

Tarea 2 - CC3501-1

roTamara Celis

15/12/20

Solución:

Para la solución se fueron creando distintas piezas para el juego, algunas con clases especiales, tales como:

- Barra(): crea un cubo (luego se escala debidamente), que en un comienzo se dibujará (self.dibujo) y además es real (self.real), esto ayuda con la implementación de plataformas falsas. A su vez se crean las posiciones en sus coordenadas posX, posY y posz. Incluye un método de dibujo de la barra en las coordenadas correspondientes. Existen los métodos moveX(), que mueven las plataformas hacia los lugares que en un futuro serán indicados por el csv.
- CreateBarras(): crea las barras al leer el csv. Dependiendo en que posición se haya un 1 o x se van creando y trasladando barras para luego guardar en dos listas su modelo y transformación. Si se crea una barra o plataforma en función a una x, se indica que esta es falsa (en el ejemplo de como leer un csv se leía el primer signo con row[0], pero en mi computador no funcionaba y al hacer esto me entregaba toda la lista, por esto se leen como se muestra en el código). Para dibujar las plataformas, se toman las listas de modelos y transformaciones y se dibujan cada una con la función dibujar de una barra en particular.
- Fondo(): crea los fondos del juego. Son 4 cubos en realidad que simulan el “manto” de un cubo. Esta clase tiene dos listas de modelos y transformaciones de todos los cubos que formen el fondo. La función Draw dibuja todos los fondos en función a las listas propias de la clase.
- FondoAzul(): Cuando Amy sube lo suficiente se van creando nuevos fondos azules que simulan el cielo, estos se van agregando a las listas de la clase Fondo()
- Flecha(): Crea los ataques hacia Amy de los enemigos, sólo se ven las flechas que se mueven hacia donde esta ella, para esto se le entrega la posición de Amy en un determinado tiempo y se crea el objeto de la clase. Para dibujar, simular a las clases anteriores tiene el método draw y además un undate(dt), que ayuda a cambiar la posición de la flecha para simular un movimiento hacia Amy.

El piso del juego y Amy se implementaron dentro de la función Amy_Jump la cual recibe la estructura del juego en un csv.

Para la función se definieron todos los pipelines a usar y creamos todos los objetos, los definidos en las clases anteriormente nombradas y además Amy, el anillo que es el trofeo (estos se crean con la lectura de un obj) y el piso. Además, se crean 3 imágenes que indican el inicio del juego, cuando se pierde y cuando se gana.

Posteriormente se definieron muchas variables que nos servirán en la implementación del juego (cada una se encuentra comentada dentro del código para entender que son), las más importantes son salto, que indica si se puede saltar o no, perder que muestra cuando se puede haber perdido dentro del juego por caerse de una plataforma o saltar y caer al suelo, arrow la cual avisa si se puede hacer una nueva flecha, bt esto sirve para saber cuándo el personaje

subió a alguna barra y se cae, indicando si es que este perdió y ganar que dice si el jugador gana.

Dentro del ciclo se tiene primero calcular una variación de tiempo que facilita el movimiento de todos los objetos (dt) y el tiempo actual ($t1$).

Para las cámaras se tiene una vista fija, pero si se presiona B, N o M esta muestra otras perspectivas de Amy, dentro de su implementación se incluyó la posición del personaje como variables dentro de la vista. Así si Amy se mueve las cámaras se mueven con ella.

Se implementa la luz, se dibuja el fondo y se va revisando si se debe crear otro fondo azul o no, así se va simulando un fondo infinito. También se dibuja el piso, Amy, el anillo (que va girando en función de "ra"), las imágenes nombradas y las barras.

Además:

- El ciclo se comienza con la gravedad activada.
- Se revisa si para la lista de barras, las posiciones calzan con las de Amy, para esto los valores en x, y y z se detecta a Amy dentro de un rango.
- Si es que se detecta a Amy revisamos si es una barra real, de serlo se apaga la gravedad con el fin de que Amy se quede en la barra y se dice en bt que el personaje ya subió a alguna barra. Si esto es falso la barra se deja de dibujar y la gravedad sigue activada.
- Para implementar la gravedad se estableció que, si Amy no esta en el piso o en una barra a la coordenada z del personaje se le fuera restando $3*dt$, así simula una caída gradual.
- El movimiento de Amy se generó con las teclas A, D (hacia los lados), W, S (adelante o atrás) y para saltar se usa la barra espaciadora. Al presionar cualquiera de estas teclas se van modificando las coordenadas de Amy en un factor de $5*dt$.
- Luego de apretar la barra espaciadora, se revisan varias cosas que permiten saltar:
 - Si la variable salto dice si es que puedo saltar, lo bloqueo diciendo que es falsa para luego calcular el máximo de mi salto
 - Si aún no llego al máximo de mi salto, a su vez no he perdido porque llegue a mi limite y no he caído en una barra, sigo subiendo
 - Si la gravedad se apagó y no he perdido por ser atacado por una flecha, puedo volver a saltar y además no he perdido así que perder es falso
 - Si mi altura es mayor que mi altura máxima, "perdí" (esto se verifica después con otras condiciones)
- Para las flechas se tiene
 - Si puedo mandar una flecha, guardo mi tiempo actual y bloqueo la opción de nuevas flechas, además de hacer una flecha hacia la posición de Amy
 - Se revisa que si el tiempo actual es mayor en 3 en relación con el tiempo de cuando se mando la flecha, de ser así se desbloquea la opción de mandar nuevas flechas.
 - Se revisa si alguna flecha toco al personaje, de ser así se dice que este perdió
- Si el tiempo es mayor a 8 ya se pueden empezar a mover las flechas hacia Amy, esto da tiempo de mostrar la imagen de inicio.
- Si se perdió por las razones nombradas en el código, se muestra la imagen de game over, además, de comenzar de rotar al personaje.

- Si llego a una plataforma de las mas altas significa que se ganó. Con esto se mueve el anillo hacia Amy y también se muestra en la pantalla la imagen de ganador.

A continuación se presentan imágenes:

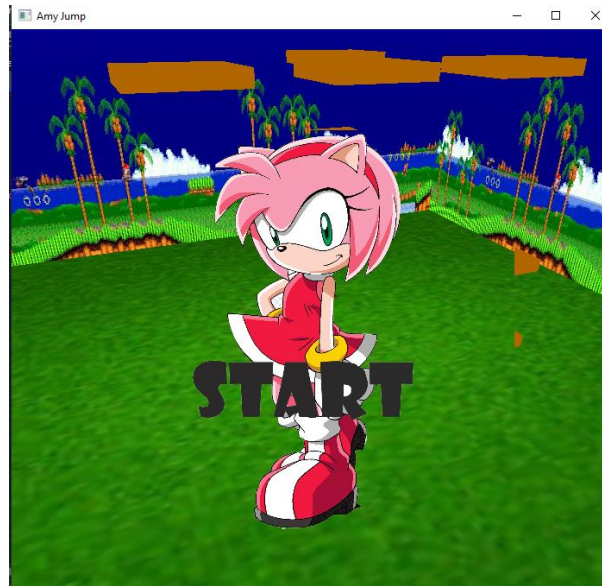


Imagen de inicio

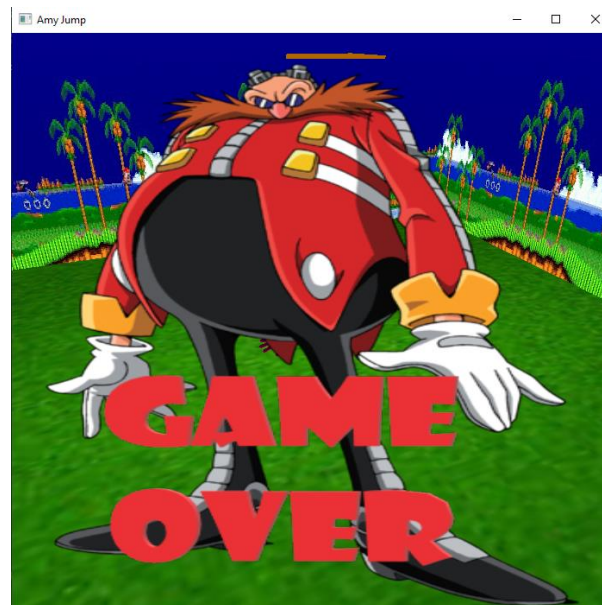


Imagen de perdida

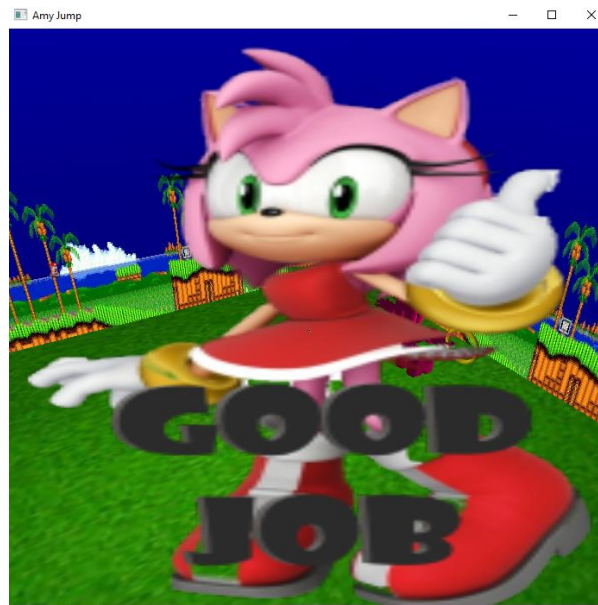


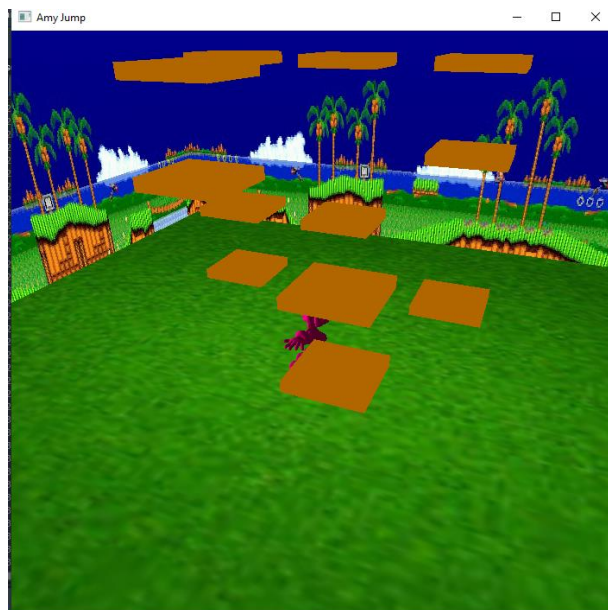
Imagen de ganador



Vista general del juego



Vista al mantener presionado la letra B



Vista de mantener presionada la letra N



Vista de mantener presionada la letra M