



PROYECTO JUEGO: DRAGON'S NOT EATING

ALUMNO: RAUDALES PALMA LEONARDO DE JESÚS

PROFESOR: M.I. MARCO ANTONIO MARTÍNEZ QUINTANA

SEMESTRE: 2021-1

FUNDAMENTOS EN PROGRAMACIÓN

26/ENERO/2021

CONTENIDO DEL PROYECTO

En este documento se abordará todo lo relacionado al proyecto Dragon's Not Eating, un juego de interfaz simple para el disfrute del consumidor. Primero, nos encontramos con una pequeña introducción a este ambiente que es la empresa de los videojuegos, la relación del proyecto con respecto al rubro general en el que se desarrolla y una pequeña prueba de lo que se busca en objetivos futuros. Después se brinda una descripción breve del programa, con qué intención se hizo y las funciones generales del mismo. Continuamos con un vasto desarrollo del proyecto, es decir, los pasos y actividades realizadas conforme se fue trabajando, desde el algoritmo, hasta el diagrama de flujo y pseudocódigos de algunos módulos del proyecto, una vez visto todos estos preámbulos, lo que sigue obviamente es el código fuente con el que se sustenta la ejecución del juego, esto es, todo el desarrollo de programación una vez que se empezó a desarrollar la parte práctica del videojuego, teniendo claro como base el esquema que se mencionó anteriormente. Este código fuente, además, está comentado en líneas de código para demostrar el uso de lo aprendido en el curso y para describir cada una de las funciones y características de este juego. Después sigue la parte demostrativa de este trabajo, como ya vimos un proyecto toma mucha organización, estructuras de realización para que esa idea que formamos en nuestra mente se convierta en algo real de manera ordenada y entendible, esto solo es comprobable, es decir, sabemos si ha sido un éxito o no, con la ejecución de dicha idea, en este caso para que nuestro videojuego se considere con proyecto realizado, este debe ser ejecutable sin ningún error y cumpliendo con las funciones que se esperaban en un inicio. Para esto, se brindan evidencias sobre los resultados, capturas e imágenes demostrativas que prueban la ejecución del programa y que aportan una imagen gráfica al lector sobre el uso práctico de este proyecto. También se añaden resultados comparativos con sus respectivas evidencias, la cotización del costo del proyecto, esto es, la diferencia entre el costo que su servidor esperaba cobrar por este proyecto cuando con lo único que se contaba era con la idea principal y el costo que ahora una vez realizado se estima para el mismo. Lo anterior también sucede con aspectos como el tipo de material y los tiempos que se pretendían al iniciar el juego y como realmente fue sucediendo todo este desarrollo a lo largo de los últimos meses. Ya en la penúltima parte de este documento, se pone a disposición el enlace respectivo donde se encontrará este documento en la nube, para lectura de cualquier persona y para futuras comparaciones con otros proyectos o con el avance futuro del mismo si se llegara a actualizar. Toda esta información también se encontrará disponible de manera digital en un vídeo explicativo dentro de la plataforma de Youtube, este enlace también estará en el documento. Para finalizar, se presentan las conclusiones, la utilidad del programa con respecto a empresas de desarrollo de tecnología y de importante trascendencia en este ambiente en la actualidad, y conclusiones personales por parte de su servidor, creador de este programa. También al final se encuentran tanto las referencias en las que se basó este proyecto como un pequeño manual de usuario, para el alcance de todos.

INTRODUCCIÓN

La industria de los videojuegos en la actualidad representa un mercado muy demandado por la población joven hoy en día, lo que en su tiempo fueron pequeñas imágenes con píxeles que comenzaban a cobrar vida dentro de la pantalla, hoy es un desarrollo insano de productos cuyos gráficos y programación superan, como siempre ha sido, los límites que la comunidad había impuesto anteriormente. No es raro imaginar que cuando se habla de la palabra juego llegue a la memoria un concepto muy visible en nuestras cabezas, un niño/niña detrás de un computador, una consola o un celular, disfrutando de las nuevas experiencias que proporciona cada año el mercado de desarrollo de juego y es que este rubro se hace tan conocido, que en toda familia un integrante al menos una vez en su vida haya jugado uno de estos programas.

Las características, la temática, el contexto y la jugabilidad en un juego son aspectos importantes para que atraiga a más consumidores, ya que por muy interesante que parezca, entre más se generan oportunidades nuevas de inmersión mejores juegos, más estricta es la demanda para estos. Son casos como un juego que este brutalmente diseñado en temas de gráficos e interfaz pero que su limitada aportación en el módulo de la trama o historia en la que se desarrolla no es suficiente para atraer a la comunidad, ocurre lo mismo en caso contrario, juegos que quizá desde antes de su lanzamiento generan una alta expectativa, ya sea porque detrás de ello yace una idea o concepto ya conocido o muy esperado de ver, o porque simplemente la presentación de la historia parece algo más que esperanzador para los amantes del gaming, pero que a falta de una buena representación gráfica o por errores presentados dentro del programa, ocasiona una pérdida de la aceptación por parte de los consumidores. Es por todo esto, que el crear un juego no es tarea sencilla, desde títulos grandes perdiendo el prestigio que tenían hasta ser una cosa lamentable, hasta pequeños juegos que no llegaron a ser lo que podían debido a fallas en los aspectos anteriores, por esto, lo más importante a la hora de pensar en integrarse a esta comunidad como desarrollador, es una construcción y elaboración ordenada y correcta de un proyecto que busque enamorar a los usuarios para así poder obtener el éxito esperado para cualquier creador.

Aquí es donde entran los pequeños juegos o aplicaciones simples que se baten la vida en lograr un éxito relacionado a una idea, esto es, que los usuarios puedan consumir su contenido tan limitado sin que se lleguen a aburrir, para esto se usan estrategias de marketing para poder encontrar lo que al usuario le gusta más, pequeñas cosas de las que nacen mentes adictivas a algo, esa sensación de volver a jugar simplemente porque querías o porque te dejó una grata sensación anteriormente. Se presentan muchos ejemplos de juegos que sin mucha profundidad logran enganchar a la comunidad y se vuelven un éxito (al menos por un tiempo determinado) de las plataformas que los albergan, la cuestión de si un juego así puede triunfar o no se ve estrictamente relacionada con la creatividad y sobretodo con el esfuerzo que hay detrás del programa y su creador.

DESARROLLO

Idea

La verdad es que pensé en varias ideas para un proyecto final, ya que en programación hay muchas posibilidades que se pueden llevar acabo, como por ejemplo alguna aplicación sencilla pero interesante, o programar una página web, quizá también programar una herramienta como un reloj o calendario. Pero sin duda hay una idea que me robó la curiosidad y que me convence más para poder ser mi primer proyecto realizado para programación. Mi idea es programar un juego de interfaz sencilla para computadora o celular, no tiene que ser independiente, es decir, creo que he visto algunas aplicaciones que te sirven de base para poder crear o programar juegos. Quiero que contenga actividades a realizar que sean sencillas pero que representen un esquema de jugabilidad interesante, con esto quiero decir que el interés del juego no dependa de una interfaz que pueda ser algo pobre, sería algo así como aprovechar las oportunidades de un programa simple lo más que se pueda. Sería un juego de pensar, o de elegir o realizar tareas, aún no pienso en escenarios ni en una idea del diseño en general, supongo que lo iré creando una vez que sepa que limitaciones tengo por ahora para crearlo. También la plataforma destino dependerá de esas limitaciones, quiero ya sea que sea totalmente funcional y capaz de compartir a mis amigos para que lo prueben, sí se puede dar la oportunidad también estaría muy bien una jugabilidad en línea.

Dada la extensión del proyecto, a continuación, se presentan solamente apartados parciales del proyecto en lo que respecta a algoritmos, diagramas de flujo y pseudocódigos.

Algoritmo MÓDULO DISEÑO VISUAL

PROYECTO: Videojuego simple.

Problema: Seguir el algoritmo para lograr la realización de un esquema base para el diseño de un videojuego de capacidades simples pero que aun así genere interés en el jugador.

ENTRADA: Concepto general. Diseño del juego, género y estilo. Diseño de historia o núcleo argumental. Interfaz.

SALIDA: Videojuego sencillo que provoque aceptación y agrado al jugador.

Algoritmo

1. Empieza pensando una idea o concepto general del juego en una perspectiva base.
2. Cuando tengas establecido bien la idea general del juego indica las premisas que necesitas para llevarlo a cabo.
3. Comienza a elaborar el diseño del juego a escala general, primero utilizando bocetos.
4. Comienza con una perspectiva más particular del diseño, pensando en aspectos más específicos. No concluyas hasta obtener la idea base que te convenza de cada elemento.
5. Realiza los bocetos y diseños de personajes, escenarios u otro material visual, auditivo y didáctico del que esté constituido el juego.
6. Confirma que el género y el estilo que buscas adaptar concuerde con los parámetros que están estableciendo y los diseños que estas efectuando, si no es así, regresa al punto 4.
7. Comienza a elaborar la historia o núcleo argumental en un boceto, comparándola con los diseños del material que ocupas, haciendo comparaciones del resultado obtenido con la idea base general que ideaste. Si no es así regresa al punto 1.
8. Confirma que el núcleo del juego como concepto este bien estructurado y cree la reacción esperada en el jugador, ya con todos los aspectos particulares añadidos.
9. Construye la idea de que interfaz de juego podría satisfacer tanto las preferencias hacia el proyecto como las capacidades que se tienen, una vez decidas la interfaz deberás adaptarla a todo el contenido del proyecto, todos los puntos de diseño que has elaborado.
10. Haz la revisión de cada aspecto si es necesario varias veces para confirmar decisiones y la calidad del diseño. Si el modelo no te convence regresa al paso 4.
11. Una vez confirmado que está todo en orden y logrado de la forma más satisfactoria, pasa la conjunción de bocetos y decisiones como un esquema que servirá de estructura base para el juego.
12. Listo, una vez tengas el esquema te servirá de guía para no alejarse del objetivo y preferencias principales, será la premisa base para comenzar a programar.

Algoritmo MÓDULO MOVIMIENTO DEL JUGADOR

PROYECTO: Videojuego simple.

Problema: Conseguir que el objeto a programar, es decir, el jugador, pueda moverse en distintas direcciones empleando las teclas del teclado.

ENTRADA: Cambio de posición del personaje con respecto a un parámetro y como herramientas, las teclas de dirección del teclado de un computador.

SALIDA: Movimientos del jugador, animaciones del personaje dentro de la pantalla del juego.

Algoritmo

1. Generar una escena nueva, un límite o fondo con características y propiedades del plano cartesiano, este será nuestro fondo o plantilla donde estará el personaje.
2. Se crea un objeto llamado personaje.
3. Añadir una imagen específica al jugador o personaje, para identificarlo dentro de nuestro fondo y así saber cuándo se está moviendo.
4. Se relaciona al personaje o jugador con un vector en el eje cartesiano.
5. Agregar una característica posición cuyas propiedades estén ligadas al vector del jugador, la posición cambia cuando se registra un cambio en el vector.
6. Bautizar el vector como movimiento.
7. Crear una condición que establezca que cuando el personaje cambie sus coordenadas dentro del vector movimiento se realice un cambio de posición.
8. Cuando se detecta un cambio de posición, sumarle a este valor la cantidad de 1.
9. Generar una relación entre el vector movimiento y las teclas del computador.
10. Si se presiona la tecla derecha, entonces el personaje se mueve hacia el valor de x positivo dentro del vector movimiento, la posición cambia a +1 hacia la derecha.
11. Si se presiona la tecla izquierda, entonces el personaje se mueve hacia el valor negativo de x dentro del vector movimiento, la posición cambia a -1 hacia la izquierda.
12. Si se presiona la tecla arriba, entonces el personaje se mueve hacia el valor negativo de la y dentro del vector movimiento, esto porque en el computador, el eje y está invertido, la posición toma -1 lo que genera que el personaje suba.
13. Lo contrario a lo anterior, el personaje baja.
14. Una vez se establezcan todos los movimientos respecto a cada tecla, asignar cada movimiento con el perfil o ángulo correctos, de la imagen que le otorgamos al principio al personaje.

Diagrama de flujo MODULO DISEÑO VISUAL

El siguiente diagrama de flujo corresponde a la realización del apartado visual del proyecto de videojuego del semestre 2021-1 correspondiente a la materia de fundamentos en programación.

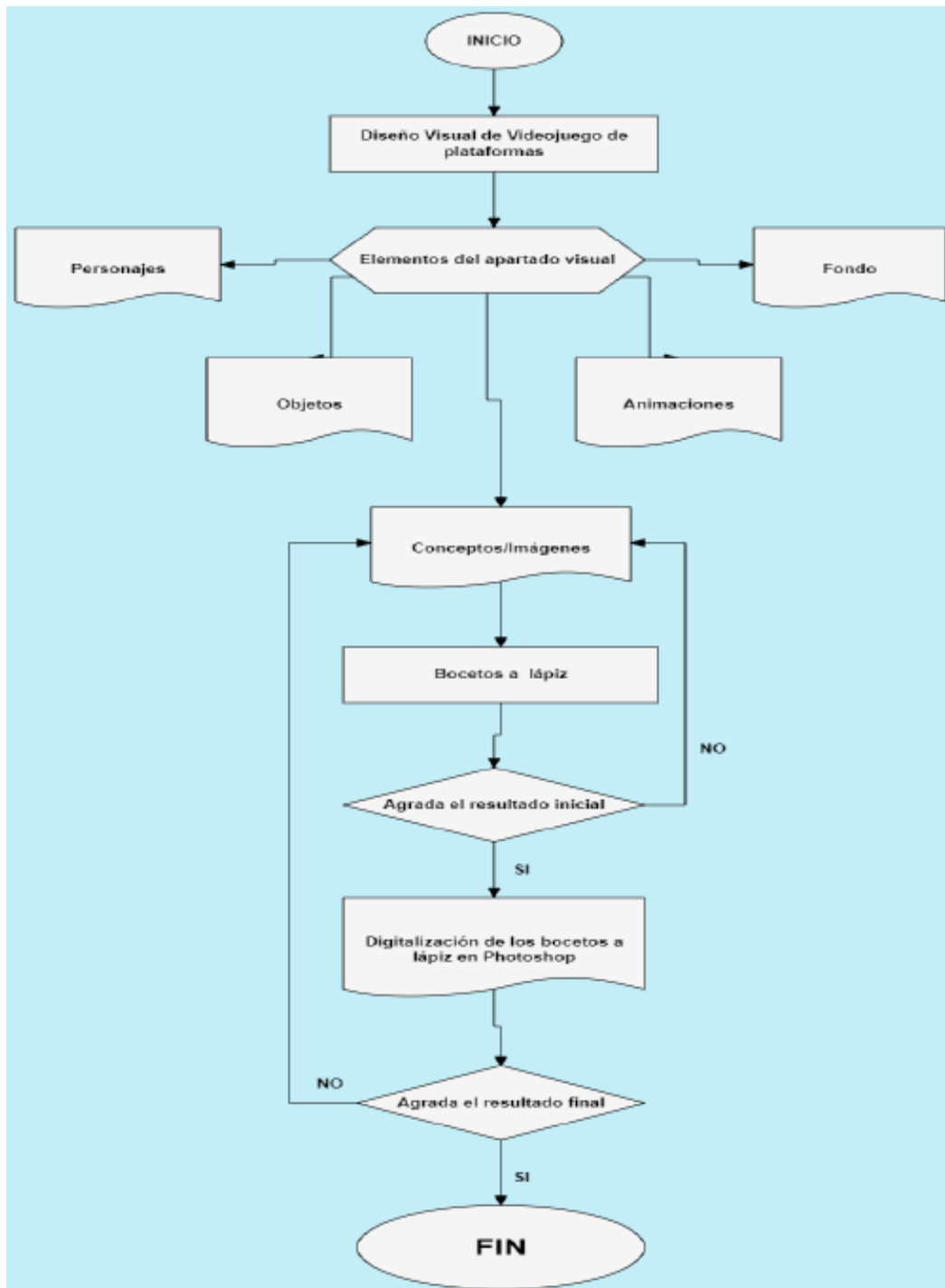
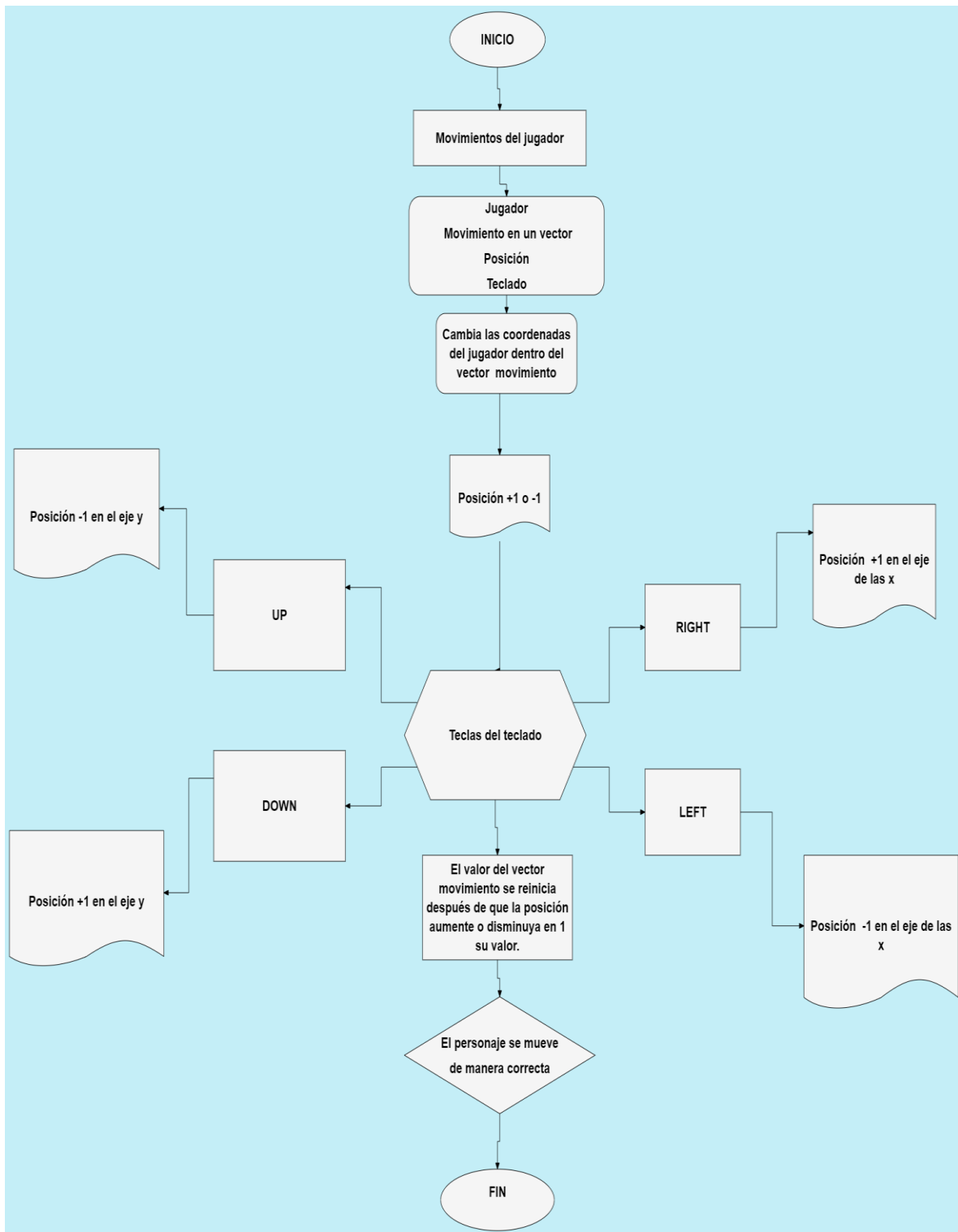


Diagrama de flujo MODULO MOVIMIENTOS DEL JUGADOR



Pseudocódigo MÓDULO DISEÑO VISUAL

INICIO

FUNC principal (Diseño visual videojuego) RET: "Diseño final"

a: CADENA

a: = "elementos visuales del proyecto"

SELECCIONAR (a) EN

CASO 1 ->

ESCRIBIR "personajes"

CASO 2 ->

ESCRIBIR "fondo"

CASO 3 ->

ESCRIBIR "objetos"

CASO 4 ->

ESCRIBIR "animaciones"

DEFECTO ->

ESCRIBIR "otro elemento visual"

FIN SELECCIONAR

b: CADENA

b: = Conceptos

Si b existe ENTONCES

ESCRIBIR "bocetos a lápiz"

FIN SI

SI "bocetos a lápiz" es correcto ENTONCES

ESCRIBIR "digitalizar bocetos a lápiz"

FIN SI

DE LO CONTRARIO

ESCRIBIR "volver a comprobar b"

FIN DE LO CONTRARIO

SI "digitalización de bocetos a lápiz" es correcto ENTONCES

ESCRIBIR "Diseño Final"

FIN SI

DE LO CONTRARIO

ESCRIBIR "volver a bocetos a lápiz"

FIN DE LO CONTRARIO

"digitalización de bocetos a lápiz" es correcto

RET: = "Diseño Final"

FIN FUNC

FIN

// >>> Diseño Final

Pseudocódigo MÓDULO MOVIMIENTOS DEL JUGADOR

INICIO

FUNC principal (IN GAME) RET: "movimientos del player"

Movimiento: = "vector"

Posición: int

Posición = Movimiento x, y

SI PRESS INPUT "arriba"

Posición -1 en el eje y

FIN SI

Movimiento = 0

SI PRESS INPUT "abajo"

Posición +1 en el eje y

FIN SI

Movimiento = 0

SI PRESS INPUT "izquierda"

Posición -1 en el eje x

FIN SI

Movimiento = 0

SI PRESS INPUT "derecha"

Posición +1 en el eje x

FIN SI

Movimiento = 0

RET=" movimientos del player"

FIN FUNC IN GAME

FIN

//>>>Movimientos del player

Código Fuente

Código y programación realizados en GDScript.

Código de programación en Godot Engine, Escena: **Mundo**

```
1  extends Node
2
3  #Declaración de variables.
4  export (PackedScene) var Manzana
5  var Score
6
7  #Función de inicio
8  func _ready():
9      >|
10     #Creación de enemigos de manera aleatoria.
11     >| randomize()
12
13     #Función para un nuevo juego.
14  func nuevo_juego():
15      >|
16      #La puntuación al inicio es igual a 0.
17      >| Score = 0
18      >|
19      #La posición del jugador al empezar cada juego es la que establecimos.
20      >| $Player.inicio($PosicionDeInicio.position)
21      >|
22      #El temporizador de inicio empieza al iniciar un nuevo juego.
23      >| $InicioTimer.start()
24      >|
25      >| #Relacionamos la interfaz con lo que se muestra en pantalla del mundo.
26      >| $Interfaz.mostrar_mensaje("Let's Fuking Go!")
27      >| $Interfaz.update_score(Score)
28      >|
29      >| #Cuando se dé a nuevo juego empieza la música.
30      >| $Musica.play()
31
32     #Función de juego terminado, se activa con la señal "golpe".
33  func game_over():
34      >|
35      #El contador de puntos se detiene cuando el jugador recibe un golpe.
36      >| $ScoreTimer.stop()
37      >|
38      #El temporizador que genera enemigos se detiene.
39      >| $ManzanaTimer.stop()
40      >|
41      >| #Relacionando el game over de la Interfaz dentro del Mundo.
42      >| $Interfaz.game_over()
43      >|
```

Código de programación en Godot Engine, Escena: **Mundo**

```
44 >| #Se reproduce el sonido de la muerte del jugador cuando acaba el juego.
45 >| $AudioMuerte.play()
46 >|
47 >| #La música para cuando el personaje muere.
48 >| $Musica.stop()
49
50 #Función que conecta el temporizador de inicio de juego.
51 >| func _on_InicioTimer_timeout():
52 >|
53 #Una vez empiece el contador de inicio, el contador de enemigos inicia.
54 >| $ManzanaTimer.start()
55 >|
56 #Una vez empiece el contador de inicio, los puntos empiezan a subir.
57 >| $ScoreTimer.start()
58
59 #Función que conecta el temporizador para los puntos.
60 >| func _on_ScoreTimer_timeout():
61 >|
62 >| #Cada que se reinicia el temporizador los puntos aumentan en 1.
63 >| Score += 1
64 >|
65 >| #Se vincula la interfaz con la actualización de puntos.
66 >| $Interfaz.update_score(Score)
67
68 #Función del temporizador que crea enemigos.
69 >| func _on_ManzanaTimer_timeout():
70 >|
71 #Usamos el nodo que recorre la pantalla de juego para crear enemigos al azar.
72 >| $Camino/ManzanaPosicion.set_offset(randi())
73 >|
74 #Usamos la variable que declaramos al inicio para indicar el enemigo a crear.
75 #La variable manzana contiene la escena donde están los sprites del enemigo.
76 >| var M = Manzana.instance()
77 >| add_child(M)
78 >|
79 #Seleccionar una dirección, rotación y velocidad para que se generen
80 #las manzanas.
81 >| var d = $Camino/ManzanaPosicion.rotation + PI / 2
82 >| M.position = $Camino/ManzanaPosicion.position
83 >| d += rand_range(-PI / 4, PI / 4)
84 >| M.rotation = d
85 >| M.set_linear_velocity(Vector2(rand_range(M.velocidad_min, M.velocidad_max), 0).rotated(d))
86
87
```

*La programación de este juego está hecha dentro del entorno de Godot Engine, en este entorno, se agregan nodos de importancia en la ejecución dentro de nuestro viewport o pantalla del juego, estos reciben el nombre de escenas, cada escena con una función recibe su propio script o código de programación, al final mediante el programa se conectan todas las escenas para que el juego funcione, estableciendo como principal escena y la que alberga a todas las demás, a la escena del “Mundo”, cuyo código es el que se ha mostrado.

Código de programación en Godot Engine, Escena: [Player](#)

```
1  extends Area2D
2
3  #Declaración de variables.
4  export (int) var Velocidad
5  var Movimiento = Vector2()
6  var limite
7  signal golpe
8
9  #Función de inicio.
10 ◀ func _ready():
11     >|
12     >|  #El personaje permanece oculto al ejecutar el programa.
13     >|  hide()
14     >|
15     >|  #Indicamos que la variable limite es igual a la pantalla del juego.
16     >|  limite = get_viewport_rect().size
17
18  #Función de proceso.
19 ◀ func _process(_delta):
20     >|
21     #Reiniciamos el valor de Movimiento cada vez que se pulsa una tecla.
22     >|  Movimiento = Vector2()
23     >|
24     #Movimientos del jugador.
25     #Indicando que al precionar la tecla se sumen valores a la variable Movimiento.
26 ◀ >|  if Input.is_action_pressed("ui_right"):
27     >|  >|  Movimiento.x += 1
28 ◀ >|  if Input.is_action_pressed("ui_left"):
29     >|  >|  Movimiento.x -= 1
30 ◀ >|  if Input.is_action_pressed("ui_down"):
31     >|  >|  Movimiento.y += 1
32 ◀ >|  if Input.is_action_pressed("ui_up"):
33     >|  >|  Movimiento.y -= 1
34     >|
35     #Normalizar la velocidad para que al ir en diagonal no aumente nos veces.
36 ◀ >|  if Movimiento.length() > 0:
37     >|  >|  Movimiento = Movimiento.normalized() * Velocidad
38     >|
39     #Cambio de posición con respecto a la variable Movimiento.
40     >|  position += Movimiento * _delta
41     >|
42     #Se establece un limite para que el jugador no salga de el.
43     >|  position.x = clamp(position.x, 0, limite.x)
44     >|  position.y = clamp(position.y, 0, limite.y)
45     >|
```

Código de programación en Godot Engine, Escena: **Player**

```
45 >|
46 #Relacionamos el movimiento del player con su respectivo sprite de animación.
47 >| if Movimiento.x != 0:
48 >| >| $Sprite_player.animation = "lado"
49 >| >| $Sprite_player.flip_h = Movimiento.x < 0
50 >| elif Movimiento.y < 0:
51 >| >| $Sprite_player.animation = "espalda"
52 >| else:
53 >| >| $Sprite_player.animation = "frente"
54
55 #Función cuando el personaje entra en colisión con otro cuerpo.
56 >| func _on_Player_body_entered(_body):
57 >|
58 #El personaje desaparece al recibir la señal "golpe".
59 >| hide()
60 >| emit_signal("golpe")
61 #Se desactiva la máscara de colisión una vez el player reciba un golpe para
62 #que el evento ocurra una sola vez.
63 >| $Colision.disabled = true
64 >|
65 #Cuando el player regresa a la posición inicial se restauran las propiedades
66 >| func inicio(pos):
67 >| position = pos
68 >| show()
69 >| $Colision.disabled = false
70
```

*Programación y código referente únicamente al jugador.

Código de programación en Godot Engine, Escena: **Enemigo**

```
1 extends RigidBody2D
2
3 #Declaración de variables.
4 export (int) var velocidad_min
5 export (int) var velocidad_max
6 var tipo_manzana = ["grande", "chiquito"]
7
8 #Función de inicio.
9 >| func _ready():
10 >|
11 #Elegir alguno de los 2 tamaños del sprite del enemigo de manera aleatoria.
12 >| $AnimatedSprite.animation = tipo_manzana[randi()% tipo_manzana.size()]
```

Código de programación en Godot Engine, Escena: **Enemigo**

```
13 >|
14 #Máscara de colisión más grande si el sprite del enemigo elegido es grande.
15 >| if $AnimatedSprite.animation == "grande":
16 >| >| $CollisionShape2D.scale.x = 1.25
17 >| >| $CollisionShape2D.scale.y = 1.25
18
19
20 #Función que sirve para cuando las manzanas salen de la pantalla.
21 >| func _on_Visibilidad_screen_exited():
22 >|
23 #Los enemigos se eliminan una vez salen de la pantalla.
24 >| queue_free()
25
```

*Este código pertenece a la escena donde se encuentran los enemigos, estos tienen el sprite de animación de una manzana, por ende, los nombres de las variables.

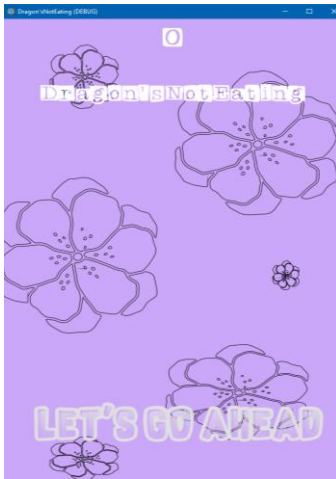
Código de programación en Godot Engine, Escena: **Interfaz**

```
1 extends CanvasLayer
2
3 #Declaramos la señal de inicio del juego
4 signal iniciar_juego
5
6 #Función que relaciona el mensaje de bienvenida del juego.
7 >| func mostrar_mensaje(texto):
8 >|
9 >| $Mensaje.text = texto
10 >| $Mensaje.show()
11 >| $MensajeTimer.start()
12
13 #Función que incluye lo que ocurre al terminar el juego.
14 >| func game_over():
15 >| mostrar_mensaje("Game Over")
16 >| yield($MensajeTimer, "timeout")
17 >| $BotonPlay.show()
18 >| $Mensaje.text = "Dragon'sNotEating"
19 >| $Mensaje.show()
20
21 #Función del aumento de los puntos con respecto al tiempo
22 >| func update_score(Puntos):
23 >| $ScoreLabel.text = str(Puntos)
24
25 #Función del temporizador del mensaje de inicio.
26 >| func _on_MensajeTimer_timeout():
27 >| $Mensaje.hide()
28
29 #Función para lo que ocurre al presionar el botón de inicio.
30 >| func _on_BotonPlay_pressed():
31 >| $BotonPlay.hide()
32 >| emit_signal("iniciar_juego")
33
```

*Este código abarca la programación de la interfaz del juego, títulos, puntajes, etc.

RESULTADOS

Dragon'sNotEating

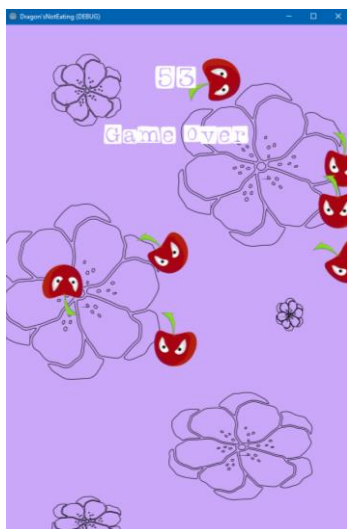
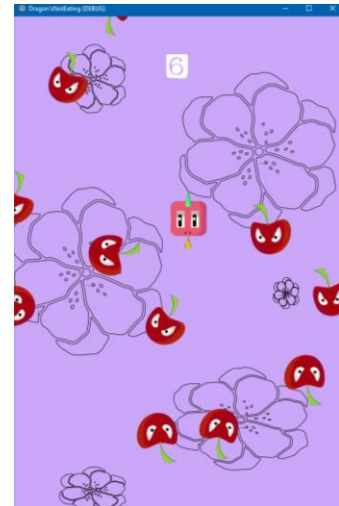


PANTALLA DE INICIO

Al ejecutar el juego aparece la pantalla de nuevo juego, en esta se muestra la interfaz del juego, aquí están detalles como el contador de puntos en cero, el título del proyecto y el botón de start, en cuanto lo presionas desaparece junto con el título, aparece una leyenda que le indica al jugador que se prepare, comienza el contador de puntos e inicia el juego.

IN GAME

El propósito del juego es el de esquivar a los enemigos que son unas manzanas que se generan aleatoriamente en la pantalla, por el mayor tiempo posible, nosotros tendremos la labor de sobrevivir de estos jugosos enemigos en los pies de un dragón rojizo que, por alguna razón, no quiere comer manzanas. Cuando se presiona start empieza a sonar la música del juego y permanece sonando hasta que se alcanza el game over, esto ocurre cuando una manzana golpea al player, se reproduce un efecto de sonido al impactar y termina el juego.



GAME OVER

Cuando el player o jugador colisiona con un enemigo (manzanas), este desaparece, sale la leyenda de juego terminado y el contador de puntos se detiene. La música deja de sonar y como se observa, el temporizador que genera los enemigos se detiene y estos ya no siguen apareciendo hasta que se vuelva a presionar el botón de start de nuevo. Una vez se detiene la creación de enemigos, en pantalla solo quedan las manzanas que ya se generaron y estas solo desaparecen hasta dejar la pantalla vacía.

Recursos informáticos: Software y hardware

Estimación Octubre 2020	Ocupación Enero 2021
<p>Pues creo que obviamente una cosa esencial es una computadora con características que me permitan ejecutar tanto la programación del juego, como el diseño digital de este. Varios bocetos, esto incluye pues quizá cuadernos y material de dibujo para diseñar los elementos del juego. Necesito conocimiento de distintos lenguajes de programación por muy básico que sea para crear el juego, obviamente pues esto representa las asesorías o cursos necesarios. Photoshop y dominio de este para ejecutar ajustes de aspectos visuales. Aún no estoy seguro si necesito licencias, tendría que investigar un poco más al respecto. Una tableta para dibujo digital también podría ayudar. Programas dado el caso de tener la opción de crear un juego a partir de un programa.</p>	<p>Investigando más a fondo, me di cuenta que existían muchas guías o bases para crear el material que buscaba en plataformas ya conocidas como YouTube, incluso el tutorial en el que se basa mi proyecto. Con esto aprendí que realmente no necesitas equipos o recursos tan potentes, de hecho, la computadora en la que elaboré todo el proyecto es de gama medio-baja, así como las licencias no fueron necesarias pues a pesar de que la estructura del proyecto se basa en un tutorial, todos los recursos de mi versión fueron generados por mí, esto lo hice con alternativas de software libre, es decir, programas gratuitos muy útiles y que no necesitaban un hardware tan potente. Obviamente supongo que la necesidad y conocimiento dependen mucho del tipo de juego que deseas realizar, pero en mi caso sin duda se requirió de mucho menos cosas de las que tenía en mente en un inicio.</p>

Costos asociados al proyecto

Estimación Octubre 2020	Ocupación Enero 2021
<p>¿Cuánto estarías dispuesto a pagar por que alguien desarrolle tu proyecto?</p> <p>Creo que primero debería informarme bien sobre un precio de referencia que esté presente en el mercado. Incluyendo en ese precio todos los aspectos ya sea de programación. Diseño y eficacia que me dan como garantía. Si simplemente especulo, creo que pagaría lo que considero que vale mi proyecto. Pienso que sería una cantidad aproximada a los \$5,000 pesos, más o menos, porque es simple</p>	<p>¿Cuánto estarías dispuesto a pagar por que alguien desarrolle tu proyecto?</p> <p>Creo que en apartado de costos no pude estar más equivocado, al principio creí que podría pagar \$5,000 por la realización de mi proyecto, esto fue porque creía que a pesar de ser simple se necesitaría de conocimientos muy específicos en el tema, cosa que se tiene que reconocer en el costo. Es cierto, requiere de conocimientos vitales en el rubro, pero son mucho menos que lo que pensaba, eso me hace rebajar el costo que aceptaría por</p>

quizá, pero requiere conocimientos específicos para poder llevarse a cabo.

¿Cuánto cobrarías si tú lo hicieras?

Yo creo que cobraría un poco más a lo que estipulé en la primera pregunta, considerando el potencial que pueda tener mi proyecto, si considero que el videojuego puede llegar a crecer y a mejorar de forma que se vuelva competitivo en el mercado, pues calcularía el precio que es lógico para las expectativas que se llegaran a tener para el futuro. En este caso el precio que cobraría lo aumentaría a \$7,000 respecto a la cantidad que dije que pagaría.

la realización de mi proyecto, ya que no es una labor tan complicada como esperaba, si se reconoce ya muy exagerado el esfuerzo de la persona, no solo en programación si no en la aportación de buenos recursos, estaría dispuesto a pagar no más de \$500.

¿Cuánto cobrarías si tú lo hicieras?

Creo que cobraría incluso menos que lo que daría por su realización, unos \$200 pues debo reconocer que me divertí mucho y estaría dispuesto a rebajar el costo pues no creo que existiera una satisfacción ahora mismo que ver a alguien disfrutar del proyecto que, aunque requiere cierto nivel de conocimientos pues sigue siendo simple.

Diagrama de Gantt diseñado en octubre del 2020

Creo que lo realizaría por etapas primero muy generales y conforme vaya adquiriendo más conocimiento pues estableciendo ya lapsos de realización más específicos.

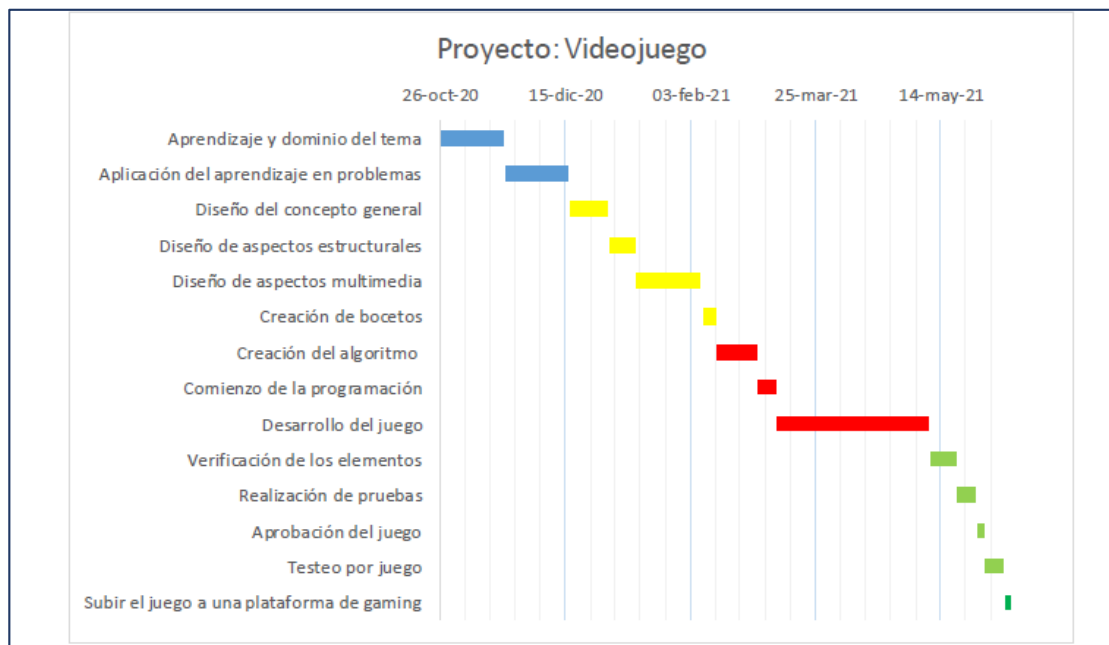


Diagrama de Gantt resultante en enero del 2021

Como se puede notar, existe mucha diferencia de tiempo estimado con el que se necesitó realmente para algunas etapas.



Todo este documento se encuentra disponible y explicado de manera expositiva en un vídeo en mi canal dentro de la plataforma de YouTube, se puede acceder desde el siguiente enlace:

<https://www.youtube.com/channel/UCiyHRBkuEUvMfrtnjMmu05w>



También se puede consultar dentro de la plataforma de repositorio de software libre GitHub, el enlace para revisar el documento escrito es el siguiente:

<https://github.com/IngenieroRaudales/Proyecto-Final>



Guía de uso de usuario

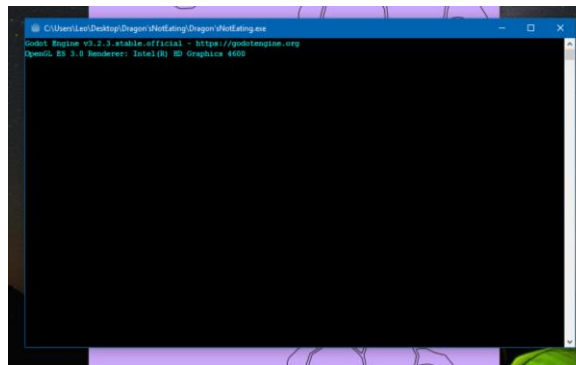
- ❖ Dentro del enlace de repositorio de GitHub se encuentra un archivo .rar llamado Dragon'sNotEating, este mismo se puede descargar desde el repositorio.
- ❖ Una vez descargado, puedes descomprimir el archivo que te dejara una carpeta con el mismo nombre.



- ❖ Dentro de esta carpeta se encuentra el ejecutable del juego, este tiene el logo de Godot Engine y el nombre del juego. Cabe mencionar que el archivo .pck que se encuentra adentro debe estar siempre en una misma carpeta con el ejecutable, de lo contrario el juego no funcionará.

Acceso rápido	Nombre	Fecha de modificación...	tipo	Tamaño
★ Acceso rápido	Dragon'sNotEating	27/01/2021 11:20 ...	Aplicación	40,505 KB
Este equipo	Dragon'sNotEating.pck	27/01/2021 11:20 ...	Archivo PCK	5,887 KB
Descargas				

- ❖ Da doble click en el ejecutable y listo, comienza a jugar ¡. Cuando se ejecuta el programa aparece una ventana negra parecida a la de la terminal de Windows, solo haz caso omiso de ella, esta solo sirve para indicar que el juego está corriendo. ¡No la cierres!



- ❖ Los controles de juego son muy sencillos, simplemente oprime las flechas de dirección ←, →, ↑, ↓ en tu teclado para desplazarte en la pantalla. Evita que las malvadas manzanas te golpeen, haz tu mejor esfuerzo ¡

CONCLUSIONES

Este proyecto a mi parecer, por muy pequeño y simple que es, se ve directamente relacionado con el software libre, dado que tanto la realización del mismo, el desarrollo, como la creación de recursos que se ocuparon, como lo fueron diseños de personajes, creación de pistas de sonido que se acoplasen al juego y animación que se les brindaron a los diseños, se realizaron con programas de software libre, ósea que son gratuitos y para libre desarrollo de creadores de todos tipos, de hecho el mismo entorno de juego, hablo de Godot Engine fue elegido por esa misma razón, por tratarse de software libre, es una plataforma diseñada con el propósito de que creadores sencillos sean capaces de diseñar juegos aun teniendo poca experiencia, Godot también permite que los usuarios no solo compartan sus plantillas o desarrollos de juegos sino que también que otros desarrolladores sean capaces de editar sus creaciones y agregar los aspectos que gusten. Así mismo, también existe la posibilidad de exportar tus proyectos, como fue el caso de mi juego, a distintas plataformas desde Godot, permitiendo así que se rompa la barrera de la compatibilidad para distintas plataformas, por eso, creo que mi proyecto como muchos otros que hoy en día se están desarrollando, es un grano de arena en el inmenso mar del desarrollo libre de programas y espero poder seguir contribuyendo a la comunidad.

Relacionado con la IEEE y la industria que vemos hoy en día creo que mi proyecto significa un futuro “prometedor” en lo que respecta a la tecnología, uno de los temas más cuestionados en el tema del desarrollo tecnológico en la actualidad es si se va a poder cubrir la demanda de expertos en programación, construcción y diseño de nuevas amalgamas dentro del mercado, se cuestiona que para el avance de la demanda que se tiene por parte de la sociedad con respecto a nuevos dispositivos y aplicaciones puede ser cubierta en un futuro no muy lejano, esto es, que entra más necesaria se vuelve la influencia de la tecnología en el día a día más se necesita de nuevos desarrolladores que aporten para que esta demanda se cumpla. Es por eso que creo que, aunque aún soy una persona inexperta, he comprobado con este proyecto que puede haber un futuro desarrollador en mí, toda la construcción obviamente se realizó dentro de las normativas de la IEEE pues son las que rigen dentro del desarrollo tecnológico al que están apegados la mayoría de programas de creación de videojuegos que existen. Espero algún día sea capaz de desarrollar un juego que permita no solo el entretenimiento del usuario, sino que inmersa a más personas con el interés de llegar a desarrollar algo similar, entre más compañeros creadores seamos dentro de este hermoso rubro más obtenciones ingeniosas se obtendrán y eso es algo que nos beneficia a nosotros los creadores, a los consumidores y sobre todo a la sociedad que se sigue abriendo paso hacia el avance de la tecnología.

Conclusiones Personales:

Me siento muy complacido con mi proyecto ya logrado, antes que nada, debo reconocer que ha sido muy divertido todo este proceso de creación, desde que tuve la idea puesta en mi cabeza, la creación de un algoritmo que pudiese expresar todo lo que en mi mente yacía. Ha sido un recorrido de lo más interesante, características como precios, tiempos y demás cambiaron totalmente a lo que me esperaba. Luego llegó la hora de por fin empezar a programar, esta fue mi parte favorita ya que, aunque me apoye de un tutorial, genere todos mis recursos y realice muy pequeños ajustes dentro del código que solo yo entendería, me encanto cada parte del diseño, de verdad, cree los bocetos y diseños desde cero y ver como tomaban color y forma para luego ser integrados en el entorno de Godot y volverse formas que cobraban vida dentro de la reproducción de la escena era algo mágico, más porque entendía el porqué de los códigos, era como si yo estuviera hablando en un lenguaje que solo conocemos la computadora y yo y que esta última iba cooperando conmigo para que del esfuerzo de los dos naciera algo increíblemente complaciente. Cada parte recuerdo me impresionaba más que la anterior, debo aclarar que este es mi primer proyecto o programación y todo resulta nuevo para mí. Cuando el personaje que cree a partir de primero unos rayones que después edité empezó a moverse con las teclas de mi teclado no sabría decir la satisfacción y la felicidad que se reprodujo en mi rostro, más porque sabía que yo mismo había hecho eso y que estaba quedando como lo esperaba, que mis esfuerzos había generado frutos. Lo mismo hice para cada uno de los aspectos que iba logrando, no me arrepiento de mis expresiones pues siento que es parte de este trabajo en el que nos estamos apenas involucrando. Otra de mis partes favoritas fue la creación de la música de fondo, recuerdo estar divirtiéndome generando ritmos muy aleatorios a los que solo yo les entendía y al final, cuando le tomé el truco pues conseguí un ritmo que para mi sorpresa quedó mejor con el juego de lo que esperaba, además de que la programación de este dentro del juego fue bastante sencilla. Ya para finalizar, decir que estoy muy agradecido con la institución, la carrera, con mi maestro y conmigo mismo por la realización de este proyecto, creo que suelo tener dudas constantemente sobre mi futuro (como cualquier joven de mi edad) y a veces las incertidumbres suelen ser más grandes que los sueños, este trabajo me ha dado una pista de lo que puedo hacer y sobretodo quiero hacer, y que quizá el camino es muy largo todavía pues apenas es el comienzo, pero que si me esfuerzo y continuo con paso firme puedo achicar esa distancia, algún día quizá recordaré este proyecto con cariño y sea capaz de decirme leyendo esto que por fin lo logré, que conseguí lo que quería después de tanto esmero y dedicación, y comparar lo que creo en 10-20 años con este proyecto que fue la semilla de todo, que fue la manzana que incitó a seguir pecando y programando dentro de este mundillo del desarrollo de software que está triunfando en la actualidad.

REFERENCIAS

Este proyecto está basado en la serie de videos de YouTube del creador Adderly Céspedes, los datos de esta misma serie tutorial son los siguientes:

Nombre de los vídeos:	Fecha de publicación:
❖ Como crear un juego simple 2D en Godot – Parte 1	10 septiembre 2018
❖ Como crear un juego simple 2D en Godot – Parte 2	14 septiembre 2018
❖ Como crear un juego simple 2D en Godot – Parte 3	12 octubre 2018
❖ Como crear un juego simple 2D en Godot – Parte 4	17 octubre 2018
❖ Como crear un juego simple 2D en Godot – Parte 5	21 octubre 2018
❖ Como crear un juego simple 2D en Godot – Parte 6	28 octubre 2018

Nombre del canal:

- ❖ Adderly Céspedes

Fecha de consulta:

Desde octubre del 2020 hasta enero de 2021

Enlace de la lista de reproducción de la serie tutorial:

https://youtu.be/BY15nUgU-vs?list=PL5K_XeiglfdJeaJ3-_YgnzD711StmJkZh

GLOSARIO

Sprites: Término referente a los recursos en imágenes que se le dan a un personaje, es decir, es el diseño gráfico de este.

Escena: Es un apartado de trabajo dentro del entorno Godot Engine que se refiere a un “evento” o algo que sucede dentro del juego, es una escena principal donde ocurre toda la reproducción de este, y escenas hijas que son eventos añadidos a la escena principal.

Ejecutable: Archivo aplicación que se puede correr en el dispositivo para el que está destinado.

Hasta la próxima ¡