UNITY

Las palabras en verde son variables creadas por mí, por lo que hay que inicializarlas

Las palabras en rojo son funciones

-Moviento WASD

if (Input.GetKey(KeyCode.D))(Movimiento derecha)

{

rb2d.AddForce(Vector2.right \* speed);

transform.localScale = new Vector3(escalaInicial, transform.localScale.y, transform.localScale.z);

}

if (Input.GetKey(KeyCode.A)) (movimientoizquierda)

{

rb2d.AddForce(Vector2.right \* -speed);

transform.localScale = new Vector3(-escalaInicial, transform.localScale.y, transform.localScale.z);

}

if (Input.GetKeyDown(KeyCode.W))

{

rb2d.velocity = new Vector2(rb2d.velocity.x, 0);

rb2d.AddForce(Vector2.up \* jumpPower, ForceMode2D.Impulse);

jump = false;

}

-Parpadeo:

void Parpadeo() (sr es el Sprite renderer, hay que inicializarlo en el start)

{

timer -= Time.deltaTime;

timerParpadeo -= Time.deltaTime;

if (timer < 0)

{

timer = 0.1f;

sr.enabled = !sr.enabled;

}

if (timerParpadeo <= 0)

{

sr.enabled = true;

}

}

-Bala:

void Update()

{

rb2d.velocity = transform.right\*vel;

}

-Explosión de la bala al chocar:

void OnTriggerEnter2D(Collider2D col)

{

switch (col.gameObject.tag)

{

case "Enemigo":

{

GameObject clon;

clon = Instantiate(prefabExplosion);

clon.transform.position = transform.position;

Destroy(col.gameObject);

Destroy(gameObject);

}

break;

}

}

-Enemigo:

void Update()

{

rb2d.AddForce(Vector2.right \* speed);

float limitedSpeed = Mathf.Clamp(rb2d.velocity.x, -maxSpeed, maxSpeed);

rb2d.velocity = new Vector2(limitedSpeed, rb2d.velocity.y);

if (speed> 0)

{

transform.localScale = new Vector3(escalaInicial, transform.localScale.y, transform.localScale.z);

}

else if (speed < 0)

{

transform.localScale = new Vector3(-escalaInicial, transform.localScale.y, transform.localScale.z);

}

}

void OnTriggerEnter2D(Collider2D col)

{

if (col.gameObject.tag =="LimiteEnemigo")

{

speed = -speed;

rb2d.velocity = new Vector2(speed, rb2d.velocity.y);

}

}

-Menú de pausa:

void Update()

{

if (Input.GetKeyDown(KeyCode.Escape))

{

if (Time.timeScale == 0)

{

Time.timeScale = 1;

}

else if (Time.timeScale == 1)

{

Time.timeScale = 0;

}

}

}

-Palanca/placa para abrir puertas(yo lo hice con palanca, por eso uso el flip)

void OnCollisionEnter2D(Collision2D col)

{

if (sr.flipX == true)

{

sr.flipX = false;

}

else if (sr.flipX == false)

{

sr.flipX = true;

}

if (sr.flipX==true)

{

door.MarcarParaAbrir();

}

else

{

door.MarcarParaCerrar();

}

}

-Puerta:

void MoverPuerta()

{

if (abriendo)

{

if (giro < 90)

{

giro += 10 \* Time.deltaTime;

transform.rotation = Quaternion.Euler(0, giro, 0);

}

else

{

transform.rotation = Quaternion.Euler(0, 90, 0);

}

}

if (cerrando)

{

if (giro >= 0)

{

giro -= 10 \* Time.deltaTime;

transform.rotation = Quaternion.Euler(0, giro, 0);

}

else

{

transform.rotation = Quaternion.Euler(0, 0, 0);

}

}

}

public void MarcarParaAbrir()

{

abriendo = true;

cerrando = false;

}

public void MarcarParaCerrar()

{

abriendo = false;

cerrando = true;

}

-Torreta:

void Start()

{

InvokeRepeating("BalaTorre", 1f, cadencia);

}

void BalaTorre()

{

GameObject clon;

clon = Instantiate(prefabBalaTorre);

clon.transform.position = transform.position;

float giro = Random.Range(0f, 360f);

clon.transform.Rotate(0,0,giro);

}

-Balas de la torreta:

void Update()

{

rb2d.velocity = transform.right \* vel;

}

void OnTriggerEnter2D(Collider2D col)

{

switch (col.gameObject.tag)

{

case "Muerte": (tag para los limites de fuera del mapa)

{

Destroy(gameObject);

}

break;

case "Player": (tag para referirme a los jugadores)

{

if (player.vida <=1) (player aquí es el nombre del script del jugador)

{

GameObject clon;

clon = Instantiate(prefabExplosion);

clon.transform.position = transform.position;

Destroy(col.gameObject);

Destroy(gameObject);

Destroy(clon.gameObject, time);

}

}

break;

}

}