Flappy Box

COMP 4046-086 - Computer Graphics
Cindy M Mendez Aviles

Descripción del Proyecto

- Flappy Box is similar al juego popular Flappy Bird, pero lo jugamos con una caja multicolor
- Es un ‘side-scroller’ donde el usuario tiene que controlar la caja sin tener contacto con los pipes
- De haber un contacto o una colisión, pues el juego se acaba y hay un game over
- Cada vez que la cajita pasa por un pipe, se le da un punto al usuario
- Hay sonidos atreves del programa cuando se hace un punto, cuando la caja sube y cuando ocurre una colisión y hay un game over
- El lugar del ‘safe-zone’ entre los pipes se genera aleatorio
- Se guarda las puntuación más alta
- Para reiniciar el juego, solo se tiene que presionar la tecla ‘R”

Diseño del Sistema

- El juego fue diseñado y desarrollado en Javascript y WebGL usando los recursos enseñados en clase y usando una de las asignaciones como ‘template’. Ningun codigo externo fue usado.
- Principalmente se usó translación, y scaling para lograr completar la aplicación.
- El ‘box’ es la figura que se usó en la clase, escalada.
- El programa tiene una docena de variables de control para asegurarse que todo funciona correctamente.
- Se genera cuatro pipes en el canvas, y un safe-zone aleatorio por cada pipe.
- Los pipes se se trasladan en el eje de X continuamente.
- Con la posición de la caja, y la posición del safe-zone, se verifica si hay colisiones en momentos específicos, y de haberlo el programa termina.
- Al programa terminar se guardan los highscore y le damos la opción a los usuarios de reintentarlo.

Herramientas de Implementación

- Javascript y HTML sin ningun codigo externo
- Variables de control para controlar la funcionalidad del programa
- Lógica de traslación y colisión para determinar cuando el usuario pierde
- Código completamente documentado en Github.

https://github.com/Cindy-Mendez/Flappy-Box

```
/* -------------------------- Variables de Control -------------------------- */
//Maneja la posicion en Y del box. Se va decrementando cada vez que se corre el
//render y se incrementa cuando el usuario entra la tecla de space
var yTranslation = 0.000001;
//Variable que controla si el juego debe empezar o no
var start = 0;
//Variable que nos dice si fue la primera vez que se corrio el render
//para esa forma hacer el render una vez, y despues parar hasta que
//el usuario empieze el jeugo
var firstTime = 1;
//Variable para llevar el score del juego actual
var score = 0;
//Variable que lleva la puntuacion mas alta
var highScore = 5;
//Variable que usamos para las translaciones en x de los pipes, se va
//incrementando cada vez que se hace el render.
var xMovement = 0;
//Variable que usamos para crear espacios entre los pipes.
var xTranslation = 0;
//Arreglo donde se guarda donde esta localizado en Y el espacio por donde
//debe pasar el box entre pipes
var randomNess = [ 0.0, 0.0, 0.0, 0.0 ];
//Variable de control donde sabemos si hubo un gameover para no refrescar
//la pantalla y no aceptar mas users inputs hasta que el juego se re-inicie.
var isGameOver = 0;
```

Problemas con la Implementación

- No hubo problemas muy grande con la implementación.
- Si tuviese que mencionar algo, sería como encontrar la colisión entre el box y los pipes, cuando el box pasa por el safe zone.
- Se pasó un poco de tiempo para resolverlo y en la figura podemos ver como lo implemente a mi manera

https://github.com/Cindy-Mendez/Flappy-Box

```
//Si el movimiento en x esta entre .1 y .3, esto implica que el box esta
//pasando por el safe zone y tenemos que verificar si hay colisiones
if (xMovement > .1 && xMovement < .3)
{
  //Esta es una ecuacion(es) que encontramos para determinar si hay colision en
  //base a la posicion en y del box, y el valor de randomNess. Se verifica
  //si hay colision en el pipe de arriba y el de abajo.
  //Para encontrar esta formula, manualmente capturamos puntos de yTranslation
  //donde habia colision arriba y abajo para varios valores de -1 a 1 con
  //incrementos en .1, y asi encontramos la formula
  if( yTranslation < randomNess[3] * .5 - 0.15 ||
      yTranslation > randomNess[3] * .5 + 0.15 )
  {
    //Hubo colision, y por eso el programa deja de renderizar imagenes ya
    //que seteamos la variable de control isGameOver a 1. Tambien enviamos
    //el audio de gameover y verificamos si hubo un nuevo highScore
    var audio = new Audio('Sounds/sfx_hit.wav');
    audio.play();
    if(score > highScore)
      highScore = score;
    document.getElementById("score").innerHTML =
              "GAME OVER! <br> SCORE: " + score + "<br>HIT 'R' TO TRY AGAIN!"
              + "<br> <br> Current High Score: " + highScore;
    isGameOver = 1;
    start = 0;
  }
}
```

Conclusiones

- Siento que aprendí muchísimas cosas nuevas en esta clase y estoy muy agradecida por ello.
- Muy retante, pero divertido.
- Valio la pena el esfuerzo ya que mi ‘skill set’ creció mucho luego de pasar tanto trabajo.
- Creo que mi proyecto final fue algo bien práctico, y algo que despertó algo en mi.
- Espero con ansias seguir trabajando en cosas así y en otros proyectos similares.