Escuela Tecnica Nº35

Ingeniero Eduardo Latzina

DONKEY KONG

Definicion Tecnica

Profesor:
Carlos Lescano

Grupo:
Mateo Ricci
Ignacio Castillo
Gustavo Delgado

Indice

Definición de clases

Clase ObjetoMovil

La clase objeto movil es la clase madre de la cual heredaran sus atributos las clases Personaje y Barril.
Sus atributos son:

RectangleShape colisionador Es un cuadrado invisible para el jugador, el cual se utiliza para el calculo de las colisiones.
float gravity Se utiliza al momento de saltar, para sumarsela a la velocidad en el eje Y, asi haciendo descender al personaje.
float jumpSpeed Es el valor que se le asigna a la velocidad en el eje Y, como velocidad inicial del salto
bool saltar Se utiliza para verificar que el personaje pueda saltar, es decir, no este en el aire.
Vector2f velocity
int posX Se usa para inicializar la posición del colisionador en el eje X
int posY Se usa para inicializar la posición del colisionador en el eje Y, esta variable y la anterior, son para facilitar el testeo del juego mientras se codifica.
int minPiso Se utilliza para establecer el mínimo de altura en el que esta el colisionador/objeto
bool puedeEscalar
int piso Se usa para identificar en que piso esta el objeto.
bool control

Clase Personaje

Esta clase hereda atributos de la clase objetoMovil y a su vez tiene sus propios atributos, se encarga de modelar los personajes
Texture textura Se utiliza para cargar la imagen que se utilizara para luego asignársela al sprite, dependiendo de el tipo de personaje que sea, y su movimiento.
Sprite personaje Se usa para asignarle la textura y dibujarla en pantalla.
int lado Se utiliza para identificar para que lado esta apuntando el personaje (derecha o izquierda).
string tipo Se utiliza para identificar si el personaje es del tipo “Mario” o “Princesa”.

Personaje(String t):
Al instanciar un objeto de la clase Personaje, se le asigna a la textura, el archivo que contiene los sprites de mario y la princesa.
Y al atributo tipo se le asigna el valor de t, que se le paso al instanciarlo.

Si el tipo de personaje es princesa, se recorta la textura a las posiciones correspondientes para mostrar la princesa, y se setea la posicion de la princesa en la parte superior de la pantalla, junto a su colisionador.

Si el tipo de personaje es mario, se recorta la textura a las posiciones correspondientes para mostrar a mario, y se setea la posicion tanto del colisiionador como del sprite, en la parte inferior de la pantalla.
Se asignan los siguientes valores:

gravity=2;
jumpSpeed=11;
saltar= true;
posX=100;
posY=471;
puedeEscalar=NULL;
velocity.x=0;
velocity.y=0;
minPiso=484;

void mostrar(RenderWindow &v):

Si el tipo de objeto es “mario”

Si se presiona la flecha derecha y salltar==true, se recorta el sprite y se muestra a mario mirando hacia la derecha. Y lado=1
Si se presiona la flecha izquierday salltar==true , se recorta el sprite y se muestra a mario mirando hacia la izquierda. Y lado=2

Si el objeto es del tipo “princesa” solamente se dibuja el sprite

Luego de esto se dibuja el sprite en la ventana (v) que se recibió como parámetro.

void caminar():

Si la tecla derecha esta presionada y la posicion en x del colisionador es menor a 415 se nueve hacia la derecha el colisionador y el sprite.

Si la tecla derecha esta presionada y la posicion en x del colisionador es mayor a 92 se nueve hacia la izquierda el colisionador y el sprite.

Si piso=6, la condicion para moverse es que la posicion en x del colisionador sea menor que 216 y mayor que 298

Void salto():

Si control==True y se presiona la tecla espacio y saltar es ==true .
A velocity.y se le asigna - jump speed y a saltar se lo pone en estado falso,lo detallado anteriormente quiere decir que se realiza el salto.
Luego de realizar el salto si control==true y la posición del colisionador + su tamaño es menor al valor de minPiso o control == true y la veloccity.y es menor que 0 y saltar == false se le suma a velocity.y la gravedad.
Si lado es == 1 se recorta el sprite mostrando a mario para la derecha en posición de salto.
Si lado es == 2 se recorta el sprite mostrando a mario para la izquierda en posición de salto.
Sino (else)
velocity.y=0
saltar=true
control=true

Mientras se ejecuta este metodo al colisionador y al sprite se los mueve 0 para la x y en velocity.y para la y.

void escalar():

Si el colisionnador intersecta con el colisionador de las escalera que recibe como parametro, saltar==false se entra en lo siguiente:

-Si la posicion del colisionador en y del personaje es mayor a la posicion en y del colisionador de la escalera y tambien se esta presionando la flecha arriba, se recorta el sprite mostrando a mario en posicion de subir la escalera y se mueve el colisionador de mario y su sprite hacia arriba control=false.

-Si la posicion en Y del colisionador de mario es menor a lla posicion en Y de la escalera mas el tamaño de y de la escalera memos el tamaño en y del colisionador de mario y se esta presionando la flecha hacia abajo,se recorta el sprite mostrando a mario en posicion de subir la escalera y se mueve el colisionador de mario y su sprite hacia abajo y control=false.

-Si la posicion de y del colisionador de mario es mayor a la posicion en y del colisionador de la escalera mas el tamaño de y del colisionador de la escalera menos el tamaño del y del colisionador de mario menos 4 o la posicion en y del colisionador demario es menor a la posicion en y de la escalera mas a el tamaño en y del colisionador de mario menos 8
puedeEscalar=false.
Sino, control=false y puedeEscalar=true.

-Si puedeEscalar==false, se ejecuta el **void caminar()**

void actualizar():

-Si la posicion en y es menor o igual a 485
piso=0;
minPiso=485;

-Si la posicion en y es menor o igual a 440
piso=1;
minPiso=430;

-Si la posicion en y es menor o igual a 383

piso=2;
minPiso=380;

-Si la posicion en y es menor o igual a 327
piso=3;
minPiso=320;

-Si la posicion en y es menor o igual a 275
piso=4;
minPiso=260;

-Si la posicion en y es menor o igual a 219
piso=5;
minPiso=210;

-Si la posicion en y es menor o igual a 159
piso=6;
minPiso=150;

bool colisionPrincesa(Personaje &target)

-Si el colisionador del objeto que esta llamando a esta funcion, en este caso mario, intersecta al colisionador del target y la funcion devuelve valor true, de lo contrario devolvera valor falso.

RectangleShape return colisionador()
Al llamar esta funcion devuelve el colisionador del objeto que la llama.

void reset()

Esta funcion resetea la posicion del colisionador y el sprite en la posicion inicial.

Clase Barril

Esta clase hereda atributos de la clase objetoMovil y a su vez tiene sus propios atributos, se encarga de modelar los barriles.

(Al hablar de barril nos referimos siempre a su sprite, su colisionador y su sumador de puntaje en conjunto)

Texture textura: Aqui se carga la imagen de los barriles
Sprite barril: Se utiliza para mostrar el recorte de la textura
RectagnleShape sumadorPuntaje: Funciona al igual que el colisionador, pero este se encarga de sumar el puntaje
int camino: Se utiliza para saber que camino tomara el barril

Barril()

Al instanciar un objeto de la clase barril, se carga la imagen en la textura, se le setea el recorte al sprite, se posiciona en la parte superior de la pantalla.
Tambien se genera un numero aleatorio entre 1 y 3, el cual se le asignara a camino
Tambien se setea el tamaño y posicion del colisionador y el sumador de puntaje.
Colisionador: Al colisionar con el, se pierde
sumadorPuntaje: Al colisionar con el, se suma puntaje

void escalar(RectangleShape m)

-Si camino==1, y se colisiona con una escalera, que fue pasada por parametros **(m)** el colisionador, el sprite, y su sumador de puntaje, se mueven hacia abajo , y si no esta colisionando, se ejecuta el metodo **mover()**

-Si camino==2 y si la posicion en x del colisionador es menor que 372 y la posicion del colisionador en y es menor que 361, se mueve el barril hacia la derecha
Al pasar la posicion 372 en x, se mueve el barril hacia abajo, al pasar la posicion 364 en y, se mueve el colisionador hacia la izquierda. Si la posicion del colisionador en y es mayor que 364 y menor que en 160 en x, camino=1;

-Si camino=3 y si la posicion en x del colisionador es menor que 92, se mueve el barril hacia la derecha, si no, si (else if) la posicion en y es menor que 310, se lo mueve hacia abajo, si no(else) se lo mueve a la derecha. Si la posicion en y del colisionador es mayor que 310 y la posicion en x menor que 320, camino=1;

Para un entendimiennto mas sencillo, cuando camino==1, el barril se mueve por los pisos hasta que colisiona con una escalera y ahi desciende, cuando camino==2 o camino==3, se realizan dos caminos definidos por nosotros, y luego se convierte a camino=1.

void actualizar():

Tiene la misma funcion que en el Personaje, pero con los valores para los barriles, establece el minimo de posicion del piso.
void actualizar():

-Si la posicion en y es menor o igual a 485
piso=0;
minPiso=489;

-Si la posicion en y es menor o igual a 440
piso=1;
minPiso=435;

-Si la posicion en y es menor o igual a 383
piso=2;
minPiso=385;

-Si la posicion en y es menor o igual a 327
piso=3;
minPiso=325;

-Si la posicion en y es menor o igual a 285
piso=4;
minPiso=280;

-Si la posicion en y es menor o igual a 219
piso=5;
minPiso=210;

-Si la posicion en y es menor o igual a 159
piso=6;

minPiso=150;

void mostrar(RenderWindow &v)

Recibe como parametro la ventana donde se lleva a cabo el juego, y en ella dibuja el sprite del barril

void mover()

-Si piso==5 o piso==3 o piso==1
Mueve el barril hacia la derecha

-Si piso==4 o piso==2 o piso==0
Mueve el barril hacia la izquierda

bool colision(RectangleShape target)

-Si el colisionador del barril intersecta con el RectangleShape recibido como parametro **(target) ,** en este caso seria el colisionador de mario, se devuelve valor true(es decir se perdio), de lo contrario se devuelve valor false(es decir la partida sigue en curso)

-Si el sumadorPuntaje del barril intersecta con el RectangleShape recibido como parametro **(target) ,** en este caso seria el colisionador de mario, se devuelve valor true(es decir, mario salto un barril), de lo contrario se devuelve valor false(mario no salto algun barril)

bool salioMapa()

Se utiliza para calcular si el barril salio del mapa.

-Si piso==0 y a su vez, la posicion en x del colisionador es menor o igual a 68, devuelve valor true, de lo contrario false.

Clase Mapa

Esta clase modela las escaleras, lo cuales sus objetos colisionaran con los objetos moviles.

Mapa(int elemento)

Al instaciar un objeto de la clase mapa, dependiendo del valor elemento, que lo percibe como parametro, setea la posicion y del tamaño del colisionador.

RectangleShape return colisionador()

Al llamar esta función devuelve el colisionador del objeto que la llama.

Main

SoundBuffer buffer: Carga en un buffer el archivo de audio que se va a reproducir. durante el juego.
Sound Musica:Se encarga de reproducir el audio que esta cargado en el buffer.
Font fuente: Se utiliza para cargar la fuente que se le asignara a los textos.
Text puntaje: Se utiliza para mostrar el puntaje.
Text puntajeAltoTexto:Es el texto que se utiliza para mostrar el highscore.
Time barrilRoto: Es el tiempo que se usa para calcular si se crea un nuevo barril o no.
Clock RelojJuego:
Time delay=seconds(3):Es el tiempo que tiene que pasar para que se cree el nuevo barril.
Time empezarJugar=seconds(4):Es el tiempo que tiene que pasar para comenzar a jugar.
Clock relojBarril:Se utiliza para contar el tiempo y se le asigna a barrilRoto.
Texture text1::Se utiliza para cargar la imagen del mapa.
Texture text2:Se utiliza para cargar la imagen del mono.
Texture text3:Se utiliza para cargar la imagen inicial del juego.
Texture text4:Se utiliza para cargar la imagen de que ganaste
Texture text5: Se utiliza para cargar la imagen de game over
Sprite sprite(text1): Se utiliza para dibujar el mapa y se le asigna la textura 1.
Sprite monito(text2):Se utiliza para dibujar el mono y se le asigna la textura 2.
Sprite menu(text3):Se utiliza para dibujar la imagen inicial del juego y se le asigna la textura 3.
Sprite win(tex4): Se utiliza para dibujar la imagen de que ganaste y se le asigna la textura 4
Sprite gameover(tex5): Se utiliza para dibujar la imagen de gameover y se le asigna la textura 5
int cantBarriles=0:Es la cantidad de barriles iniciales.
int puntajeJugando=0: Es el puntaje que esta en juego.
int puntajeAlto=lectura(): Es el puntaje mas alto cargado desde la función lectura.
int estadoJuego=0:Es el estado del juego inicial.
stringstream conversionPuntajeAlto: Se utiliza para convertir el puntaje de int a string y asi poder asignarselo al texto.
string puntajeAltoConvertido=conversionPuntajeAlto.str():Se le asigna al stream lo obtenido en la conversion del entero a la cadena del puntaje.
Personaje*mario=new Personaje(”mario”): Crea un objeto de la clase personaje del tipo mario.

Personaje*mario=new Personaje(”princesa”): Crea un objeto de la clase personaje del tipo princesa.
Barril arrayBarriles=new Barril*[10]**: Reserva 10 espacios de memoria para crear en ellos “barriles”.
Mapa escalera=new Mapa*[15]**: Reserva 15 espacios de memoria para crear en ellos objeto de la clase mapa.
buffer.loadFromFile("snd/musiquita.ogg")**:** Se carga el archivo de audio en el buffer
tex1.loadFromFile("img/mapa.png"):
tex2.loadFromFile("img/kong.png"): Se le cargan las imagenes a las texturas
tex3.loadFromFile("img/menu.png"):
tex4.loadFromFile("img/win.png"):
tex5.loadFromFile("img/lose.png");
monito.setTextureRect(IntRect(42,6,26,24)): Se recorta el sprite, mostrando al mono en estado idle
monito.setScale(2,2): Se reescala el sprite del mono
monito.setPosition(85,170): Se setea la posicion del mono
fuente.loadFromFile("font/Commodore Angled v1.2.ttf"): Se carga la fuente
puntaje.setFont(fuente): Se asigna la fuente a puntaje.
puntajeAltoTexto.setFont(fuente): Se asigna la fuente a puntajeAltoTexto
puntaje.setPosition(92,73): Se establece la posicion del texto puntaje
puntaje.setCharacterSize(20): Se establece el tamaño de carácter de puntaje
puntajeAltoTexto.setCharacterSize(20): Se establece el tamaño de carácter de puntajeAltoTexto
RenderWindow ventana(sf::VideoMode(505, 570), "Donkey Kong" , Style::Titlebar | Style::Close); Crea la ventana donde se desarrollara el juego.
ventana.setFramerateLimit(30): Se fija el maximo de fotogramas por segundo
ventana.setKeyRepeatEnabled(false): Se deshabilita la repeticion de teclas
musica.setVolume(50.f): Se asigna el volumen de musica al 50%
musica.setLoop(true): Se habilita el loop de la musica, es decir, cuando llega a su final, vuelve a empezar
musica.play(): Se reproduce la musica.

-Si el estado del juego es igual a 0 a arranque se le asigna el tiempo que paso del relojJuego.

-Si arranque es mayor a empezarJugar, estado de juego es igual a 1 y se reinicia el relojJuego.
Se dibuja la imagen inicial y se muestra el puntaje mas alto y se posiciona el texto en la parte superior
-Si el estadoJuego es igual a 1, el puntaje mas alto se posiciona mas en el centro.

Se utiliza un ciclo for para llamar al método escalar del objeto mario pasando como parametro el colisionador que devuelve el metodo returnColisionador de array de escalera y aquí termina este ciclo.

Método actulizar
Método salto
Método mostrar
Se les llama estos metodos del objeto mario.

-Si el metodo colisionPrincesa del objeto mario pasando como parametro la princesa, devuelve valor verdadero, se muestra en la pantalla que ganaste.

-Se llama el método mostrar del objeto princesa pasando como parámetro la ventana.
-barrilRoto es igual al tiempo que paso del relojBarril.
Se usa un ciclo for para llamar al metodo actualizar de cada posicion del array de barriles. Aquí termina este clclo.

-Si el tiempo de barrilRoto es mayor al delay y la cantidad de barriles es menor que 5, se reinicia el relojBarril, se setea el sprite del momno tirando el barril, se crea un nuevo barril usando como indice cantBarriles, cantidadBarriles se le suma 1.

Sino (else) se setea al sprite del monito sin estar tirando un barril.
Se utiliza un for donde el maximo valor de la i es la cantidad de barriles, dentro de el se le llama al metodo mostrar el arry Barriles pasando como parametro la ventana, dentro de este for se encuentra otro en el cual se llama a la funcion Escalar de el array Barriles pasando como parametro el colisionador que devuelve el metodo returnColisionador del objeto escaleras,saliendo de ese ciclo, si el metodo **SalioMapa** de array Barriles, devuelve el valor verdadero, se elimina el objeto el cual esta devolviendo el valor verdadero y se crea un nuevo barril en la misma posicion dell que se elimino.
-Si el metodo colision del array barriles, pasando como parametro el colisionador que devuelve el metodo returnColisionador del objeto mario devuelve el valor verdadero, es decir, mario colisiona con un barril, estadoJuego=3 y si puntajeJugando es mayor a puntajeAlto se ejecuta el metodo escritura pasandolo como parametro puntajeJugando. Lo que hace este metodo es guardar en el archivo el puntaje.

-Si el metodo puntaje del array barriles, pasando como parametro el colisionador que devuelve el metodo returnColisionador del objeto mario devuelve el valor verdadero, es decir, mario salto un barril, se le suma 10 al puntajeJugando .

-Se ejecuta el metodo draw de la ventana dibujando puntaje,puntajeAltoTexto, el sprite del mapa y se ejecuta el metodo display de la ventana.

-Si juego= 3,cantBarriles=0,arranque es igual a tiempo que paso del relojJuego y si arranque es mayor a empezarJugar, se reinicia el relojJuego y estadoJuego=1.

int lectura():

Esta funcion devuelve lo leido del archivo puntaje como un entero

ifstream archivoLectura: Es el archivo que se va a usar para el puntaje.
string texto: Se iutilizara para guardar lo leido del archivo.
string nombreArchivo=”res/puntaje.txt”: Se usa para indicar la ruta del archivo.
Se abre el archivo como lectura.
Mientras no sea el final del archivo, se guarda el texto la linea del archivo.
Se cierra el archivo
stringstream conversion(texto)
int p=atoi(texto.c_str()):: Se realiza la conversion de string a entero.
Return p: Se devuelve el valor del p.

void escritura(int p):
Este metodo se utiiliza para escribir en ell archivo del puntaje mas alto que se recibe como parametro.
ofstream archivo:: Es el archivo que se va a usar en modo de escritura.
string nombreArchivo=”res/puntaje.txt”:Se usa para indicar la ruta del archivo.
Se abre el archivo como escritura.
-Se escribe en el archivo el valor de p.
Se cierra el archivo.