

**INTEGRANTES**

**BRANDON FABIÁN MORALES MELÉNDREZ 0907-23-10341**

**ILIANA YURIDIA ORELLANA MARROQUIN 09-07-23-2396**

**ANGELICA MARÍA MEJÍA TZOC 0907-23-8872**

**17 DE JUNIO DE 2023**

**UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA**

**FACULTAD DE INGENIERÍA EN SISTEMAS**

**PRIMER SEMESTRE**

**INGENIERO: GERSON ALTAMIRANO**

**PROYECTO FINAL DE LÓGICA**

**Planteamiento del problema**

Crear un juego creativo que contenga javascrip, css y html.

**Análisis básico**

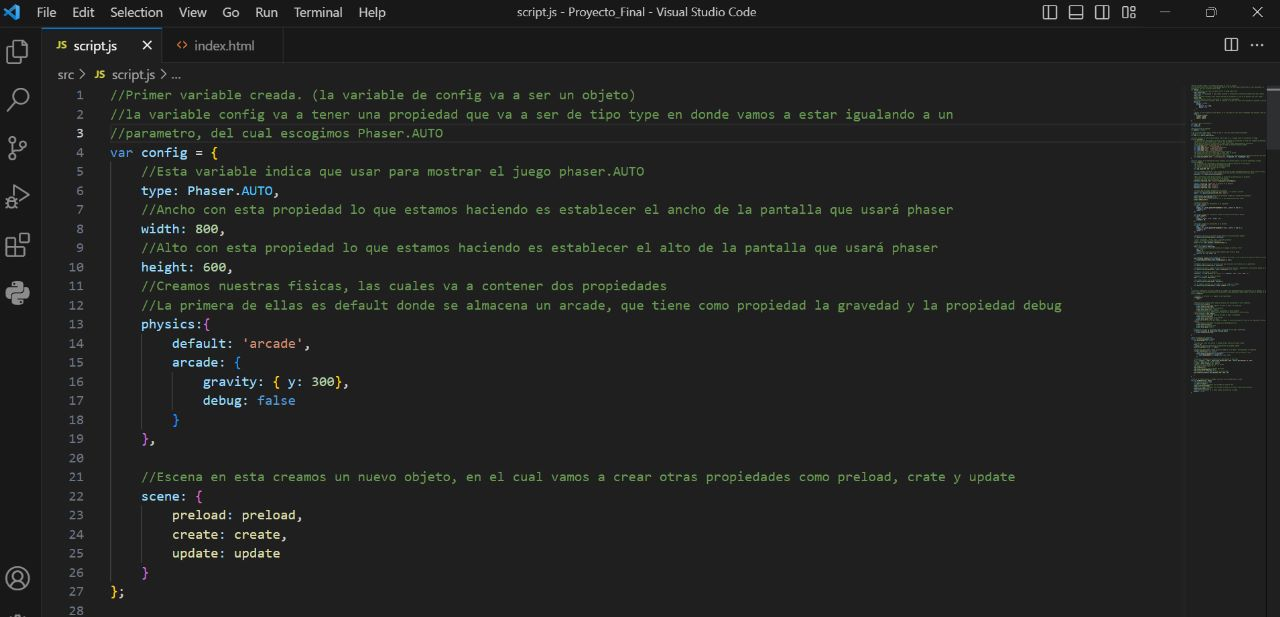
Una entrada básica con un botón star para iniciar.

Un marcador que indica el puntaje del jugador.

Y movilidad con las teclas de desplazamiento.

**Diseño de la solución**

El código es un juego creado con Phaser, un framework de JavaScript para el desarrollo de juegos en 2D. A continuación, se describirá la estructura y las funciones principales del código:



* Se define un objeto `config` que contiene las propiedades del juego, como el tipo de renderizado, el ancho y alto de la pantalla, y la configuración de físicas.
* Se establece un objeto `scene` que contiene tres funciones principales: `preload`, `create` y `update`.

Texto

Descripción generada automáticamente

* + Se crea una variable `score` inicializada en 0 para almacenar la puntuación del jugador.
  + Se define una variable `scoreText` para mostrar la puntuación en la pantalla.
  + Se establece la variable `gameOver` como falsa para controlar el estado del juego.
* . Inicialización del juego:
  + Se crea una instancia del objeto `Phaser.Game` utilizando la configuración definida en `config`.
  + El juego se iniciará automáticamente.

Texto

Descripción generada automáticamente

* . Función `preload`: Esta función se encarga de precargar los recursos necesarios para el juego, como imágenes y hojas de sprites.
* Se cargan las imágenes del fondo (`sky`), la plataforma (`ground`), la estrella (`star`), la bomba (`bomb`) y el sprite del personaje (`dude`).

Texto

Descripción generada automáticamente

* **. Función `create`:**
  + Esta función se ejecuta después de que se hayan cargado los recursos y se encarga de crear los elementos del juego.
  + Se agrega una imagen de fondo utilizando la imagen cargada previamente (`sky`).
  + Se crean las plataformas del juego utilizando el grupo `platforms`.
  + Se crea un sprite de jugador (`player`) con el sprite del personaje cargado previamente (`dude`).

Texto

Descripción generada automáticamente

* Se definen las animaciones para el jugador, como caminar a la izquierda, caminar a la derecha y estar quieto.
  + Se establece la colisión entre el jugador y las plataformas.

Texto

Descripción generada automáticamente

* + Se crean las estrellas utilizando el grupo `stars` y se establece la colisión entre las estrellas y las plataformas.
  + Se define una función de superposición (`overlap`) entre el jugador y las estrellas, que se ejecuta cuando el jugador recolecta una estrella.
  + Se crea un texto para mostrar la puntuación del jugador (`scoreText`).

Texto

Descripción generada automáticamente

* + Se crean las bombas utilizando el grupo `bombs` y se establece la colisión entre las bombas y las plataformas.
  + Texto

    Descripción generada automáticamenteSe establece una función de colisión entre el jugador y las bombas, que se ejecuta cuando el jugador golpea una bomba.
* . **Función `update`:**
  + Esta función se ejecuta en cada fotograma y se encarga de actualizar el estado del juego.
  + Si el juego ha terminado (`gameOver` es verdadero), la función se detiene.
  + Se comprueba la entrada del teclado para mover al jugador a la izquierda, derecha o saltar.

Texto

Descripción generada automáticamente

* + Se actualiza la animación del jugador según la entrada del teclado.
  + Si el jugador está en contacto con una plataforma y se pulsa la tecla de salto, el jugador se impulsa hacia arriba.
  + Texto

    Descripción generada automáticamenteSi el jugador recolecta todas las estrellas, se reinician y se generan nuevas estrellas y una bomba.
* **. Función `collectStar**
  + Esta función se llama cuando el jugador colisiona con una estrella.
  + La estrella se desactiva y se oculta.

Texto

Descripción generada automáticamente

* + Se incrementa la puntuación del jugador y se actualiza el texto de puntuación.
  + Si no quedan estrellas activas, se reinician y se generan nuevas estrellas y una bomba
* Texto

  Descripción generada automáticamente. **Función `hitBomb`:**
  + Esta función se llama cuando el jugador colisiona con una bomba.
  + La física del juego se pausa.
  + El jugador se tiñe de rojo y se muestra la animación de estar quieto.
  + Se establece `gameOver` como verdadero para finalizar el juego.

Escala de tiempo

Descripción generada automáticamente con confianza baja

**Diagrama de flujo o algoritmo.**

**Manual de usuario.**

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

**2**

**1**

Aplicación, Icono

Descripción generada automáticamente

**3**

**4**

Interfaz de usuario gráfica

Descripción generada automáticamente

**1:**Primer boton para dar incio a la pantalla de nuestro juego llamado FIA.

**2:**Entraran al cuadro de nuestro juego donde caaeran estrellas.

**3:**Jugar utilizando las teclas de comando de el teclado para recolectar estrellas

**4:**Usar la rueda de refrescar para volver a comenzar el jueg.

**Autores**

**Coordinador:** Brandon Fabian Morales Melendrez.

**Grupo.**

* Iliana Yuridia Orellana Marroquín.
* Angelica María Mejía Tzoc.