to goose  
properties

Final progress report.

The game works with two functional modes. Flying and Standard. Was not able to get endless mode working due to time frame but its left over code is saved separately and can be implemented given extra time. Game is punishingly hard on higher levels as the geese get faster but I believe that's part of the fun as they're only hard as the geese are fast so if a player is fast enough they can beat it. Page is styled well with clear labels on all functions and an alert that locks players to a level but also pauses the round when they try.