Practica Intermedia Javascript

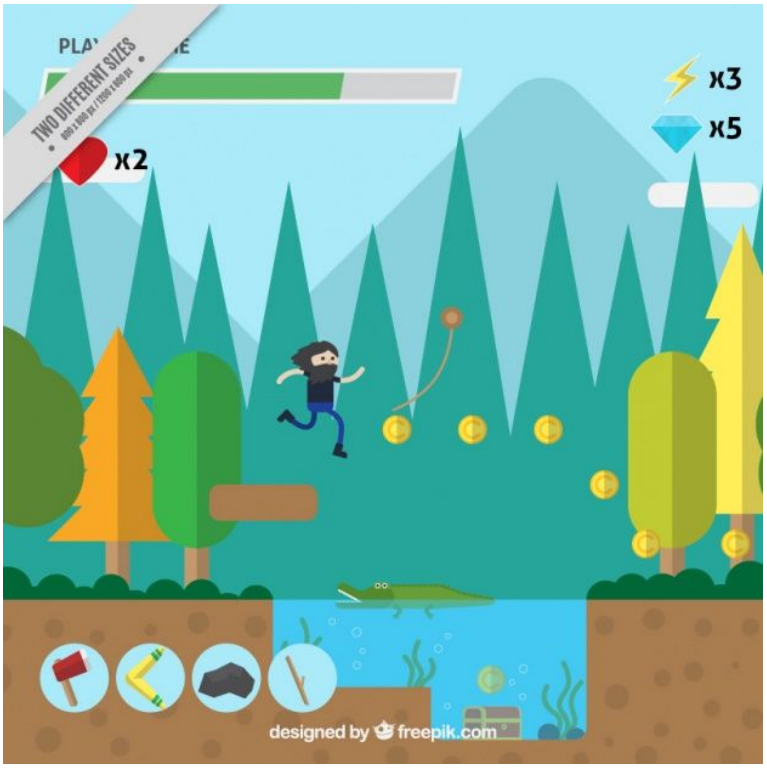
Introducción

Para la siguiente práctica se quiere implementar un juego estilo Pang: el jugador puede moverse hacia los lados, saltar y disparar a los objetos que vayan cayendo del cielo.

Estos objetos que caen serían las diferentes páginas de mi portfolio personal, por lo que si el jugador dispara a estas cajas, el juego mostrará la información correspondiente. Para esta versión dejaremos un texto de placeholder.

Para esta versión, disparar a una caja pausa la ejecución del juego y muestra un mensaje por pantalla. Para la siguiente versión, el jugador entrará en otras escenas con dicha información y tendrá que alcanzar una puerta de salida mediante el salto.

Debido a que mis habilidades artísticas son las de un niño de ocho años, en un futuro se quieren implementar gráficos basados en el flat design: un diseño minimalista y plano. Así se puede obtener un buen acabado gráfico sin emplear mucho en la búsqueda de imágenes apropiadas para los gráficos. Le dejo una imagen de referencia:



Para esta práctica, se han incluído sprite sheets para diferentes animaciones, aunque para el acabado final serán animaciones más sencillas y planas.

Alejandro Borreguero
Grupo 33

Desarrollo de la práctica

Se ha tomado como proyecto de partida uno de los proyectos base vistos en clase, por lo que la clase coins sería el equivalente a las boxes del futuro juego.

Para la clase del proyectil, se ha implementado una función que mueve el proyectil en la escena hacia arriba y otras para la detección de colisiones con el límite superior y las cajas que caen.

- El jugador puede moverse, saltar y disparar. Usamos la clase input como base para recoger la entrada por teclado. Se ha separado el Update general del juego entre UpdateInputs y otro más general para poder quitar la pausa del juego correctamente.
- El jugador y el proyectil presentan una animación por spritesheets. Utilizamos el framewidth y frameheight para seleccionar la cantidad de imagen en X y en Y que cogemos para cada frame.
- El juego tiene efectos de sonido para el salto y el disparo del personaje.
- En el HUD se muestra el tiempo de juego, los FPS, las cajas/ ventanas del portfolio conseguidas, una barra de vida del personaje y el score.
- El fondo tiene un gradiente 'animado', es decir, que cambia sus valores con el paso del tiempo.
- El score se acumula a lo largo de las diferentes partidas que juguemos, no se reinicia cuando morimos. El juego termina cuando te golpean tres veces las cajas que caen.

Clases y código

- **Barbackground**: clase utilizada para controlar el fondo de color de las barras de vida, maná, etc. Esta clase puede utilizarse para distintas barras cambiando solo el color de la imagen de barra utilizada.
- **code**: contiene el funcionamiento principal de nuestro juego.
    - Init inicializa el juego, cargando imágenes y sonidos.
    - Start inicializa las variables del juego, llamando al Loop.
    - Loop llama contínuamente al Draw general, que pinta toda la escena, Update y UpdateInputs, para recoger solo los inputs de teclado. Lo hemos separado para poder pausar correctamente el juego.
    - Drawbackground pinta un fondo animado hecho con un gradiente de dos colores.

    - UpdateProgress maneja un array de booleanos que sirven para mostrar los ticks o checks de qué parte del portfolio has abierto/ completado.
- **coins**: clase que maneja las cajas/ monedas del juego. Para una futura versón, se buscará implementar un sistema para que las monedas/ cajas aparezcan con cierto retardo. He intentado utilizar setTimout pero no me ha funcionado.
- **greentick**: clase que controla los ticks/ checks de la barra de progreso. Estos se dibujan si se ha completado/ roto la caja correspondiente.
- **Input**: maneja y recoge la información del teclado y del mouse.
- **player**: clase que controla las acciones del player/ personaje en pantalla. Este puede moverse, saltar y disparar.
- **playerBar**: esta clase sirve para manejar las barras de los personajes, para manejar la imagen que se utiliza de contorno sobre el fondo de color que cambiamos de tamaño.
- **progressBar**: esta clase se utiliza para manejar la imagen que corresponde a la barra de progreso, es decir, la barra que muestra cuantas partes del portfolio hemos completado/ abierto. Esta barra se utiliza crear al usuario la necesidad de completar todo el portfolio.
- **projectile**: maneja el proyectil que disparamos para romper las cajas.

Conocimientos aprendidos

- ➔ Manejo de inputs de teclado
- ➔ Creación y manejo de clases para imágenes en distintas capas de organización
- ➔ Manejo de un HUD dinámico
- ➔ Utilización de spritesheets y su adaptación para ser mostradas por pantalla mediante una parámetros de animación que ‘adaptan’ esta imagen
- ➔ Manejo de sonidos
- ➔ Manejo de Box2D
- ➔ Uso de setTimeout