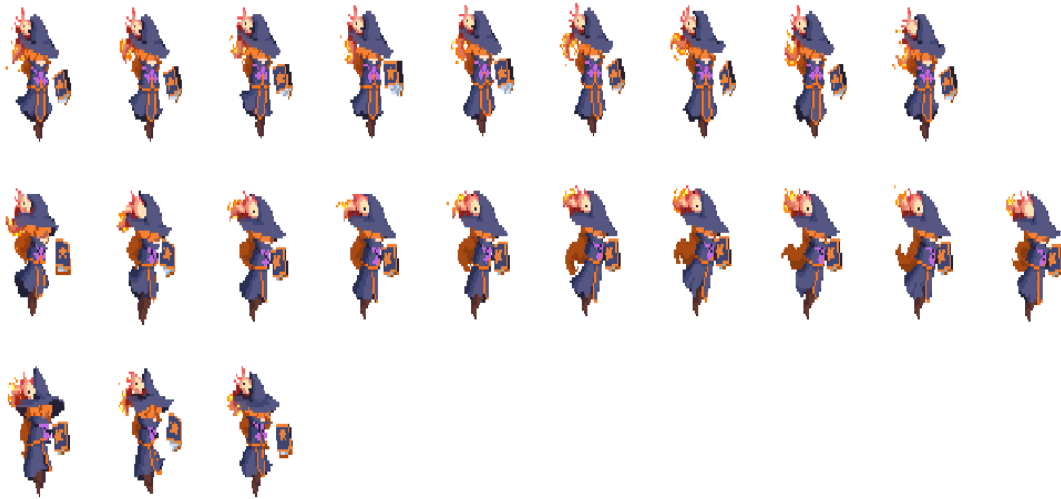


GUIA Sistema Animacion

ParteXML:

Paso 1:

Sprite de ejemplo:



Codigo

```
<scene>
<!-- <player x="1450" y="2500" texturePath="Assets/Textures/Entidades/Jugador/salamanda.png" coinfxpath="Assets/Audio/Fx/retro-video-game-coin-pickup-382
-->
<player texturePath="Assets/Textures/Entidades/Jugador/salamanda.png" Posx="150" Posy="570" Tsprite="23" Pwidth="1000" sprite_x="100" sprite_y="160"/>
<map path="Assets/Mapas/" pathTextures="Assets/Textures/Mapas/" name="Testing/TestingMap.tmx"/>
<item x="1200" y="2600" texturePath="Assets/Textures/Entidades/Jugador/salamandal.png" Tsprite="23" Pwidth="1000" sprite_x="100" sprite_y="160"/>
</scene>
```

texturePath: ruta de Sprite

Tsprite: cuanto sprite tiene(aqui como ejemplo tiene dos fila de 10(espacio de primera fila tambien se cuenta) y un fila de 3, pues $10 \times 2 + 3 = 23$)

Pwidth: ancho de sprite, aqui era 1000(de parte izquierda)



sprite_x: ancho de sprite(forma de calcular: anchoDeSprite(aquí es 1000) / cuandoSpriteHayEnUnaFila(aquí es 10) = 100)
 sprite_y: alto de sprite(forma de calcular: altoDeSprite(aquí es 480) / cuandoSpriteHayEnUnaColumna(aquí es 3) = 160)

Paso 2:

```
<!--Animation-->
<animations>
  <player>
    <!-- chibi 100-->
    <idleAnim start="0" end="9" speed="0.2f" loop="true"/>
    <runAnim start="11" end="23" speed="0.2f" loop="true"/>
  </player>

  <item>
    <idleAnim start="0" end="23" speed="0.2f" loop="true"/>
  </item>

</animations>
```

Crear etiqueta.nombre lo que quiera pero recomento como nombre de clase.

Dentro de etiqueta crear otro etiqueta para animación, recomento usar nombre de animación ej idle, run, ataque, etc.

Y dentro de este etiqueta rellena datos

start = este animación en que Sprite se comenza.

end = este animación en que Sprite se termina.

speed = la velocidad de este animacion

loop = pone true o false

Ej:

```
<nomClase>
  <nomdeanimacion start ="0" end ="10" speed = "0.1f" loop = "true"/>
  <idle start ="11" end ="20" speed = "0.2f" loop = "true"/>
  <ataque start ="21" end ="30" speed = "0.3" loop = "false"/>
</nomClase>
```

Parte de Clase

documento .h

```
//Animation Obligatorio
Animation* currentAnimation = nullptr;
Animation SPosition;
SDL_Rect* spritePositions;
SDL_Rect rect;
SDL_Texture* texture;
const char* texturePath;
int TSprite;
int SpriteX;
int SpriteY;
int Photowidth;

//Animation personalizar
Animation idleAnim;
```

Parte obligatorio, Hacer copia y pegar y ya

Parte de personalizar se puede cambiar nombre lo que quieres

documento .cpp

```
//personal, opcional
position.x = parameters.attribute("x").as_int();
position.y = parameters.attribute("y").as_int();

//obligatorio
texturePath = parameters.attribute("texturePath").as_string();
TSprite = parameters.attribute("Tsprite").as_int();
SpriteX = parameters.attribute("sprite_x").as_int();
SpriteY = parameters.attribute("sprite_y").as_int();
Photowidth = parameters.attribute("Pwidth").as_int();
spritePositions = SPosition.SpritesPos(TSprite, SpriteX, SpriteY, Photowidth);
//Animation
idleAnim.LoadAnim("item", "idleAnim", spritePositions);

//initilize textures
texture = app->tex->Load(texturePath);
return true;
```

Parte obligatorio, hacer copiar y pegar,(!!!dentro de attribute tiene que ser mismo como XML, aquí aviso por acaso)

Luego //Animacion,

Coger Animacion, aquí ejemplo se llama idleAnim, usar

“.LoadAnim(“nomClase”,”nomdeanimacion”,spritePositions);”

nomClase = el nombre de clase que ha creado por xml
nomdeanimacion = el nombre de animacion que ha creado por xml
spritePositions = siempre era este

```
<!--Animation-->
<animations>
  <player>
    <!-- chibi 100-->
    <idleAnim start="0" end="9" speed="0.2f" loop="true"/>
    <runAnim start="11" end="23" speed="0.2f" loop="true"/>
  </player>

  <item>
    <idleAnim start="0" end="23" speed="0.2f" loop="true"/>
  </item>

</animations>
```

Parte de Draw

```
bool Item::PostUpdate()
{
    if (currentAnimation == nullptr) { currentAnimation = &idleAnim; }
    rect = currentAnimation->GetCurrentFrame();
    app->render->DrawTexture(texture, 1400, 2300, 2, SDL_FLIP_NONE, &rect);
    currentAnimation->Update();

    return true;
}
```

Y aquí tenemos todos



Mas función sobre Animation.h

`nombreDeAnimacion.getNameAnimation()`

`.getNameAnimation()` Se puede obtener nombre de animacion, nombre que has creado en XML

ej:

```
if (testNombre.HasFinished() && testNombre.getNameAnimation() == "nombre
deAnimacion") {
    printf("Este Animacion esta terminado");
}
```

Si testNombre hay varios animacion, ej puede ser ataque1, ataque2 y ataque3 todo se usar uno, asi cuanto terminar, no sabemos quina, pero ahora si usamos `.getNameAnimacion()`, ya podemos saber, y hacer los cosa nos queremos.

Si deseas que la hoja de sprites se invierta, por ejemplo, comenzando desde el 10 y terminando en el 0, simplemente debes cambiar 'start' a '10' y 'end' a '0' en el XML. El sistema determinará si estás utilizando un orden ascendente o descendente.