

1. Declaración de Variables

javascript

```
let state = {};  
let isDragging = false;  
let dragStartX = undefined;  
let dragStartY = undefined;  
  
let previousAnimationTimestamp = undefined;  
let animationFrameRequestID = undefined;  
let delayTimeoutID = undefined;  
  
let simulationMode = false;  
let simulationImpact = {};  
  
const darkModeMediaQuery = window.matchMedia("(prefers-color-scheme:  
dark)");  
  
const settings = {  
  numberOfPlayers: 1,  
  mode: darkModeMediaQuery.matches ? "dark" : "light",  
};  
  
const blastHoleRadius = 18;
```

- **state**: Un objeto que guarda el estado actual del juego.
- **isDragging**: Un booleano que indica si se está arrastrando el ratón para ajustar la velocidad de la bomba.
- **dragStartX y dragStartY**: Coordenadas donde comenzó el arrastre del ratón.
- **previousAnimationTimestamp**: Marca temporal del último frame de animación.
- **animationFrameRequestID y delayTimeoutID**: IDs para controlar la animación y el tiempo de espera.
- **simulationMode**: Un booleano que indica si se está en modo de simulación.
- **simulationImpact**: Guarda el impacto de la simulación para la IA.
- **darkModeMediaQuery**: Consulta para detectar si el usuario prefiere un tema oscuro.
- **settings**: Configuración del juego que incluye el número de jugadores y el modo de color.
- **blastHoleRadius**: Radio del agujero de explosión.

2. Configuración del Lienzo y Elementos del DOM

javascript

```
const canvas = document.getElementById("game");

canvas.width = window.innerWidth * window.devicePixelRatio;

canvas.height = window.innerHeight * window.devicePixelRatio;

canvas.style.width = window.innerWidth + "px";

canvas.style.height = window.innerHeight + "px";

const ctx = canvas.getContext("2d");


// Obtención de elementos del DOM para mostrar información y
// controles

const windmillDOM = document.getElementById("windmill");

const windmillHeadDOM = document.getElementById("windmill-head");

const windInfoDOM = document.getElementById("wind-info");

const windSpeedDOM = document.getElementById("wind-speed");

const info1DOM = document.getElementById("info-left");

const name1DOM = document.querySelector("#info-left .name");

const angle1DOM = document.querySelector("#info-left .angle");

const velocity1DOM = document.querySelector("#info-left .velocity");

const info2DOM = document.getElementById("info-right");

const name2DOM = document.querySelector("#info-right .name");

const angle2DOM = document.querySelector("#info-right .angle");

const velocity2DOM = document.querySelector("#info-right
.velocity");

const instructionsDOM = document.getElementById("instructions");

const gameModeDOM = document.getElementById("game-mode");
```

```
const bombGrabAreaDOM = document.getElementById("bomb-grab-area");

const congratulationsDOM =
document.getElementById("congratulations");

const winnerDOM = document.getElementById("winner");

const settingsDOM = document.getElementById("settings");

const singlePlayerButtonDOM =
document.querySelectorAll(".single-player");

const twoPlayersButtonDOM =
document.querySelectorAll(".two-players");

const autoPlayButtonDOM = document.querySelectorAll(".auto-play");

const colorModeButtonDOM = document.getElementById("color-mode");
```

- **canvas**: El elemento de lienzo para dibujar el juego.
- **ctx**: El contexto de dibujo 2D del lienzo.
- **Elementos del DOM**: Para mostrar información del juego y controles.

3. Configuración del Modo de Color

javascript

```
colorModeButtonDOM.addEventListener("click", () => {

    if (settings.mode === "dark") {

        settings.mode = "light";

        colorModeButtonDOM.innerText = "Dark Mode";

    } else {

        settings.mode = "dark";

        colorModeButtonDOM.innerText = "Light Mode";

    }

    draw();

});
```

```
darkModeMediaQuery.addEventListener("change", (e) => {  
  
    settings.mode = e.matches ? "dark" : "light";  
  
    if (settings.mode === "dark") {  
  
        colorModeButtonDOM.innerText = "Light Mode";  
  
    } else {  
  
        colorModeButtonDOM.innerText = "Dark Mode";  
  
    }  
  
    draw();  
  
});
```

- **Botón de cambio de modo de color:** Permite al usuario alternar entre el modo claro y oscuro. Llama a la función `draw` para redibujar el juego con el nuevo modo.

4. Inicialización del Juego

javascript

```
newGame();
```

```
function newGame() {  
  
    state = {  
  
        phase: "aiming",  
  
        currentPlayer: 1,  
  
        round: 1,  
  
        windSpeed: generateWindSpeed(),  
  
        bomb: {  
  
            x: undefined,
```

```
    y: undefined,  
    rotation: 0,  
    velocity: { x: 0, y: 0 },  
    highlight: true,  
  },  
  backgroundBuildings: [],  
  buildings: [],  
  blastHoles: [],  
  stars: [],  
  scale: 1,  
  shift: 0,  
};
```

```
for (let i = 0; i < (window.innerWidth * window.innerHeight) /  
12000; i++) {
```

```
    const x = Math.floor(Math.random() * window.innerWidth);  
    const y = Math.floor(Math.random() * window.innerHeight);  
    state.stars.push({ x, y });  
}
```

```
for (let i = 0; i < 17; i++) {  
    generateBackgroundBuilding(i);  
}
```

```
for (let i = 0; i < 8; i++) {
```

```
        generateBuilding(i);
    }

    calculateScaleAndShift();

    initializeBombPosition();

    initializeWindmillPosition();

    setWindMillRotation();

    cancelAnimationFrame(animationFrameRequestID);

    clearTimeout(delayTimeoutID);

    if (settings.numberOfPlayers > 0) {
        showInstructions();
    } else {
        hideInstructions();
    }

    hideCongratulations();

    angle1DOM.innerText = 0;

    velocity1DOM.innerText = 0;

    angle2DOM.innerText = 0;

    velocity2DOM.innerText = 0;

    simulationMode = false;

    simulationImpact = {};
```

```

draw();

if (settings.numberOfPlayers === 0) {
    computerThrow();
}
}

```

- **newGame**: Función que reinicia el juego y establece el estado inicial. Genera estrellas, edificios, agujeros de explosión, y configura la posición de la bomba y el molino de viento. También muestra u oculta instrucciones y configuraciones.

5. Funciones de Utilidad para el Juego

Estas funciones generan edificios de fondo y edificios normales, calculan el escalado y el desplazamiento, y configuran la posición de la bomba y el molino de viento. También manejan la visualización de instrucciones y mensajes de felicitación.

6. Funciones de Dibujo

javascript

```

function draw() {

    ctx.save();

    ctx.scale(window.devicePixelRatio, window.devicePixelRatio);

    drawBackgroundSky();

    ctx.translate(0, window.innerHeight);

    ctx.scale(1, -1);

    ctx.translate(state.shift, 0);

```

```
ctx.scale(state.scale, state.scale);

drawBackgroundMoon();

drawBackgroundBuildings();

drawBuildingsWithBlastHoles();

drawGorilla(1);

drawGorilla(2);

drawBomb();

ctx.restore();
}
```

- **draw**: Función principal para dibujar el juego. Dibuja el fondo, los edificios, los gorilas y la bomba en el lienzo.

7. Funciones de Animación y Movimiento

javascript

```
function throwBomb() {

  if (simulationMode) {

    previousAnimationTimestamp = 0;

    animate(16);

  } else {

    state.phase = "in flight";

    previousAnimationTimestamp = undefined;

    animationFrameRequestID = requestAnimationFrame(animate);

  }
}
```



```
}
```

```
function animate(timestamp) {  
  if (previousAnimationTimestamp === undefined) {  
    previousAnimationTimestamp = timestamp;  
    animationFrameRequestID = requestAnimationFrame(animate);  
    return;  
  }  
  
  const elapsedTime = timestamp - previousAnimationTimestamp;  
  
  const hitDetectionPrecision = 10;  
  for (let i = 0; i < hitDetectionPrecision; i++) {  
    moveBomb(elapsedTime / hitDetectionPrecision);  
  
    const miss = checkFrameHit() || checkBuildingHit();  
    const hit = checkGorillaHit();  
  
    if (simulationMode && (hit || miss)) {  
      simulationImpact = { x: state.bomb.x, y: state.bomb.y };  
      return;  
    }  
  
    if (miss) {
```

```
state.currentPlayer = state.currentPlayer === 1 ? 2 : 1;

if (state.currentPlayer === 1) state.round++;

state.phase = "aiming";

initializeBombPosition();


draw();


const computerThrowsNext =

    settings.numberOfPlayers === 0 ||

    (settings.numberOfPlayers === 1 && state.currentPlayer ===
2);


if (computerThrowsNext) setTimeout(computerThrow, 2000);

return;

}


if (hit) {

    state.phase = "celebrating";

    draw();

    return;

}

}


previousAnimationTimestamp = timestamp;

animationFrameRequestID = requestAnimationFrame(animate);
```

}

- **throwBomb**: Lanza la bomba y maneja la animación de su trayectoria.
- **animate**: Actualiza la posición de la bomba y verifica las colisiones con precisión. Cambia el estado del juego en función de si la bomba golpea o no.

HTML

Cabecera (**<head>**):

- **meta charset="utf-8"**: Define la codificación de caracteres del documento como UTF-8.
- **<title>Juego</title>**: El título de la página que aparece en la pestaña del navegador.
- **<link rel="stylesheet" href="index.css" />**: Enlaza con un archivo CSS externo para estilos.
- **<script src="index.js" defer></script>**: Enlaza con un archivo JavaScript externo. El atributo **defer** indica que el script se ejecutará después de que el documento haya sido completamente cargado.

Cuerpo (**<body>**):

- **<canvas id="game"></canvas>**: Un área de dibujo que probablemente se usará para mostrar el juego.
- **SVG de un molino de viento**:
 - El elemento **<svg>** define un gráfico vectorial escalable. En este caso, está creando un molino de viento con un **<path>** que define las palas del molino y un **<circle>** para el centro.
 - **defs** contiene definiciones reutilizables para las partes del molino.
 - **<g transform="translate(100, 100)">** agrupa elementos y los traslada a una posición específica.
 - **<path>** al final define la base del molino.
- **Información del viento**:
 - **<div id="wind-info">**: Muestra la velocidad del viento, que se actualizará dinámicamente.
- **Información del jugador y del computador**:
 - **<div id="info-left">** y **<div id="info-right">**: Muestran información sobre el ángulo y la velocidad tanto del jugador como del computador.
- **Instrucciones del juego**:
 - **<div id="instructions">**: Proporciona instrucciones sobre cómo jugar.
- **Área para agarrar la bomba**:

- `<div id="bomb-grab-area"></div>`: Un área donde se puede interactuar con una bomba, probablemente para arrastrarla y lanzarla.
- **Pantalla de felicitaciones:**
 - `<div id="congratulations">`: Muestra quién ganó y ofrece opciones para comenzar un nuevo juego con diferentes modos.
- **Configuraciones:**
 - `<div id="settings">` incluye:
 - Un botón de menú desplegable para comenzar un nuevo juego en diferentes modos.
 - Un botón para alternar entre modo claro y oscuro.
 - Un botón para cambiar entre pantalla completa y modo ventana, con un SVG que representa los íconos de entrar y salir de pantalla completa.

CSS

Importación de Fuentes

CSS

Copiar código

```
@import
url("https://fonts.googleapis.com/css2?family=Inconsolata:wght@400;700&display=swap");
```

Importa la fuente **Inconsolata** desde Google Fonts con dos pesos de fuente (400 y 700) para usar en el resto del CSS.

Estilo del Cuerpo

CSS

Copiar código

```
body {

    margin: 0;

    padding: 0;

    font-family: "Inconsolata", monospace;

    font-size: 14px;

    color: white;
```

```
user-select: none;

-webkit-user-select: none;

display: flex;

justify-content: center;

align-items: center;

height: 100%;

overflow: hidden;

}
```

- El **body** tiene márgenes y relleno cero.
- Se utiliza la fuente **Inconsolata** con un tamaño de 14px.
- El color del texto es blanco.
- La selección de texto está desactivada.
- La visualización es flexible, centrando el contenido tanto vertical como horizontalmente.
- Oculta el desbordamiento.

Estilo de Botones

css

Copiar código

```
button {

  cursor: pointer;

  border: none;

  color: white;

  background: transparent;

  font-family: "Inconsolata", monospace;

  padding: 10px;

  font-size: 1em;
```

```
}
```

```
button:hover {  
    background-color: rgba(255, 255, 255, 0.1);  
}
```

- Los botones tienen un cursor de puntero, sin borde y fondo transparente.
- La fuente es `Inconsolata`, con un tamaño de fuente de 1em.
- Al pasar el cursor sobre el botón, el fondo cambia a un color blanco semitransparente.

Estilos para Información y Contenidos

css

Copiar código

```
#info-left, #info-right {  
    position: absolute;  
    top: 20px;  
}
```

```
#info-left {  
    left: 25px;  
}
```

```
#info-right {  
    right: 25px;  
    text-align: right;  
}
```

- `#info-left` y `#info-right` están posicionados absolutamente a 20px del borde superior.
- `#info-left` está alineado a la izquierda a 25px.
- `#info-right` está alineado a la derecha a 25px con texto alineado a la derecha.

Área de Agarre

css

Copiar código

```
#bomb-grab-area {  
  
    position: absolute;  
  
    width: 30px;  
  
    height: 30px;  
  
    border-radius: 50%;  
  
    background-color: transparent;  
  
    cursor: grab;  
  
}
```

- El área de agarre (`#bomb-grab-area`) es un círculo de 30x30px con fondo transparente y un cursor de agarre.

Instrucciones y Felicitaciones

css

Copiar código

```
#instructions, #congratulations {  
  
    position: absolute;  
  
    transition: visibility 0s, opacity 0.5s linear;  
  
}
```

```
@media (min-height: 535px) {  
    #instructions {  
        min-height: 200px;  
    }  
}
```

```
#congratulations {  
    background-color: rgba(255, 255, 255, 0.9);  
    color: black;  
    padding: 50px 80px;  
    opacity: 0;  
    visibility: hidden;  
    max-width: 300px;  
    backdrop-filter: blur(5px);  
}
```

```
#congratulations p a {  
    color: inherit;  
}
```

```
#congratulations button {  
    border: 1px solid rgba(0, 0, 0, 0.9);  
    color: inherit;
```



```
}
```

- Ambos elementos están posicionados absolutamente y tienen una transición de visibilidad y opacidad.
- `#instructions` tiene una altura mínima de 200px en pantallas con una altura mínima de 535px.
- `#congratulations` tiene un fondo blanco semitransparente, color de texto negro, y un filtro de desenfoque de fondo. Está inicialmente oculto con opacidad 0 y visibilidad oculta.

Configuración y Otros Elementos

CSS

Copiar código

```
#settings {  
  
    position: absolute;  
  
    top: calc(20px + 16.385px - 10px);  
  
    display: flex;  
  
    align-items: center;  
  
    gap: 10px;  
  
    right: 11em;  
  
}
```

```
#settings, #info-left, #info-right {  
  
    opacity: 0;  
  
    transition: opacity 3s;  
  
}
```

```
@media (max-width: 450px) {
```

```
#settings, #info-left, #info-right {  
    opacity: 0;  
}  
  
#instructions {  
    visibility: hidden;  
}  
}
```

- `#settings` está posicionado absolutamente con un ajuste en la parte superior y derecha, usando flexbox para alinear sus elementos.
- `#settings`, `#info-left`, y `#info-right` tienen una transición de opacidad de 3 segundos.
- En pantallas menores a 450px, `#settings`, `#info-left`, y `#info-right` tienen opacidad 0, y `#instructions` se oculta.

Menú Desplegable

css

Copiar código

```
.dropdown {  
    position: relative;  
    display: inline-block;  
}  
  
.dropbtn:after {  
    content: "▼";  
    margin-left: 7px;  
    font-size: 0.8em;  
    vertical-align: text-top;
```

```
}
```

```
.dropdown-content {  
  display: none;  
  position: absolute;  
  background-color: #f9f9f9;  
  min-width: 120px;  
  box-shadow: 0px 8px 16px 0px rgba(0, 0, 0, 0.2);  
}
```

```
.dropdown-content a {  
  color: black;  
  padding: 12px 16px;  
  text-decoration: none;  
  display: block;  
  white-space: nowrap;  
  font-size: 0.9em;  
}
```

```
.dropdown-content a:hover {  
  background-color: #f1f1f1;  
}
```

```
.dropdown:hover .dropdown-content {
```

```
display: block;

}
```

- `.dropdown` es un contenedor para el menú desplegable.
- `dropbtn:after` añade un triángulo hacia abajo al botón del menú desplegable.
- `.dropdown-content` está oculto por defecto y se muestra cuando se pasa el cursor sobre el contenedor `.dropdown`.
- Los enlaces dentro del menú desplegable tienen un color de texto negro, relleno, y un fondo que cambia al pasar el cursor.

Molino de Viento

CSS

Copiar código

```
#windmill {

    position: absolute;

    right: 0;

    fill: rgba(255, 255, 255, 0.5);

    transform-origin: bottom;

}

#windmill-head {

    animation-name: rotate;

    animation-duration: 4s;

    animation-iteration-count: infinite;

    animation-timing-function: linear;

}

@keyframes rotate {
```

```
from {  
    transform: rotate(0deg);  
}  
to {  
    transform: rotate(360deg);  
}  
}
```

```
#wind-info {  
    position: absolute;  
    width: 100px;  
    text-align: center;  
    margin-bottom: 30px;  
}
```

- `#windmill` está posicionado absolutamente en la parte derecha y tiene un relleno semitransparente.
- `#windmill-head` tiene una animación de rotación continua de 4 segundos.
- La animación de rotación (`rotate`) gira el molino de viento 360 grados.
- `#wind-info` está posicionado absolutamente con un ancho de 100px y texto centrado.

Pantalla Completa

CSS

Copiar código

```
#fullscreen {  
    all: unset;  
    cursor: pointer;
```

```
position: absolute;

right: 10px;

bottom: 10px;

}
```

- `#fullscreen` es un botón para pantalla completa sin estilos predeterminados y está posicionado en la esquina inferior derecha.