



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Fundamentos de Programación



Proyecto: Videojuego 2D de plataformas.

Nombre del videojuego: La Rana Ninja



Semestre 2021-1

Profesor: Marco Antonio Martínez Quintana.

Alumno: Rosario Vázquez José André

Fecha de elaboración: 30 de enero del 2021

Contenido de mi Proyecto.

Mi proyecto consiste en la elaboración de un videojuego de plataformas parecido a lo que es Mario Bros; pero debido al tiempo y unos problemitas en la computadora me limite en hacer un mapa. Este videojuego fue desarrollado en unity, ya que es uno de los softwares más completos que hay gratuitos, además su lenguaje de programación es C#, que a mi parecer es muy parecido a JAVA.

Para el videojuego utilice un plantilla de personajes y texturas para crear mapas en juegos de plataformas que se encuentran en el mismo unity, para desarrollarlo necesite apoyo directamente de videos, ya que no conocía nada de este mundillo de la programación de videojuegos; pero interesaba mucho como como es el proceso de crear un juego y aproveche en este proyecto para darme una idea.

Este proyecto contendrá el tema que aborda mi proyecto, es decir, para qué público está dirigido y por qué, además de la visión que le tengo al proyecto en un enfoque emprendedor.

El videojuego contiene varios códigos en C# y dada la complejidad de la programación se va a mostrar el algoritmo, diagrama de flujo y pseudocódigo para que tenga una forma más estructurada y sea más fácil de comprender lo que se hizo, para demostrar los resultados se va a tomar capturas de esta, además de un video en donde se va apreciar las mecánicas de este. También se va a encontrar una comparativa de lo que tenía previsto hace meses en el proyecto con lo se consiguió en este último mes.

Este videojuego tiene como carácter principal una Rana Ninja el cual tiene que agarrar todas las frutas y no morir por los spikes , además en el mapa se encuentra un enemigo que está subiendo y bajando constantemente hay que evitar tocarlo porque si nos toca muere el personaje.

Por último, mis conclusiones sobre el proyecto, que van a ser de suma importancia ya que definirá el futuro de este proyecto, de que si voy a continuar para mejorarlo y empezar a generar algo serio a base de este o dejar a un lado el proyecto para encontrar algún otro que me guste más, ya que al haber intentado hacerlo no me agrado el proceso y mejor intentaría otro concepto de proyecto.

Tema que aborda mi proyecto.

Este proyecto, está destinado para el entretenimiento personal ya que no es un juego multijugador, además al ser de plataformas suele tener cierta comparación con Mario Bros.

Los videojuegos se pueden definir como como una aplicación interactiva orientada en el entretenimiento que, a través de la programación se puede controlar, editar y simular distintas experiencias, para representarlo o demostrarlo en una pantalla que sea capaz de apreciarlo.

Este tema de videojuegos es controversial ya que hay comunidades que desprecian a este, por que puede generar adicción a algunos individuos , además de poder generar daño a tus ojos por estar tanto tiempo de frente de una pantalla; pero lo dicho no quita el arte de un videojuego, ya que no solo es programar a un personaje predeterminado y diseñar el mapa (así lo pensaba), sino que lleva un proceso de diseño de caracteres como del fondo del videojuego, además de diseñar los distintos movimientos que va a tener el personaje, el ambiente musical, etc. Todo esto para integrarlo en algún motor físico que haga realidad en la pantalla los movimientos, animaciones y diseños dentro un espacio.

Conociendo todo este trasfondo es el justificante de por qué un videojuego se tarda demasiado en salir a la venta, ya que al diseñar o un rediseñar un videojuego no es fácil, además de hacer todas las pruebas posibles dentro del videojuego para no encontrar un fallo (bug) dentro de él, todo debe ser minuciosamente inspeccionado ya que un error dentro del juego puede arruinar la experiencia de la persona que está jugando.

En fin, los videojuegos son un medio interesante que no son malos del todo, sino que nos ayudan a mejorar tus reflejos y tu inteligencia para resolver problemas dados, además hay videojuegos donde estimulan tu creatividad y tu trabajo en equipo, ese es el motivo por el cual decidí hacer uno aunque sea muy pequeño y no tan complejo este desarrollo mi creatividad para hacer un mapa con un personaje en condiciones.

Idea con enfoque de emprendimiento.

Para mi proyecto que fue la creación de un videojuego 2D de plataformas, lo primero que haría es integrar mi juego en una aplicación en la que puedas comprar y descargar videojuegos, esto para que se dé a conocer mi trabajo y posiblemente ser contactado por alguna desarrolladora de videojuegos que desee comprar mi proyecto o mejorarlo en conjunto.

Este videojuego está pensado para jugadores de todo tipo, desde inexpertos hasta experimentados. Donde el único objetivo del videojuego es disfrutarlo sin necesidad de enojarse por no pasarte el nivel; ya que no está pensado para un ámbito competitivo.

La manera en la que se empezará a monetizar no será por anuncios, ya que por experiencia propia es muy molesto que aparezca un anuncio cada 5 segundos; por lo que una manera de monetizar es diseñar distintos trajes o distintos personajes y venderse por la moneda del juego, donde esta moneda debe ser comprada por dinero real, dependiendo de la rareza de las skins será más cara o barata.

Una forma interesante de promover el juego y que llegue más gente a jugarlo sería pagarle a algún youtuber o streamer que juegue este juego en un video o durante un tiempo determinado y dar su opinión sincera del juego (si le gusto o no).

Como mi videojuego consta actualmente de un solo mapa lo trataría de completar en más tiempo para así poder promocionarlo y venderlo dentro de las aplicaciones antes mencionadas o basarme de este proyecto para hacer uno completamente mejor en el que sea más grande el mapa y con mis propios diseños de los personaje y texturas de los mapas, ya que en el mercado de videojuegos no es tan común como parece encontrar un juego de plataformas 2D, que sea gratis sin anuncios y con venta de skins para verte diferente dentro del juego.

Esa es la visión que le tengo a mi proyecto.

Algoritmo de mi Proyecto (instalación y desarrollo del mapa; parte del algoritmo ya que es muy extenso).

1. Descargar unity Personal (version 2019.4.18f1) para realizar el videojuego.
2. Descargar Visual Studio community con la extensión de programación de unity.
3. Una vez instalado Unity dar click a crear proyecto.
4. Estando en la pestaña poner el nombre de tu proyecto y elegir juego en 2D.
5. Dar click a crear.
6. Estando dentro del motor físico, dar click arriba a la izquierda en "edit" y seleccionar "preferences".
7. Estando en External Tools, seleccionar External Script Editor y elegir la versión de Visual Studio.
8. Cerrar Preferences.
9. Estando en el motor, dar click en la opción "window", estando ahí seleccionar "Asset Store"
10. Estando en el "Asset Store", buscar "Pixel Adventure 1" seleccionarlo y descargar.
11. Una vez descargado, dar click en la opción "importar".
12. Una vez importado se encontrará la carpeta "Pixel Adventure" en la parte inferior del motor (depende de la versión de unity), que contiene todas plantillas para la creación de mapas y con personajes.
13. Buscar en la carpeta la subcarpeta "Main Characters" .
14. Estando en la carpeta "Main Characters" dar click, donde hay 4 carpetas con 4 personajes correspondientes.
15. Elegir el personaje (en mi caso se seleccionó Ninja Frog).
16. Estando dentro de la carpeta de tu personaje, buscar la animación con el nombre "idle".
17. Una vez ubicada la animación "idle", arrastrarlo en la parte central de tu "Scene" del motor físico.
18. Al arrastrarlo se abrirá el explorador de archivos, donde vas a agregar el nombre que quieras a tu animación con un ".anim" (en mi caso de la versión se hace directo)
19. Dar click a guardar.

20. Una vez guardado, en la interfaz de tu motor se encuentran tres botones con su símbolo, uno de "Play", "Pause", "Step".
21. Estando en tu escena con nombre en unity de "Scene", puedes ver un puntito que es tu personaje donde puedes acercarte con el scroll de tu mouse.
22. Una vez ubicado tu carácter en la escena puedes presionar play, donde se va a ver la animación de tu personaje.
23. Quitar la función play volviendo a dar click en el botón "play", para continuar trabajando.
24. Seleccionar el módulo "Main Camera" que se encuentra en la parte izquierda de tu motor.
25. Al dar click se abren las opciones con la que vamos a modificar el tamaño de nuestra escena.
26. Estando en este módulo buscamos la opción size y como valor predeterminado aparece "5", donde vamos a cambiar el número anterior por un "1.5".
27. Con el tamaño ya hecho, buscar en la carpeta "Pixel Adventure" la subcarpeta "Fruits".
28. Ya ubicada la carpeta Fruits, dar click, donde vamos a ver todas las frutas que posee nuestra plantilla de videojuego.
29. Elegimos cualquier fruta y la arrastramos a nuestra escena.
30. Donde se va a repetir el paso 18 y 19, obviamente poniendo otro nombre a la animación de la fruta.
31. Ya que la fruta está en la escena, puedes repetir el paso 22 y 23 (solo es para ver la animación de tu rana y fruta)
32. Para hacer el mapa se va a buscar la parte derecha (donde se encuentran todos los objetos que hemos puesto) en una parte en blanco dar click derecho seleccionar "2D object" y se va a generar una sub columna donde vamos a seleccionar "Tilemap".
33. Una vez hecho esto se añadió un objeto llamado "Grid", que contiene el tilemap.
34. Seleccionamos el objeto grid.
35. (Siempre que seleccionemos un objeto en la parte de la derecha va a aparecer la personalización), estando en las opciones del objeto buscamos la que dice "Cell Size"

36. Una vez ubicada "Cell Size", cambiamos el valor de "x" y "y" por 0.16.
37. Vamos a crear un "Palette" para dibujar nuestro mapa.
38. Vamos a buscar en nuestra carpeta "Pixel Adventure" la subcarpeta "Terrain" y la seleccionamos.
39. Van a aparecer dos tipos de terrenos, vamos a elegir la que dice "terrain sliced" (solo ubicarla).
40. Para hacer uso de esta carpeta vamos a dar click al botón "window" y seleccionar 2D, aparecerá una sub columna en la que le daremos click a "Tile Palette".
41. Aparecerá una ventana.
42. Estando en la ventana nueva, seleccionar "Create New Palette" y agregarle un nombre a tu Pallete después dar click a create.
43. Aparecerá el explorador de archivos donde vamos a guardar nuestro palette.
44. Una vez hecho esto, vamos a arrastrar el terrain sliced en nuestra ventana del palette.
45. Se va a abrir el explorador de archivos donde vamos a guardar esto nuevamente.
46. Hecho esto, va a cargar todas las texturas de nuestro terreno para empezar a hacer el mapa.
47. Después de haber cargado nuestras texturas nos aparecen todos los terrenos que tenemos disponibles en nuestro palette.
48. Vamos a ajustar nuestra escena, seleccionando el objeto "main camera" buscar la opción "size" y cambiarla por 1.44.
49. Una vez ajustada ya podemos empezar el diseño del mapa con la ventana Palette seleccionando un un cuadrado en el paquete y plasmandolo en nuestra escena.
50. Diseña tu mismo tu mapa.
51. Para agregar un fondo se va a buscar la subcarpeta "Background" y la seccionamos.
52. Elegir cualquier color que tenga la subcarpeta y arrastrarla en la escena.
53. Se observa el el objeto se pone por delante del mapa.
54. En las opciones del objeto buscar la opción "Order in layer" cambiarlo por -1
55. Como el cuadrado es muy chico vamos a la opción Scale del objeto y cambiamos "x" y "y" por 2.

56. Ajustamos el rectángulo en una esquina de nuestra escena.
57. Hecho esto, hacemos (ctrl+d) para copiar el número de rectángulos hasta llenar todo nuestro background o fondo.
58. Creamos un objeto para almacenar los backgrounds, dando click derecho y seleccionar "Create empty", cambiar el nombre a background.
59. Para guardarlos en background solo basta con arrastrar los objeto hacia "Background"
60. Oprime el botón "Play", se observa que el personaje no cae, es decir no tiene físicas.
61. El subobjeto de "grid" que es "tilemap", se selecciona y se baja hasta la parte que dice "Addcomponent", se selecciona.
62. Aparecerá un buscador donde se va a buscar "Tilemap Collider" y lo agregamos.
63. Ahora se selecciona nuestro personaje que tiene como nombre "idle" (es recomendable cambiarle ese nombre para evitar confusiones), de las opciones de los objetos bajamos todo y se selecciona add component.
64. Al seleccionar Addcomponent, se va a buscar Box Collider 2D.
65. Se ajusta el box collider del personaje presionando el símbolo del collider que añadió para que se vea más realista el lugar donde está parado el personaje.
66. Listo, ya se tiene un mapa y las características físicas del personaje.

Diagrama de flujo (Movimiento del personaje)

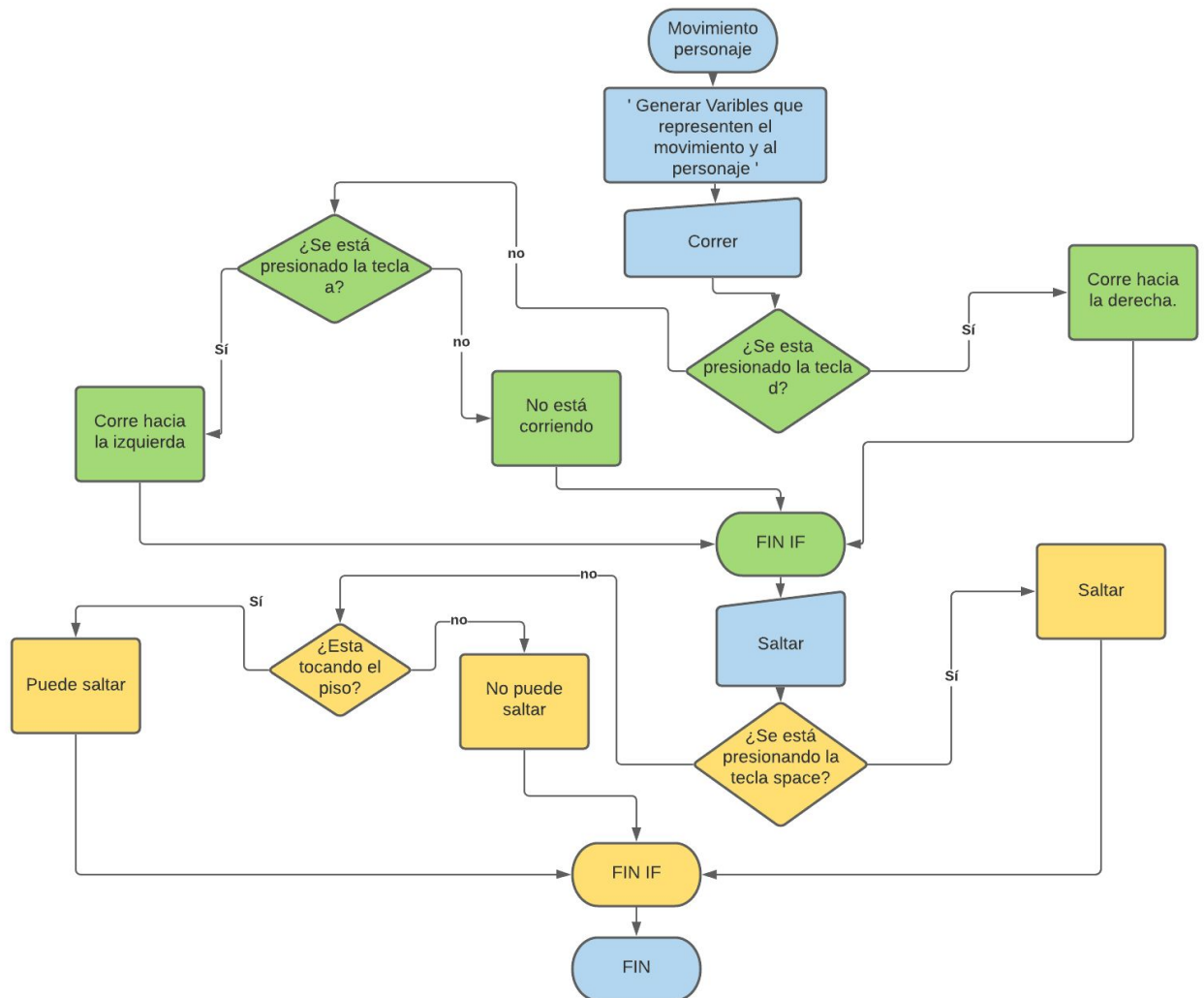


Diagrama de flujo (interacción con enemigos (spikes)).

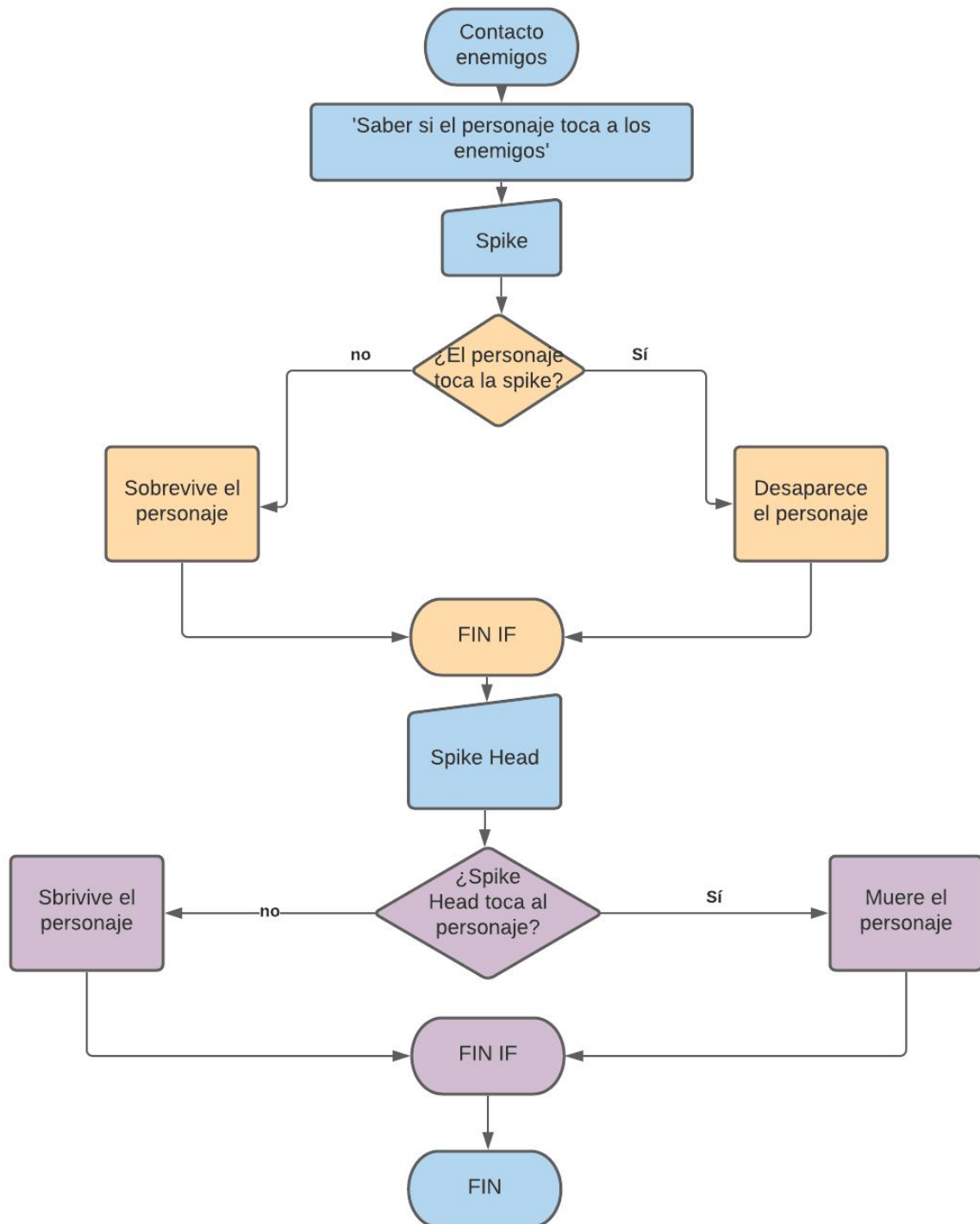
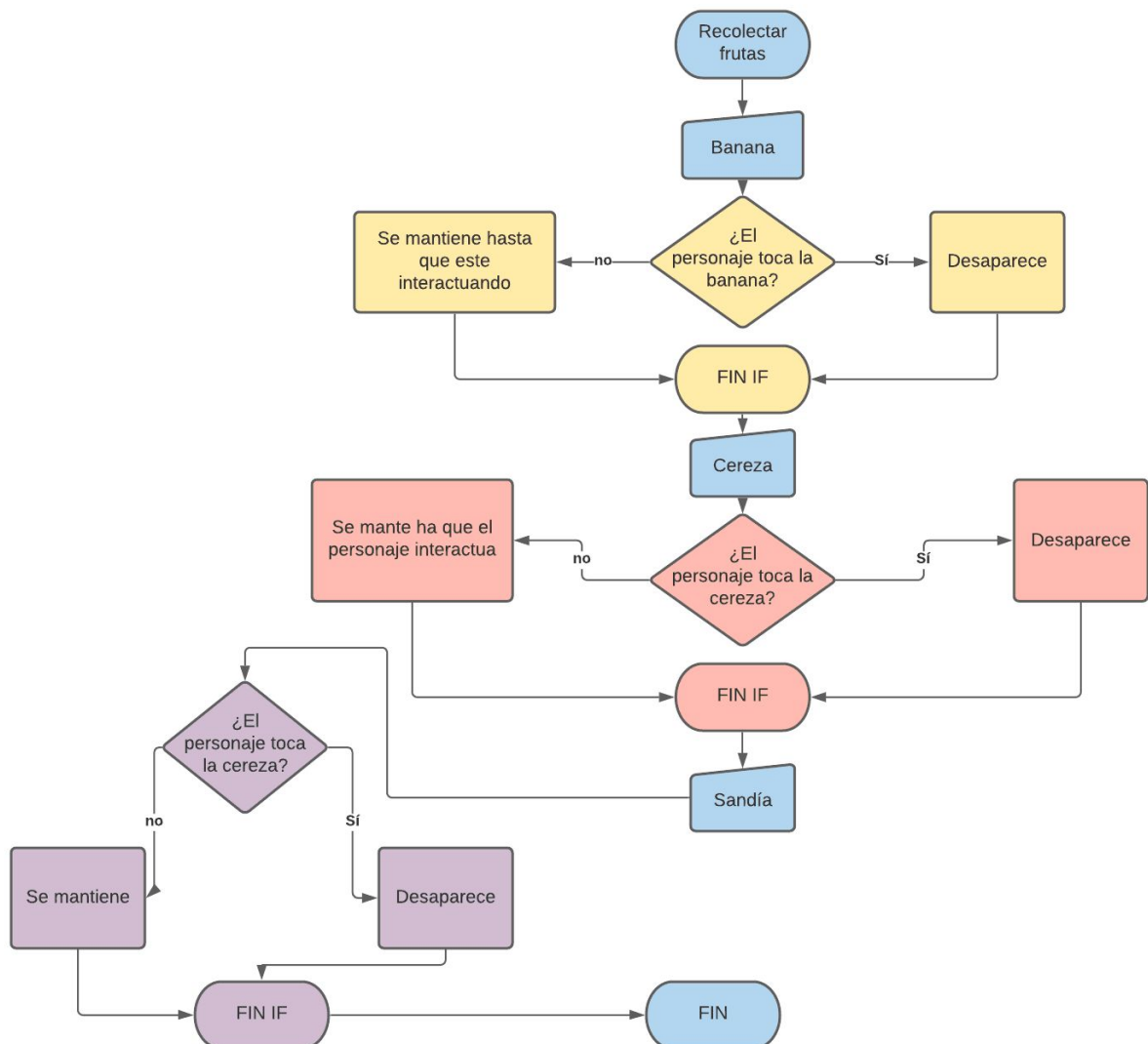


Diagrama de flujo (interacción con frutas).



Pseudocódigo (Movimiento player).

Programa: Move Player.

INICIO

runSpeed = 2 :PUBLIC FLOAT

jumpSpeed = 3: PUBLIC FLOAT

rb2D : RIGIDBODY

SI INPUT.GETKEY ("d")

ANIMATOR.SETBOOL("RUN",TRUE)

FIN SI

```

SI MAS INPUT.GETKEY("a")
ANIMATOR.SETBOOL("run",TRUE)
FIN SI MAS
ELSE
ANIMATOR.SETBOOL("run",FALSE)
FIN ELSE
SI INPUT.GETKEY("space") && CheckGround.isGrounded
rb2D.VELOCITY= NEW VECTOR2(rb2D.VELOCITY.x, jumpSpeed)
FIN SI
SI CheckGround.isGrounded==FALSE
ANIMATOR.SETBOOL("jump", TRUE)
ANIMATOR.SETBOOL("run", False)
FIN SI
SI CheckGround.isGrounded==TRUE
ANIMATOR.SETBOOL("jump",FALSE)
FIN SI

```

Las variables jump y run no se introducen ya que no están definidos dentro del motor físico de unity.

Programa: EnemySpike

```

INICIO
PLAYER TAG ("Player")
SI COLLISION.TRANSFORM.CompareTag("Player")
DEBUG.LOG("Player Damage")
DESTROY(COLLISION.gameObject)
FIN SI
SI

```

Estás variables al igual que el otro pseudocódigo pasado se encuentran ya definidas dentro del motor físico de Unity.

Programa: FruitCollected

```

INICIO

```

PLAYER TAG ("Player")

SI COLLISION.TRANSFORM.CompareTag("Player")

GETCOMPONENT SPRITERENDERER.ENABLED = FALSE

gameObject.TRANSFORM.GetChild 0. gameObject.SetActive TRUE

DESTROY (gameObject, 0.5f)

FIN SI

FIN

Son muy parecidos los últimos dos pseudocódigos ya que ambos tienen muy parecidas funcionalidades solo que uno desaparece y el otro te elimina; pero los comandos utilizados en C# con comandos de unity son muy complejos para hacerlos en pseudocódigo.

Código fuente.

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  [Script de Unity | 0 referencias]
6  public class MovePlayer : MonoBehaviour
7  {
8      public float runSpeed = 2;
9      public float jumpSpeed = 3;
10
11      Rigidbody2D rb2D;
12
13
14      public bool betterJump = false;
15
16      public float fallMultiplier = 0.5f;
17
18      public float lowJumpMultiplier = 1f;
19
20      public SpriteRenderer spriteRenderer;
21
22      public Animator animator;
23
24
25  [Mensaje de Unity | 0 referencias]
26  void Start()
27  {
28      rb2D = GetComponent<Rigidbody2D>();
29  }
30
31  [Mensaje de Unity | 0 referencias]
32  void FixedUpdate()
33  {
34      if(Input.GetKey("d") || Input.GetKey("right"))
35      {
36          // ...
37      }
38  }
```

```

32     if(Input.GetKey("d") || Input.GetKey("right"))
33     {
34         rb2D.velocity = new Vector2(runSpeed, rb2D.velocity.y);
35         spriteRenderer.flipX = false;
36         animator.SetBool("run", true);
37     }
38
39
40     else if (Input.GetKey("a") || Input.GetKey("left"))
41     {
42         rb2D.velocity = new Vector2(-runSpeed, rb2D.velocity.y);
43         spriteRenderer.flipX = true;
44         animator.SetBool("run", true);
45     }
46     else
47     {
48         rb2D.velocity = new Vector2(0, rb2D.velocity.y);
49         animator.SetBool("run", false);
50     }
51     if (Input.GetKey("space") && CheckGround.isGrounded)
52     {
53         rb2D.velocity = new Vector2(rb2D.velocity.x, jumpSpeed);
54     }
55
56     if(CheckGround.isGrounded==false)
57     {
58         animator.SetBool("Jump", true);
59         animator.SetBool("run", false);
60     }
61     if (CheckGround.isGrounded==true)
62     {
63         animator.SetBool("Jump", false);
64     }

```

```

60     }
61     if (CheckGround.isGrounded==true)
62     {
63         animator.SetBool("Jump", false);
64     }
65
66     if (betterJump)
67     {
68         if (rb2D.velocity.y<0)
69         {
70             rb2D.velocity += Vector2.up * Physics.gravity.y * (fallMultiplier) * Time.deltaTime;
71         }
72
73         if (rb2D.velocity.y>0 && !Input.GetKey("space"))
74         {
75             rb2D.velocity += Vector2.up * Physics.gravity.y * (lowJumpMultiplier) * Time.deltaTime;
76         }
77     }
78
79 }
80

```

Este es el "código fuente" la cual se enfoca en el movimiento del personaje y la físicas de este. En este código se declaran varias variables con public float ya que esta contiene valores que pueden cambiar constantemente dentro de la ejecución del programa y otras variables especiales que se utilizan en unity, por ejemplo para definir el mismo personaje y la otra para la animación de este todo esto enfocado en el eje "x" y "y" ya que se está programando un videojuego en segunda dimensión.

En este código se encuentran otras variables que no se encuentran definidas dentro del código, porque ya lo están en el motor Unity. Esto debido a que exista cierta conexión entre el programador y el motor físico.

En el código lo único que se encuentra de estructuras de repetición es la condición IF y IF-ELSE, donde definimos las teclas con la que va a ser controlado el personaje, además dentro de las condiciones se va a encontrar animator que este es el encargado de dar la animación dentro del juego y el velocity es la que permite esa movilidad, la parte de correr suele programarse de manera menos complicada.

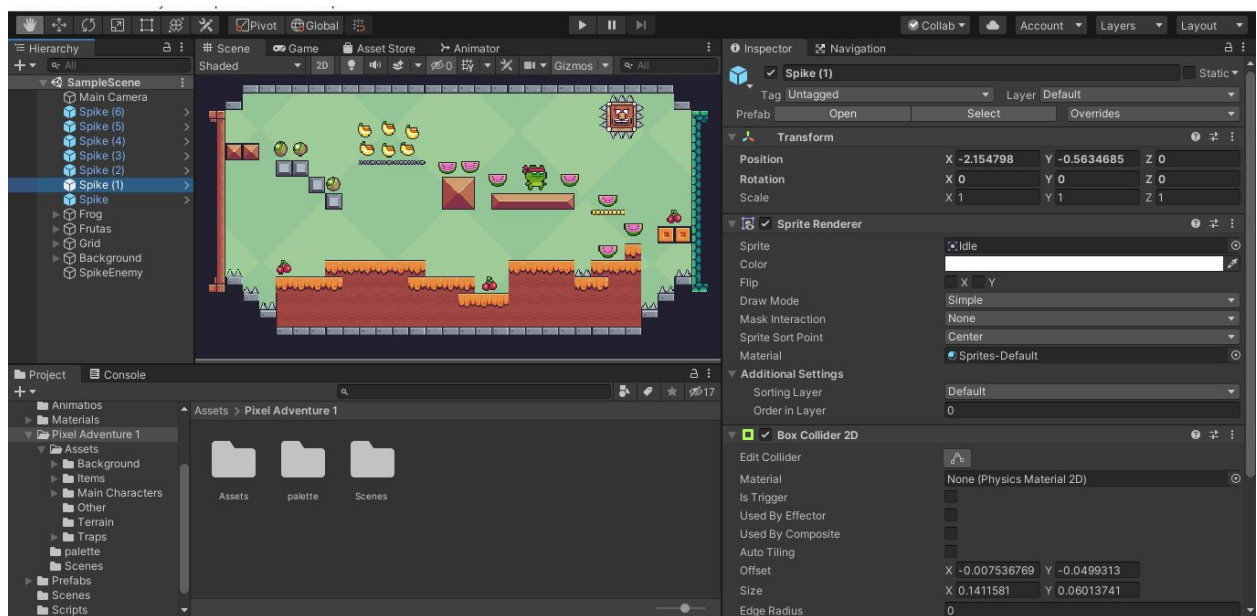
Para el salto que se encuentra dentro del código, se hizo más un poco más complicado ya que se programó de tal forma que al caer se vea y se sienta más natural, todo esto dentro de condiciones IF, además la componente Time y GRAVITY nos permite tener la peculiaridad de que al momento de mantener la tecla espacio el personaje va a permanecer más en el aire y si solo das un pequeño golpe al espacio salta poquito.

Una vez compilado el código se puede ver dentro del motor de Unity el movimiento que tiene el personaje.

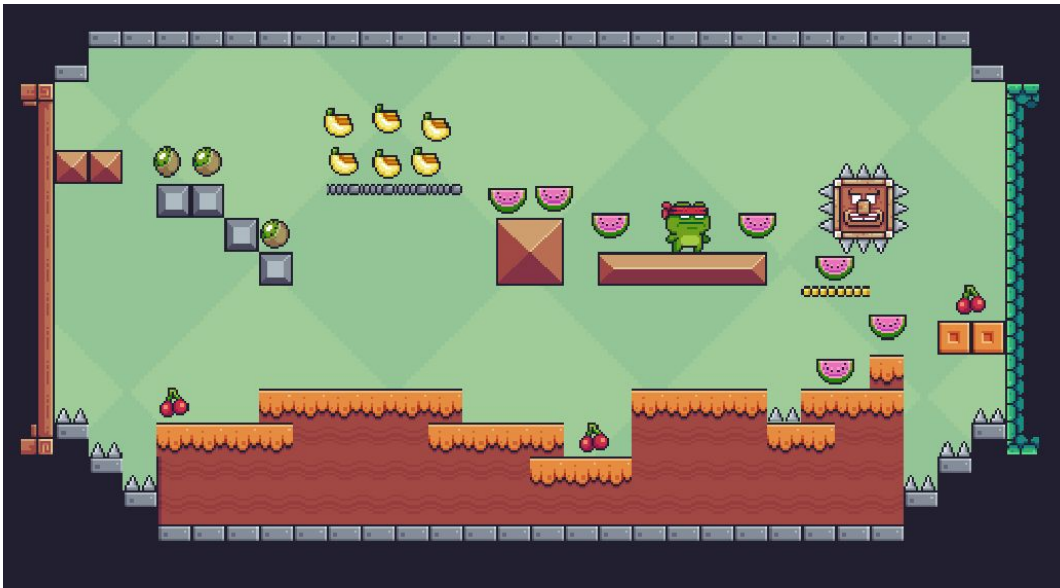
Cabe de aclarar que este videojuego fue basado en una serie de videos (curso) sobre la programación de videojuegos en 2D.

Capturas sobre el funcionamiento del videojuego.

Dentro del motor físico .



Juego en vista maximizada.



En esta captura se encuentra el personaje y los enemigos que son los pequeños pinchito y la cabeza cuadrada gigante que esta tiene movimiento y le genera daño al personaje, además las frutas interactúan con el personaje junto con una animación.

Captura estando en movimiento.



Como se puede observar ya faltan frutas, además el personaje está en otra posición donde se puede decir que está en movimiento hasta que se vea el movimiento.

Captura al momento de que muere el personaje.



Este desaparece y para volver a jugar hace falta volver a iniciar el videojuego ya que el juego así lo requiere.

Tabla comparativa de recursos informáticos [Hardware y Software] (Octubre 2020 vs Enero 2021)

Octubre 2020	Enero 2021
Planee utilizar una laptop o PC, mouse, teclado y un pad de dibujo.	Para desarrollar utilicé todos los recursos planeados excepto el pad de dibujo, ya que utilice una plantilla elaborada.
Programe el uso de un curso de diseño y animación digital.	Para el proyecto no tomé ningún curso de animación y diseño digital, como había comentado antes hice uso de una plantilla.
Planifique el uso de asesoría con personas experimentadas al tema de programación de videojuegos.	En realidad no conocí a alguna persona que haya sido experto en el tema de los videojuegos.
Organice el uso de softwares que son: <ul style="list-style-type: none"> • Illustrator • Unreal Engine • Unity 	En verdad solo hice uso del motor Unity, ya que no se puede hacer un híbrido entre dos motores físicos, y para mayor comodidad y evitar fallos de mi computadora se eligió Unity
Uso de plantillas de animación como alternativa de los programas de diseño.	En verdad solo se utilizó una plantilla que contenía todo lo necesario para la creación de mapas y personajes con animaciones.

Tabla comparativa de costos asociados al proyecto (Octubre 2020 vs Enero 2021)

Octubre 2020	Enero 2021
Planee que el proyecto no sería elaborado por un tercero, ya que es más importante adquirir el conocimiento; pero sí pagaría para asesorías sobre programación y diseño.	Mantuve lo dicho en esos meses el proyecto lo desarrollé yo, puesto que para mí es de suma importancia la experiencia que uno tiene al programar el videojuego y lo tanto que lo conoces; pero esto yo no lo hice solo, sino que me apoye en una serie de vídeos de youtube para la elaboración de este.
Comente que si yo fuera el que programara el mismo proyecto de otra persona no cobraría nada debido a que prefería el conocimiento que el pago, puesto que en ese tiempo era más importante para mi la experiencia.	Actualmente opino lo contrario, si yo cobrara por algún proyecto cobraría como 5000 pesos para todo el proyecto ya que es mucho trabajo hacerlo y desarrollarlo, además el tiempo que se lleva haciendo no es poco.
Pensé en invertir como 1000 pesos para cursos y licencias de software, todo esto para que vaya lo mejor posible en mi proyecto.	En estos momentos no he gastado ni un solo peso, ya que en ese tiempo no conocía que había versiones gratuitas para la licencia de software, además no tome ningún curso si que todo lo aprendí por medio de youtube.
Aunque en la tabla no dice tiempo, lo decidí agregar por que me parece de suma importancia, para esto calcule un año para desarrollar el proyecto, que me parece un tiempo razonable.	Actualmente lo sigo viendo como un periodo excelente debido a la dificultad del desarrollo del juego, ya que no poseo los conocimientos suficiente para desarrollarlo en menos tiempo.

Comparación de diagramas de Gantt (Octubre 2020 vs Enero 2021).

Diagrama de gantt (Octubre 2020)

Nombre de la actividad	Fecha de inicio.	Duración en días.	Fecha fin.
Diseño diseño del personaje	26-oct	31	26-nov
Diseño de mapas.	27-nov	30	27-dic
Diseno del HUD.	28-dic	31	28-ene
Progrmación de movimientos del personaje.	29-ene	31	01-mar
Programación de los mapas del juego.	02-mar	31	02-abr
Insertar dentro le los mapas obstaculos.	03-abr	30	03-may
Integrar el personaje con sus movimientos en los mapas.	04-may	31	04-jun
Realizar divisiones de nivel en los mapas (facil, medio,dificil)	05-jun	30	05-jul
Programar el HUD del videojuego	06-jul	31	06-ago
Integrar dentro del HUD los niveles de cada mapa.	07-ago	31	07-sep
Probar el juego ya terminado.	08-sep	30	08-oct

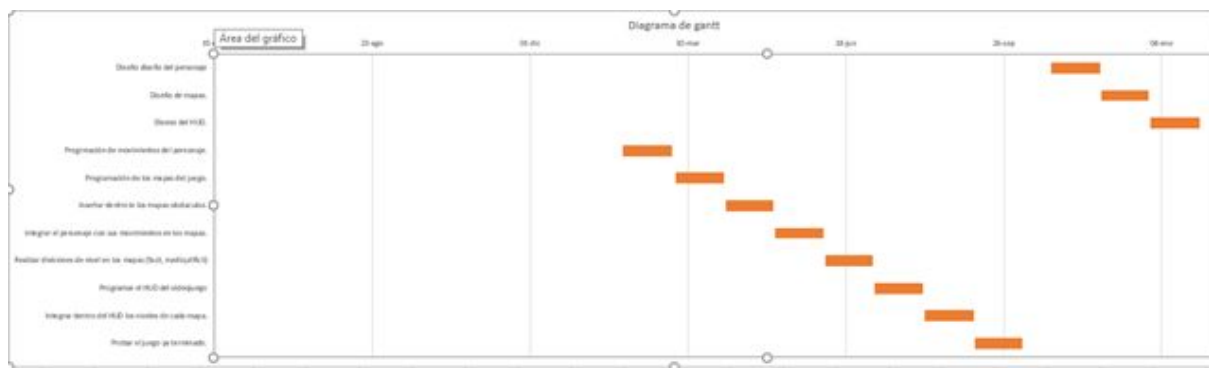
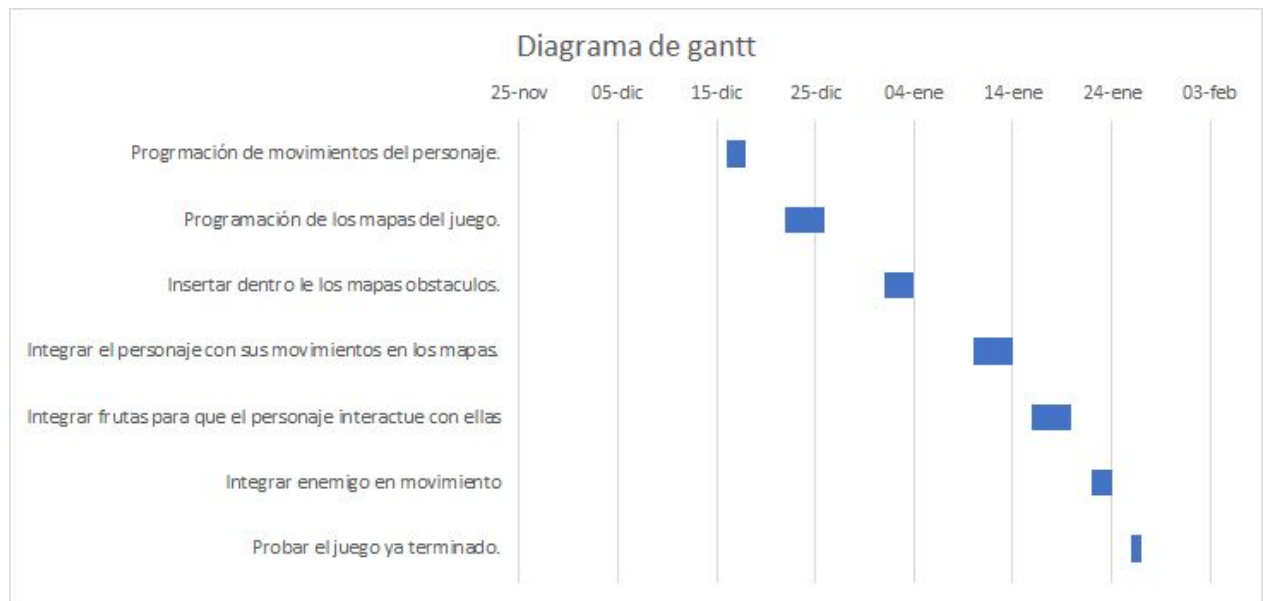


Diagrama de Gantt (Enero 2021)

Nombre de la actividad	Fecha de inicio.	Duración en días.	Fecha fin.
Progrmación de movimientos del personaje.	16-dic	2	18-dic
Programación de los mapas del juego.	22-dic	4	28-dic
Insertar dentro le los mapas obstaculos.	01-ene	3	04-ene
Integrar el personaje con sus movimientos en los mapas.	10-ene	4	14-ene
Integrar frutas para que el personaje interactue con ellas	16-ene	4	20-ene
Integrar enemigo en movimiento	22-ene	2	24-ene
Probar el juego ya terminado.	26-ene	1	27-ene



Se observa una reducción inmensa producción y una mejora en el orden de realización del proyecto, donde a mi parecer el diagrama de gantt de Enero se me hace más estructurado, es decir, tiene más concordancia la gráfica actualizada, debido a la falta de conocimiento sobre el proyecto, ya que en esos momentos todavía no empezaba mi proyecto así que no tenía ideas cien por ciento claras sobre las actividades que se realizan en la programación de videojuegos.

Link del canal de youtube, junto con el link del video (por si falla).

- **Link Video.**

<https://youtu.be/mMLXWgjjmw>

- **Link Canal de YOUTUBE**

<https://www.youtube.com/channel/UCeJ1XIASNtmbIJfxxUe6YA>

- **Repositorio de GitHub del Proyecto final.**

<https://github.com/androsario5/ProyectoFinal>

Conclusiones.

En mi proyecto se encuentra directamente relacionado con GNU ya que el sistema operativo tipo unix es uno de los medios en el que se puede trabajar en el motor físico de Unity, está al poseer GNU está completamente apoyado por la FSF que es

una organización que apoya el software, es decir, están totalmente en desacuerdo las restricciones que ponen los softwares privados, por eso el concepto de software libre. En cuanto a la IEEE está aún más conectado ya que posiblemente una gran parte de la teoría dentro de este software puede encontrarse en la información que te otorga este instituto, sobretodo porque promueven la creatividad que una parte muy importante de trabajar en unity, además todo esto es parte de los avances en las ciencias computacionales ya que está en constante evolución.

En cuanto en lo personal el proyecto me pareció un medio interesante ya que activa la creatividad eligiendo lo que tu quieras hacer, debido a esto siempre me dio curiosidad crear un videojuego. Al principio del proyecto veía muy fácil el desarrollo de un videojuego; pero no, todo esto se somete a códigos complejos en un lenguaje que aún no conozco al cien por ciento, lo que me gustó es que hacer este tipo de proyecto te da una vista panorámica de lo que se puede hacer programando con apoyo de algún software, lo que no me gustó es que al momento de fallar la escritura de una línea del código puede empezar a fallar todo el software, por lo que tuve que hacer prácticamente el proyecto dos veces; pero a fin de cuentas son aprendizajes que va hacia uno mismo. Por último voy a continuar el desarrollo de otros niveles, funcionalidades ya que me sirvió de motivación para seguir programando. :D

Referencias.

Bibliografía Digital.

Unity. (s. f.). *Unity Asset Store - The Best Assets for Game Making*. Unity Asset Store. Recuperado 16 de diciembre de 2020, de <https://assetstore.unity.com/>

Soluciones MyL. (s. f.). *Curso Completo*. Recuperado 20 de diciembre de 2020, de http://solucionesmyl.com/cursos/lenguaje_c/home.php

Videos de youtube.

Video 1

Nombre del video: Juego de Plataformas 2D/Unity Tutorial

2021/1-Capitulo/Escenario

Nombre del Canal: Luis canary

Fecha de Publicación: 14 de abril 2020

Fecha de consulta: 22 de diciembre del 2020

Enlace del video: <https://www.youtube.com/watch?v=-m7ZaHhkDAc&t=1s>

Video 2

Nombre del video:Juego de Plataformas 2D/Movimiento y salto /unity Tutorial 2021

/1 -Capitulo /Escenario

Nombre del Canal: LuisCanary

Fecha de Publicación: 17 de abril del 2020

Fecha de consulta: 16 de diciembre del 2020

Enlace del video:https://www.youtube.com/watch?v=5o9b_VdjfDU&t=134s

Video 3

Nombre del video:Juego de Plataformas 2D/Recoger frutas /unity Tutorial 2021 /3

-Capitulo /Programación de videojuegos

Nombre del Canal: LuisCanary

Fecha de Publicación: 22 de abril del 2020

Fecha de consulta: 16 de enero del 2021

Enlace del video: <https://www.youtube.com/watch?v=vIXF6XhAje8&t=24s>

Video 4

Nombre del video: Juego de Plataformas 2D/Animaciones 2D /unity Tutorial 2021 /4 -Capitulo /Programación de videojuegos

Nombre del Canal: LuisCanary

Fecha de Publicación: 24 abril del 2020

Fecha de consulta: 18 de enero del 2021

Enlace del video: <https://www.youtube.com/watch?v=Z25BldJbDZU&t=16s>

Video 5

Nombre del video: Juego de Plataformas 2D/Enemigos /unity Tutorial 2021 /5 -Capitulo /Programación de videojuegos

Nombre del Canal: LuisCanary

Fecha de Publicación: 27 de abril del 2020

Fecha de consulta: 22 de enero del 2020

Enlace del video: <https://www.youtube.com/watch?v=7NCdAJ0g7bs&t=42s>