Guion de Persefone

Buenos días visitantes, nuestro proyecto para la jornada tecnológica V3.0 de 2023 dará a conocer una manera de realizar un simulador didáctico básico o mejor dicho un videojuego pero antes de empezar hablar del tema central vamos a aclarar unos conceptos básicos para el mejor entendimiento del proyecto.

¿Que es un simulador?

Esta jornada tiene como tema central, los simuladores pero varias personas desconocen este termino.

Un simulador es un aparato, por lo general informático, que permite la reproducción de un sistema. Los simuladores reproducen sensaciones y experiencias que en la realidad pueden llegar a suceder.

Un simulador pretende reproducir tanto las sensaciones físicas (velocidad, aceleración, percepción del entorno) como el comportamiento de los equipos de la máquina que se pretende simular.

En resumen, emitan la realidad pero esta definición es algo ambigua porque un simulador no siempre esta anclada a la imitación de la realidad sino que también la falsean para crear una realidad nuevo, por ejemplo:

- Mundos fantásticos.
- Mundos ciencia ficción.
- Mundos pos-apocalípticos.

Los videojuegos son un tipo de simulador didácticos, ya que estos le dan interactividad al usuario de su entorno.

Ya con este termino claro, ya podemos avanzar.

Etapas para la creación de un videojuego

Como todo en la vida tiene unos protocolos a seguir para que las cosas vaya bien y crear un videojuego no es la excepción, las etapas son:

- visión
- estrategia
- producto

Visión

Esta es la etapa mas importante en la creación de un videojuego porque de este punto sera la representación del producto final y por eso muy importante porque esto puede decidir por un 50%, si un juego es bueno o malo.

En esta etapa se piensan las características del videojuego, como:

- mecánicas

  las mecánicas son todos los objetos o cosas que utiliza o interactúan con el jugador en el uso del videojuego, por ejemplo:

  un salto, un proyectil, enemigos, plataformas, una cámara, los niveles, etc..

- Genero

  Los videojuegos se pueden clasificar en géneros atendiendo a factores como el sistema de juego, el tipo de interactividad con el jugador, sus objetivos, etc. La evolución de los videojuegos desde sus comienzos ha dado lugar a una variedad creciente y cambiante de géneros, muchas veces en relación con lo que los avances en la tecnología han ido haciendo posible. Entre los géneros de videojuegos más populares están los de acción, estrategia, rol, aventura, rompecabezas, simulación, deportes o carreras, cada uno de ellos con varios subgéneros. Por otro lado, hoy en día son habituales los videojuegos que toman elementos de más de un género, lo que ha dado lugar a géneros mixtos (por ejemplo rol-acción, aventura-acción, etc.).

- demografía

  es el publico que esta dirigido el videojuego puede ser para niños, adolescentes, hombres, mujeres, etc.

  la demografía esta ligada al genero del videojuego ya que estos son mas o menos populares dependiendo de la demografía de los consumidores.

- Arte

  este apartado nos referimos principalmente si es un juego2D o un juego3D, ya que dependiendo

  de la tecnología a usar lograremos aplicar un estilo artístico.

  Este apartado afecta a las mecánicas ya que el arte puede hacer imposible la recreación y aplicaciones de algunas mecánicas.

- plataformas de publicación

  En este apartado nos referimos,Los distintos tipos de dispositivo en los que se ejecutan los videojuegos se conocen como plataformas.

  Es muy importante diferenciar las plataformas que existen ya que estas nos entregan como desarrolladores ventajas y desventajas al momento de la creación que podemos aprovechar, Los cuatro tipos de plataformas más populares son el PC, las videoconsolas, los dispositivos portátiles y las máquinas de árcade.

  Este apartado es el primero a pensar ya que esta define el arte y las mecánicas porque un juego para consolas es casi imposible de portar a un dispositivo portátil.

Estrategia

Si ya tenemos claro nuestra visión podemos pasar a la fase de estrategia, en donde todas las ideas que realizamos en la etapa de visión las aplicaremos para analizrlas si son viables de estar en el producto final.

La primer paso a realizar en esta etapa es la elección de un motor grafico pero primero, ¿que es un motor grafico?

Un motor de videojuego (en inglés game engine), es un entorno de desarrollo que proporciona herramientas para la creación de videojuegos.

Su función principal es dotar al videojuego de un motor para renderizar gráficos 2D y 3D, un motor físico que simule las leyes de la física y detección de colisiones, y herramientas para poder crear las animaciones, scripts, sonidos, inteligencia artificial, redes, gestión de memoria, y demás sistemas del videojuego.

Esta elección debe ser sabia porque no todos los game engine tiene las mismas funcionalidades y prestaciones para la creación de videojuegos porque dependiendo de tu visión uno sera mejor o peor que otro para la aplicación de tu visión.

Los los Game Engines mas populares son:

- Unreal Engine
- Unity
- Godot engine

para el material didáctico de este proyecto usamos Godot porque tiene las prestaciones que nosotros estábamos buscado:

1. poco uso de memoria ram y CPU.
2. Programación sencilla.

¿creando un jugador?

Para crear un jugador en godot se hace mediante nodos.

Los nodos son objetos que tiene características dependiendo de su tipo, que puede ser nodos3d, nodos3D o interfaz de usuario.

1. Primero colocamos un nodo de tipo kimeebody2D.
2. Agregamos un nodo hijo de tipo Sprite.
3. Agregamos un colisionShape2D.
4. Agregamos un AnimatePlayer.

Después de acomodar estos nodos ahora hay que configurar sus propiedades.

Sprite

Este le agregamos un archivo .png que queramos que el jugador controle.

ColisionShape

Damos click en el nodo y en la primera opción le damos click para crear la forma de colisión que tendrá nuestro jugador.

AnimatePlayer

Le damos click al nodo animationPlayer y en la parte inferior, saldrá un nuevo menú que a continuación el daremos click en el apartado de animation y luego en new para crear las animaciones que tendra el jugador.

Despues de configurar las propiedades de los nodos, debemos programar el movimiento de jugador para empezar a testear.

Codigo

```
extends KinematicBody2D


# Declare member variables here. Examples:
# var a = 2
# var b = "text"


# Called when the node enters the scene tree for the first time.
func _ready():

	pass # Replace with function body.



export (int) var run_speed = 100
export (int) var jump_speed = -400
export (int) var gravity = 1200

var velocity = Vector2()
var jumping = false
```

```
#esta funcion sirve para cuando el jugador caiga al vacio
func get_position_vacio():
	#calcula la posion en el eje y
	if position.y >= 780:
		print_debug("esta en el vacio")
		queue_free()
		pass
		#calcula la posion en el eje x
	if position.x >= 1024:
		print_debug("esta en el vacio")
		queue_free()
		pass

	pass

#esta funcion maneja los inout
func get_input():
	velocity.x = 0
	var right = Input.is_action_pressed("derecha")
	var left = Input.is_action_pressed("izquierda")
	var jump = Input.is_action_just_pressed("jump")

	if jump and is_on_floor():
		jumping = true
		velocity.y = jump_speed
	if right:
		velocity.x += run_speed
	if left:
		#$flip_h
		velocity.x -= run_speed
	else:
		$Sprite/AnimationPlayer.play("Quito")

func _physics_process(delta):
	get_position_vacio()
	get_input()
	velocity.y += gravity * delta # esta linea hace que al sprite le afecte la gravedad

	# es ray casting, es decir no deja hacer doble ssalto o un flappy beard
	if jumping and is_on_floor():
		jumping = false
	velocity = move_and_slide(velocity, Vector2(0, -1))
```

creación de un nivel

Ahora que creamos el jugador, ahora debemos crear el ambiente en donde el jugador va interatuar.

Creamos una nueva escena y después agregamos un nodo padre de tipo 2D.

Ahora agregamos un tilemap y lo configuramos.