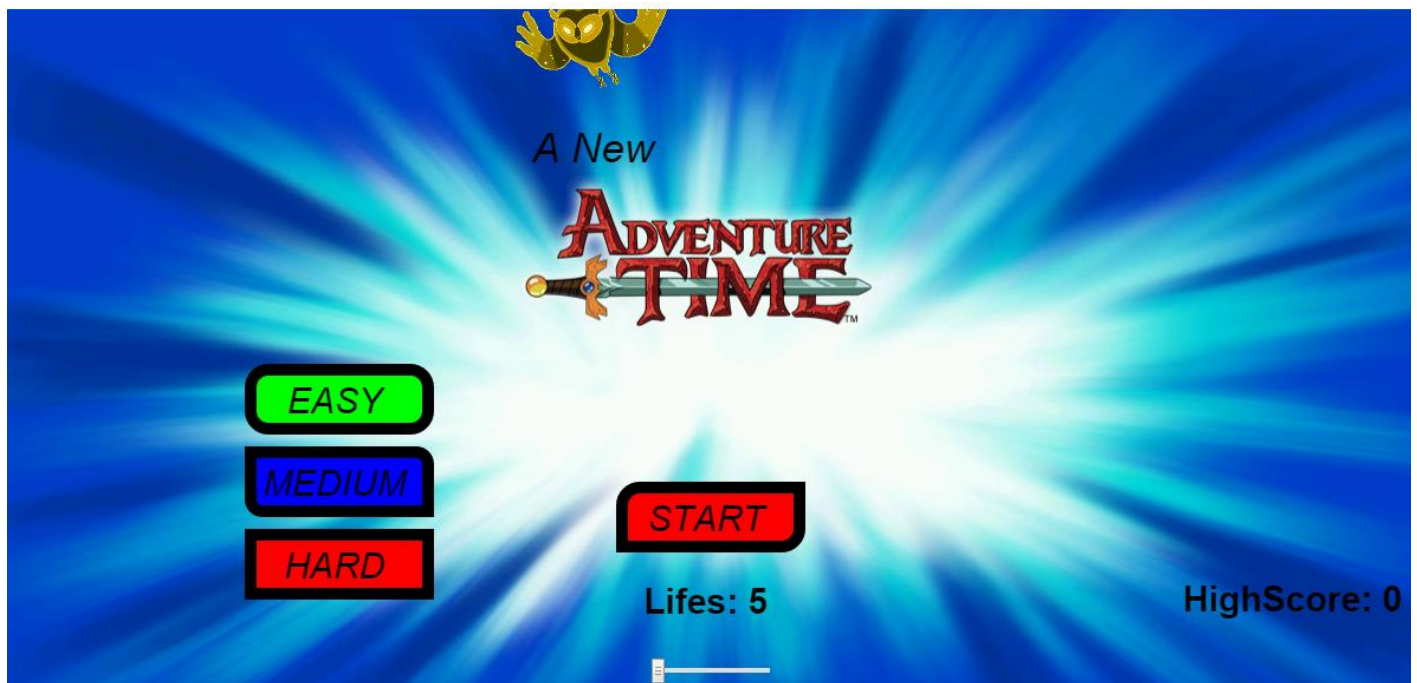


ITESM, Campus Santa Fe
Sistemas de audio digital para web y dispositivos móviles
Eric Pérez Segura
26 de Noviembre de 2015
Vicente Alcantara Santana A01016948
Sebastián Cors Mijares A01019733
María Montserrat Urdapilleta Carrasco A01019557
Alejandro Lajud García A01019323
Proyecto Final

The Jewelscript Wizard: A New Adventure Time



Para el proyecto final de la materia se plantearon 3 principales ideas en cuanto a qué se podría hacer para superar los proyectos anteriores: un sampler hecho con Faust, un secuenciador hecho con Processing o un videojuego en Processing.

Para determinar qué es lo que se debería hacer nos resultó demasiado sencillo pues la idea de Faust creímos que ya estaba muy utilizada y Processing fue

actualizado, de tal manera que no le funcionaba a casi nadie en el curso, por lo que no teníamos en realidad un proyecto ya establecido, pero en una investigación ardua y muy larga, encontramos un lenguaje en JavaScript sencillo, deriva de Processing que funcionaba a la perfección: p5.js es el nombre de este proyecto apoyado por muchos desarrolladores, capaz de soltar la creatividad de cualquiera, está basado en el lenguaje de Processing pero aplicado totalmente a trabajos online.

Después de estudiarlo bastante bien, decidimos realizar el videojuego en p5.js pues era nuestra única opción a seguir.

La idea de hacer un videojuego resultó algo difícil pues no teníamos idea de qué hacer, pero resultó de varias pruebas con diferentes sprites de juegos diferentes que se hiciera un pequeño juego de recolección de joyas con los personajes de una gran serie de televisión para niños y no tan niños: Hora de Aventura, donde el Rey Helado hace de las suyas de nuevo para tratar de vencer a nuestros héroes favoritos. Finn y Jake se enfrentan a la amenaza helada una vez más para recuperar el dinero que les ha robado.

Este proyecto fue construido única y exclusivamente para la educación, no se buscan formas de lucrar con el contenido ni con el programa por completo, se busca presentar el trabajo realizado con mucho esfuerzo y dedicación aprendiendo sobre lenguajes de programación que hasta hace 1 mes descubrimos que les habían actualizado.

Las fuentes de contenido de este trabajo están listadas a continuación para conocimiento del público y para demostrar que no somos dueños del contenido que se ha utilizado, lo único que hemos realizado es el código fuente para poder lograr este pequeño videojuego con el lenguaje de programación p5.js,

desarrollado por una comunidad de programadores que se esmeran en proponer nuevas formas de presentar y complementar proyectos tanto artísticos como generales.

Se supone que el proyecto debe de trabajar en páginas web, siendo que está hecho en JavaScript, solo que actualmente el proyecto es cargado desde un servidor local (Mi laptop) por lo que se pueden presentar diversos tipos de problemas como: carga lenta, error de URL, error de DNS, etc., pero espero que muy pronto pueda averiguar cómo subir el proyecto completo a un servidor online todo el tiempo para poder disfrutar de este gran juego que se los dedico con mucho cariño a todos los que lo jueguen.

Actualizado:

Luego de una ardua búsqueda, encontramos la manera de darle host al proyecto dentro de GitHub, el cual nos permite acceder a los archivos de manera online y ahora sí funciona la página donde se ha subido el proyecto.

Fuentes:

[p5.js - Processing intuition times JavaScript power](#)

[Basic Interactivity with p5.js](#)

[The spriters Resources - 3DS Adventure Time: Ice King...](#)

[YouTube - 10 minutes of Finn And Jake Quotes \[Mostly Finn\] W/ Spanish Subs](#)

[SoundCloud - Adventure Time: Hey Ice King! Why'd You Steal Our Garbage?!! OST](#)

[WampServer](#)

[The Jewelscript Wizard – A New Adventure Time](#)

FUNCIONAMIENTO/CÓDIGO DEL PROYECTO

El código del proyecto está construido dentro de 816 líneas de código, siendo algo muy grande como para explicarlo por completo, pero trataremos de resumirlo lo mejor que se pueda en este documento.

El funcionamiento principal consta de 3 pantallas que interactúan con el usuario, de tal manera que el videojuego tiene un menú, la pantalla del videojuego en sí, y la pantalla del Fin del Juego.

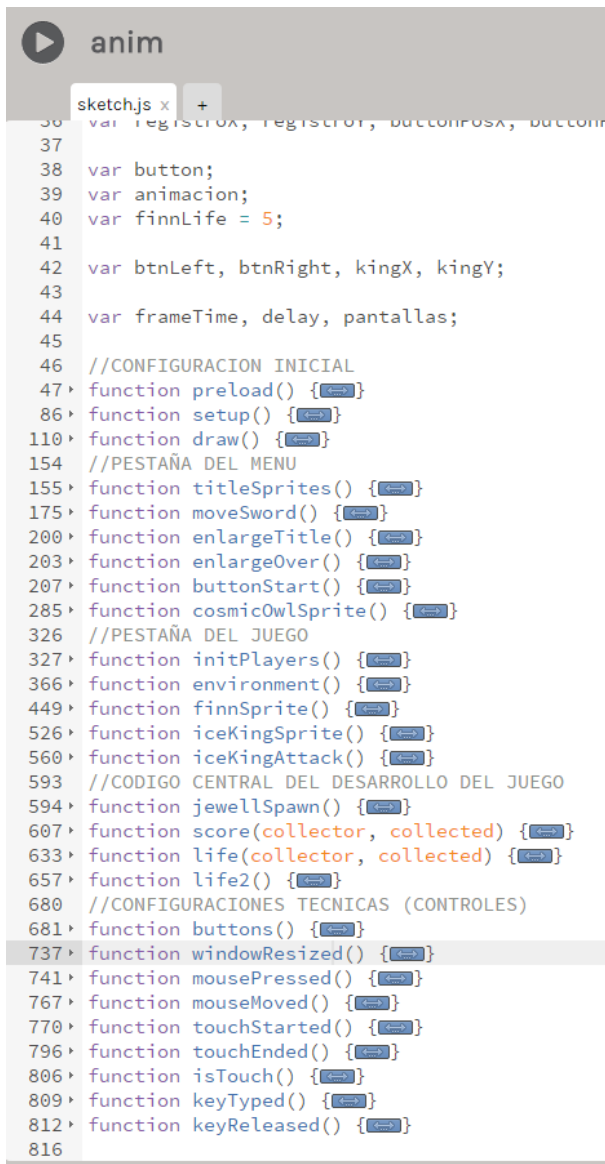


El juego consiste en atrapar tantas joyas como le sea posible al jugador utilizando a Finn Y Jake, antes de que el Rey Helado les quite todas sus vidas.

El juego se controla tanto con la pantalla como con el teclado y el mouse,



donde el jugador puede moverse con estos botones y un circulo para saltar, o con el teclado pueden usar las flechitas para moverse y la tecla de la letra X para saltar.



```
anim
sketch.js x +
30 var registroX, registroY, botonPosX, botonPosY;
37
38 var button;
39 var animacion;
40 var finnLife = 5;
41
42 var btnLeft, btnRight, kingX, kingY;
43
44 var frameTime, delay, pantallas;
45
46 //CONFIGURACION INICIAL
47 function preload() {}
86 function setup() {}
110 function draw() {}
154 //PESTAÑA DEL MENU
155 function titleSprites() {}
175 function moveSword() {}
200 function enlargeTitle() {}
203 function enlargeOver() {}
207 function buttonStart() {}
285 function cosmicOwlSprite() {}
326 //PESTAÑA DEL JUEGO
327 function initPlayers() {}
366 function environment() {}
449 function finnSprite() {}
526 function iceKingSprite() {}
560 function iceKingAttack() {}
593 //CODIGO CENTRAL DEL DESARROLLO DEL JUEGO
594 function jewelSpawn() {}
607 function score(collector, collected) {}
633 function life(collector, collected) {}
657 function life2() {}
680 //CONFIGURACIONES TECNICAS (CONTROLES)
681 function buttons() {}
737 function windowResized() {}
741 function mousePressed() {}
767 function mouseMoved() {}
770 function touchStarted() {}
796 function touchEnded() {}
806 function isTouch() {}
809 function keyTyped() {}
812 function keyReleased() {}
816
```

En cuanto al código, este se fue realizando de acuerdo a las necesidades del momento, esto fue así:

Al principio necesitábamos ver como se creaban y se utilizaban las animaciones en p5.js, utilizando una librería llamada p5.play, y entonces se empezó a estudiar y trabajar con ese código, hasta que resultó que las animaciones ya estaban listas.

Después buscamos la manera de reproducir y utilizar los sonidos de acuerdo a como funcionarían en un videojuego, esto es como incidentales y diálogos, esto fue solucionado con la librería de p5.js llamada p5.sound, la cual es muy parecida a la librería *Maxim* que se utiliza en Processing.

Al final se empezó a realizar el código de lo más técnico como funciones, variables y

demás recursos para estructurar el código y funcionara, pues antes solo teníamos las piezas, ahora estaba el tiempo de armarlas para que formaran un videojuego.

En la función de *preload* se precargaban todas las imágenes que se utilizarán después; en la función *setup* se crea el *canvas* y se estipula qué cosas servirán como ajustes iniciales, y de ahí pasamos a la función *draw* tiene el código más corto pero es fundamental para que se pueda cambiar de pantallas y así el

juego tenga un inicio y un fin; las siguientes funciones ya son el código que permite que se realicen funciones como animaciones de textos, de personajes, de ataques, del ambiente del nivel cuando estás jugando, también el código que permite que te dañen los ataques o que cuando toques al rey helado se te baje la vida, siendo esto lo que permite que termine el juego y así se recaude un score que se presenta en el menú.

ACTUALIZACIONES A FUTURO

Sentimos que para ser uno de los primeros videojuegos que hemos intentado realizar, está bastante bien, pero creemos que necesitará más actualizaciones como algún tipo de actualización del highscore, para que muestre distintos scores de la gente que lo haya jugado y así ir metiendo un poco más de competitividad.

Siento que ya no hay nada más qué explicar, por lo que pido que vayan a jugar el juego y prueben el proyecto final que nos quedó.

Gracias =D.

```
function draw() {  
  if (isTouch()) {  
  } else {  
  
    switch (pantallas) {  
      case 0:  
        background(bg0);  
        moveSword();  
        drawSprites();  
        break;  
      case 1:  
        background(bg0);  
        enlargeTitle();  
        cosmicOwlSprite();  
        buttonStart();  
        drawSprites();  
        break;  
      case 2:  
        background(bg1);  
        finnSprite();  
        iceKingSprite();  
        buttons();  
        drawSprites();  
        break;  
      case 3:  
        background(bg1);  
        enlargeOver();  
        drawSprites();  
        break;  
      default:  
        break;  
    }  
  }  
}
```

