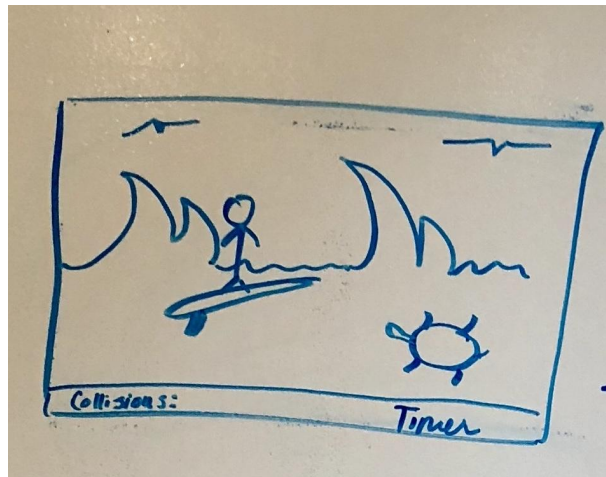


Surf's Up

A game by Stephanie Burr



- arrows $\uparrow \downarrow$ control position
 - dodge road blocks
 - Start & end pt.
 - "rate" surfer based on time it took to get from A to B.
 - Time penalty for collisions
- collision = 2 divs touching
- store time
- Road blocks
- turtle
 - shark
 - other surfer



Surf's Up

-Player moves the surfer up/down the screen with keyboard arrows and tries to avoid hitting turtles.

-2D-collision with `setInterval` to constantly check location, look for collision and push to an empty array if there is collision.

```

//*****COLLISION DETECTION*****
window.setInterval(detectCollision, 330);

function detectCollision() {
  let surferWidth, surferHeight, surferX, surferY,
      turtleWidth, turtleHeight, turtleX, turtleY;
  let turtle = $('.background-image div:last-child');

  surferWidth = surfer.outerWidth(); //Width, Height, X & Y check of surfer's position
  surferHeight = surfer.outerHeight();
  surferX = surfer.offset().left;
  surferY = surfer.offset().top;
  turtleWidth = turtle.outerWidth(); //Width, Height, X & Y check of turtle's position
  turtleHeight = turtle.outerHeight();
  turtleX = turtle.offset().left;
  turtleY = turtle.offset().top;

  if ((surferX + surferWidth) > (turtleX) && surferX < (turtleX + turtleWidth) &&
      (surferY + surferHeight) > (turtleY) && surferY < (turtleY + turtleHeight)) {
    turtle.css('display', 'none');
    timePenaltyArray.push('hit'); //push to an array and access sum of array for Total Count
  }
}

```

Surf's Up

-Used `setInterval()` again to make new turtles.

-Used `Math.random()` to get random Y coordinates for turtle positioning

-Pushed Y value to empty array to assign to newly built turtles in the dom

-Empty array to access again later

```

//*****MAKE AND LAUNCH TURTLES*****
const turtleGenerator = setInterval(makeAndMoveTurtles, 3000); // if less than 3 sec intervals then
                                                                // collision detection stops working

function randomTurtles() {
  let y = Math.floor(Math.random() * (window.innerHeight - 200)); //adds random start location of
                                                                    //within height of window
  turtlePositionArray.push(y);
}

function makeAndMoveTurtles() {
  randomTurtles();
  for (let i = 0; i < turtlePositionArray.length; i++) {
    let newTurtles = $('background-image').append('<div class="turtle"></div>');
    $('background-image div:last-child').css("top", turtlePositionArray[i]);
    $('background-image div:last-child').css("right", -200); //starts turtle off screen
    $('background-image div:last-child').css("display", "block");
    $('background-image div:last-child').css("animation-name", "turtlemove");

    turtlePositionArray = [];
  }
}

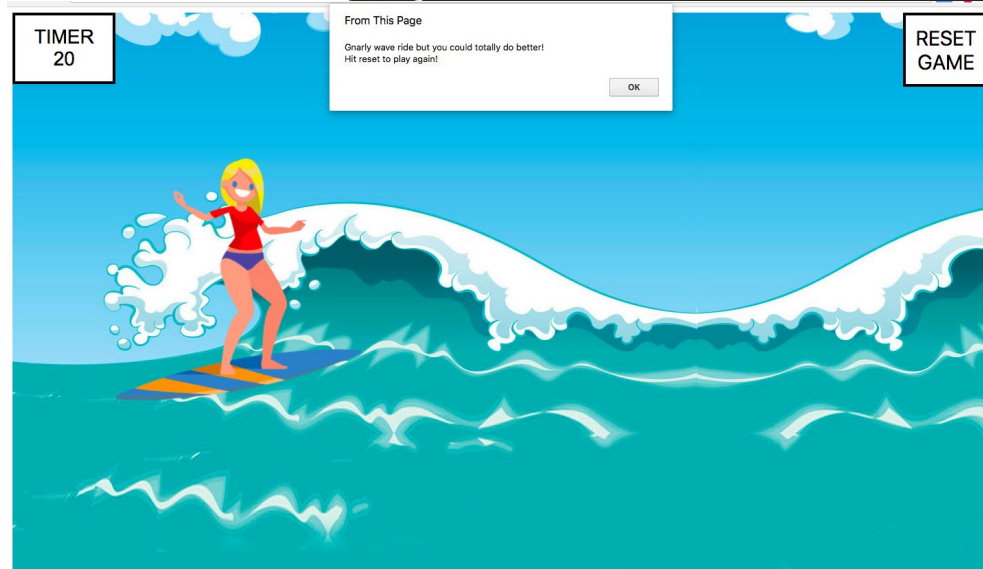
function stopTurtles() {
  clearInterval(turtleGenerator); //function to stop making turtles later
}

```

```

//*****SCALE TO SEE HOW WELL YOU DID*****
function scale() {
  let totalTime = parseInt($('.seconds').html()) + (timePenaltyArray.length*5);
  if (totalTime === 20) {
    alert("Cowabunga Dude! You dodged all the turtles!\nHit reset to play again!");
  } else if (totalTime > 20 && totalTime < 30) {
    alert("Gnarly wave ride but you could totally do better!\nHit reset to play again!")
  } else {
    alert("Sorry Bro! You may need to go back to surf school!\nHit reset to play again!")
  }
}

```



-Player gets a time penalty for hitting turtles

-Player gets graded on a scale with option to play again

Fun things I learned...

- 2D collision detection for two axis aligned bounding boxes
- Used CSS animation to make a “marquee” scrolling background to give the appearance of movement.
- Used CSS animation to launch turtles

Roadblocks

- No more than 1 turtle on the screen at a time or collision detection stops working because I used “last child” to apply css to each new turtle div instead of `getBoundingClientRect`.
- storing data from form
- would have liked to give option of male surfer avatar and gnarley surf music