

Informe Tarea 2B

Monkey Jump

Integrantes:	Valentina Garrido
Profesor:	Nancy Hitschfeld
Auxiliar:	Andrés Cerda
	Pablo Pizarro
	Alonso Utreras
Ayudantes:	Tomás Calderón
	Beatriz Grabolosa
	Sebastián Olmos
	Nadia Decar

Fecha de realización: 13 de diciembre de 2020

Fecha de entrega: 13 de diciembre de 2020

Santiago, Chile

Preámbulo

Un gatito observa en lo más alto de la torre de rascaderos una bola de lana con la cuál jugar, pero mientras escala, la mascota del hogar (un travieso perro) se da cuenta y quiere meter sus narices donde no debe.

Deberás nuevamente ayudar al gato a conseguir la bola de lana mientras evitas que caiga a la base. En altura estará tranquilo fuera del alcance de ciertas patitas felpudas. Sin embargo, esta vez será más difícil.

El escenario se ha vuelto más complejo y muchas estructuras se han desgastado por lo mucho que el gato jugó durante el proyecto anterior. Si vas a guiarlo por estas, debes hacerlo con cuidado pues se desprenden tras un momento de ser pisadas.

Además, estamos en una complicada transición de primavera a verano de un año muy caótico y, en particular, plagado de polillas.

Estos animalitos no son dañinos, pero han llamado la atención de los humanos sirvientes del gato, quienes se han juntado a observar la aventura, expectantes a que ocurra algún tipo de cacerío.

La verdad es que nuestro amigo gatuno en su comodidad ha olvidado cómo lidiar con insectos. Por alguna razón que desconoce, los humanos se decepcionan cuando los deja en paz.

Sus sirvientes toleran que el gato deje a lo más a dos polillas con las cuales se topa seguir volando. Repetir esto más veces hará que se impacienten y les de el impulso de tomar al gato y bajarlo de la estructura .

En lo posible, evita que el gato se tope con polillas, así nadie se decepciona y la aventura puede continuar ininterrumpida.

El diagrama anterior muestra a grandes rasgos la estructura del programa.

La escena se construye a partir de elementos en tres dimensiones entregando los datos del modelo (matrices de transformaciones), vista y proyección (MVP) y con iluminación acorde al modelo de Phong.

Durante el juego, se puede controlar al gato, el personaje principal, moviéndolo en el escenario y saltando sobre las plataformas presentes, hasta que se llega a una condición de término, que sería ganar o perder el juego.

En ambos casos, se desactivan los controles y la generación de enemigos y se produce una animación final, con una cámara que muestra el escenario en rotación.

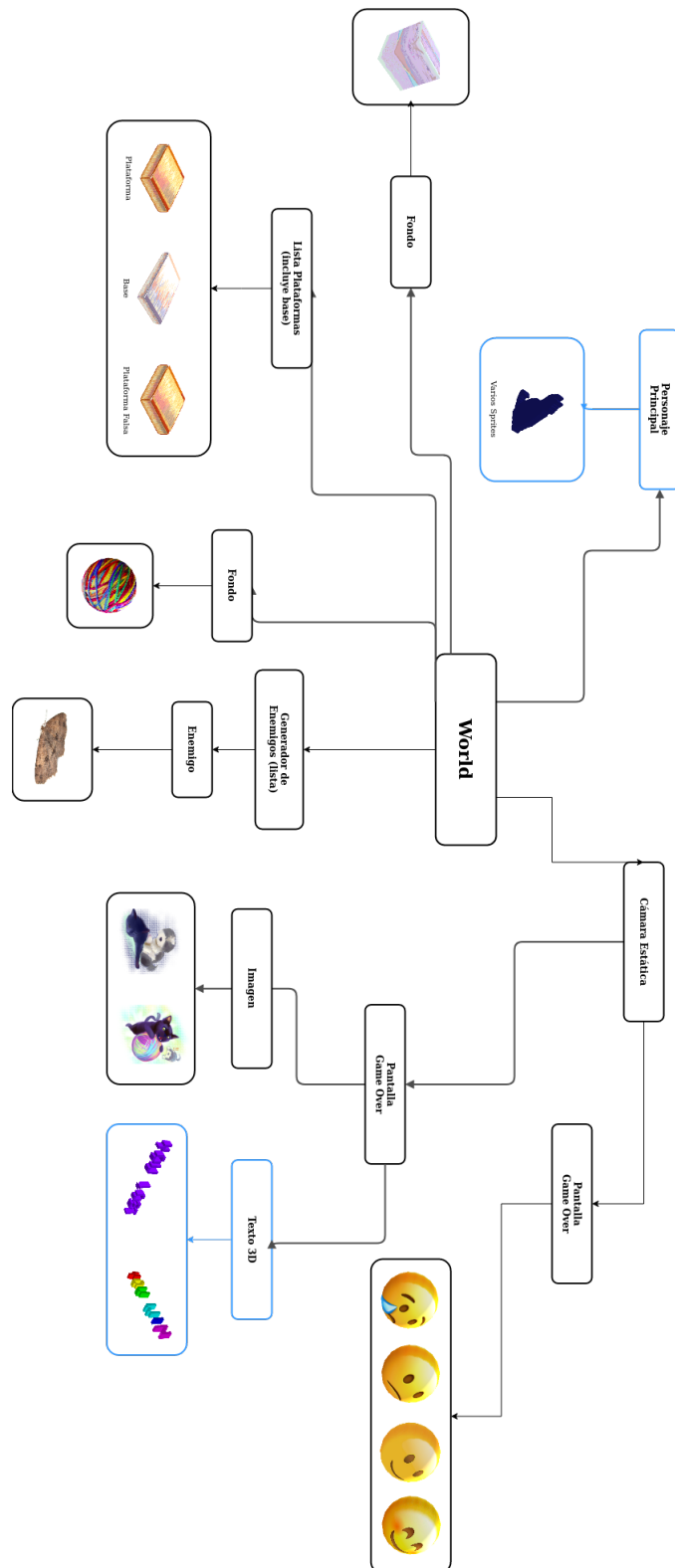
1.1. Eventos

Se detallan los eventos que desencadena el personaje principal al interactuar con ciertos objetos que aparecen en el diagrama:

- Cuando el jugador ha logrado que el gato haya saltado por las plataformas a la cima y toque la bola de lana, se produce una animación y gana la partida.
- El jugador pierde cuando el gato cae a la base. Se perdona que esto ocurra desde la primera fila de plataformas, pero a partir de la segunda en adelante, la condición de término aplica.
- Chocar con una polilla hace que el jugador pierda una vida. El gato empezará a parpadear durante los próximos 1,5s, donde será invencible y interactuará con más enemigos hasta que se acabe la animación.
- La pérdida de una vida actualiza el medidor de vidas en el costado superior derecho (el emoji) haciendo que cambie a una cara de estado anímico más bajo. Cuando se pierde por cualquier método el juego, la cara cambia a llanto, cuando se gana, cambia a alegría.
- Tocar una plataforma falsa provoca que esta comience una animación de temblor. Se dan 3s pensados para que el jugador reaccione antes de que la plataforma comience a caer y salga de la escena

El programa termina cuando se cierra la ventana.

1.2. Jerarquía de la Escena



En el diagrama se ven los objetos que componen la escena (el mundo) con su apariencia.

Los objetos que aparecen delineados con azul son dibujados con un shader de luz para objetos que no poseen texturas, sino que especifican colores, los demás objetos usan shaders de iluminación con textura.

Los objetos que pertenecen a la rama de cámara estática usan otra cámara que, como el nombre indica, no se mueve de su posición, para simular un heads-up display (HUD) de estos elementos.

- El personaje principal y las letras se dibujan leyendo una imagen y generando de manera raster cuadriláteros que cubren zonas contiguas donde la opacidad es uno. Se crean dos tapas y se unen con cuadriláteros en los bordes para que la figura se vea cerrada.
- La bola de lana y las vidas se modelaron como dos semiesferas juntas, usando parametrizaciones por ángulos ϕ y θ . Las coordenadas UV para las texturas se obtienen con la misma parametrización y se aplican a ambas semiesferas.
- Las plataformas son modeladas como cubos con textura. Las plataformas normales y las falsas usan la misma textura, por la que sólo pueden distinguirse una vez pisadas
- Los enemigos se modelan con cubos que tienen la textura sólo en la tapa superior
- El fondo es un gran cubo con textura que contiene la escena
- La imagen de Game Over es un cuadrilátero con textura.

1.3. Colisiones

El desplazamiento por el plano xy ocurre aumentando el parámetro de velocidad en xy, calculando las velocidades v_x y v_y con el coseno y seno del ángulo actual del gato respectivamente y actualizando sus posiciones con el tiempo obtenido como $x = v_x t$ e $y = v_y t$. La velocidad v_{xy} es positiva o 0 y posee un límite máximo.

El desplazamiento vertical se calcula a partir de una velocidad en z que es 0 sobre una superficie y disminuye en el aire con una gravedad definida. Esta velocidad sí tiene signo. Positivo provocará que el personaje se eleve y negativo que caiga.

Colisionar con paredes o plataformas cambia v_{xy} y v_z inmediatamente a 0.

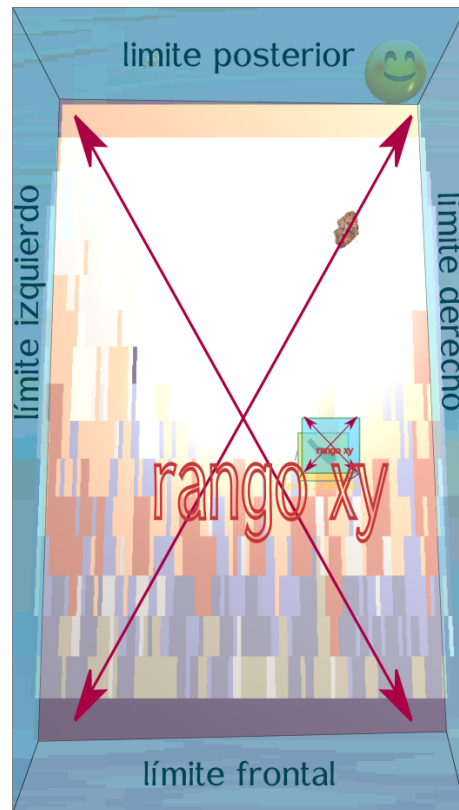
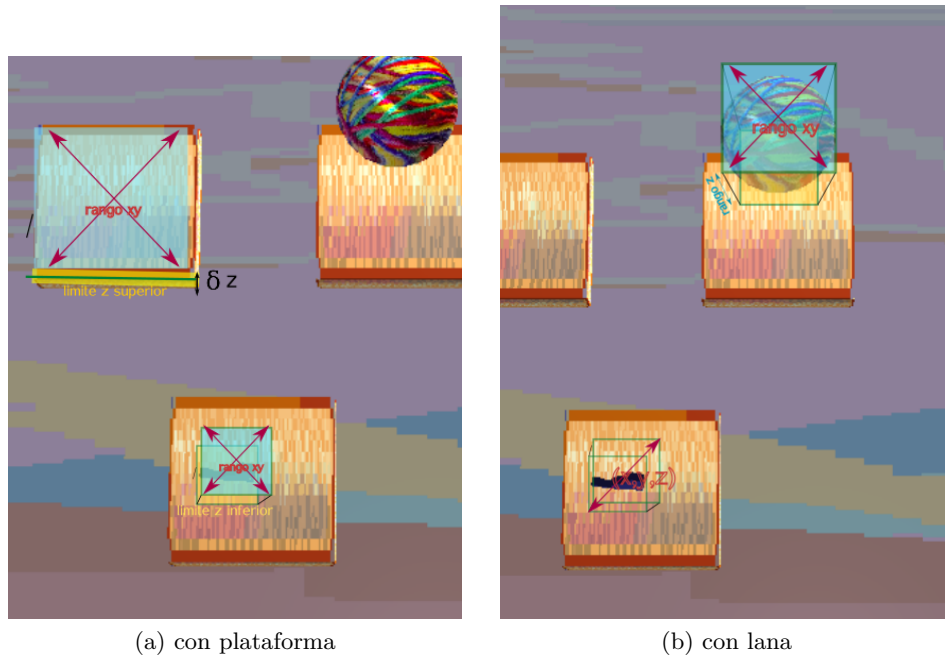


Figura 1: Tipos de colisiones

En las ilustraciones se observan los 4 tipos de colisiones del gato con los distintos objetos con los que puede interactuar (en celeste).

Entre las colisiones con objetos sólidos se tienen el caso (a), donde hay colisión si entre el gato y la plataforma se intersectan sus rangos en x e intersectan sus rangos en y , y además si el límite inferior del gato y el límite superior de la plataforma en z están dentro de un rango delta.

En ocasiones cuando el gato cae con una velocidad muy grande, puede que las diferencias en posiciones en y superen el rango delta, causando que el gato atraviese plataformas.

Otras colisiones sólidas se ven en (c). La base es un caso especial de las plataformas donde además de intersectar los rangos en x e y , el límite inferior del gato debe ser menor al límite superior de la base más un δ . Así se evita que por la razón anterior el gato atraviese la base y caiga infinitamente.

En la imagen también se observan dónde están las paredes invisibles del juego, que evitan que el jugador pueda sacar al gato de escena. En este caso, los rangos x e y del gato deben estar contenidos dentro de los límites respectivos de las paredes para que no haya choque. Si hay choque, el jugador debe cambiar el ángulo del personaje para seguir avanzando.

Las colisiones con objetos no sólidos que se pueden tocar se esquematizan en (b). En el ejemplo se muestra la bola de lana, pero aplica a los enemigos también. En este caso, el centro (x,y,z) del gato debe contenerse dentro de los límites x , y y z para que haya interacción.

1.4. Enemigos

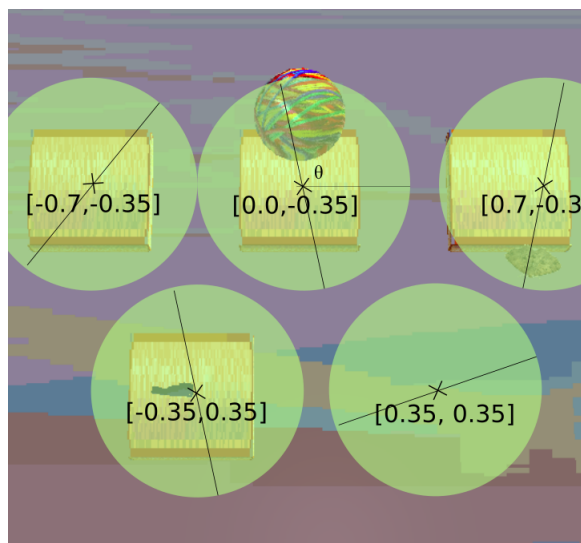


Figura 2: Posibles centros por donde pasa el enemigo.

Los enemigos tienen un alto comportamiento aleatorio el cual se modela con las funciones del módulo `numpy.random`.

Aparece un enemigo en escena si el número entregado por `numpy.random.random()` es menor que un parámetro `appear_tolerance`, que está definido como 0.02. Puede haber un máximo de 8 enemigos simultáneos en escena.

El movimiento que define el enemigo comienza en un punto de un círculo dado por un radio de aparición, por el cuál se desplaza en línea recta a velocidad constante través del diámetro. Cuando llega al final de su recorrido el enemigo deja de estar activo y desaparece de la escena.

El ángulo que hace el diámetro con el eje x se escoge de manera aleatoria. El centro del círculo

puede estar en 5 posiciones correspondientes al de las plataformas como se ve en la imagen (notar que el radio del círculo es sólo ilustrativo, pero en realidad es más grande).

La posiciones z donde aparece el enemigo corresponden también a las posiciones z de las plataformas y se escogen al azar.

1.5. Cámaras

Para generar las vistas de las escenas se usan varias cámaras.

1.5.1. Cámara Frontal

La cámara frontal, como indica el nombre, muestra la vista desde la cara frontal de la escena. Está inicialmente a 5 unidades desplazada del centro en el eje X y a 1 unidad del eje Z . Así, esta vista es la más semejante a la del proyecto en 2D, pero el desplazamiento en Z permite ver algo de la profundidad para que el jugador se desplace bien entre distintas columnas de plataformas.

Tiene su centro inicialmente en el punto $(0,0,0)$.

La cámara frontal y el centro se desplazan en el eje Z acorde al movimiento en Z del personaje.

1.5.2. Cámara Aérea

También llamada Eagle Camera en el código. Muestra la escena como si estuviera una distancia arriba del personaje principal y mirando hacia la base.

Esta cámara también tiene una visibilidad de plataformas variable. Se pueden observar 3 niveles, el que el personaje principal está, el nivel bajo este y el de arriba. Esto es para que la visión no se vea tan obstruida por varios niveles.

Está desplazada inicialmente en $(0,1,4)$ y se desplaza en z hacia arriba con el movimiento en z del personaje, como la cámara frontal.

1.5.3. Cámara de Primera Persona

También llamada Follow Camera. Muestra la escena desde los ojos del personaje principal. Parte en su posición, se desplaza la cámara y el centro en xyz con los movimientos del personaje y la dirección varía también al cambiar el ángulo del personaje.

El dibujo del gato se desactiva temporalmente para que no obstruya la vista.

1.5.4. Cámara Estática

Como fue explicado en la sección de jerarquía, es usada exclusivamente para los elementos que componen el HUD. Está a la misma distancia en X que la cámara frontal, pero Z está en 0. Su centro también está en $(0,0,0)$. No se altera con los cambios en el personaje principal

1.5.5. Cámara Game Over

Una cámara especial que aparece cuando se termina el juego. Tiene la vista, el centro y la posición de una cámara frontal, pero va rotando lentamente alrededor de la escena.

2. Instrucciones de ejecución

2.1. Lista de Estructuras

El programa es capaz de personalizar las plataformas del nivel pasando un archivo csv en la línea de comando como parámetro adicional a la ejecución. En el siguiente ejemplo, se pasa un archivo llamado `structure.csv`:

```
1 python cat_jump.py structure.csv
```

Se deben declarar exactamente 5 columnas, indicando con 1 la presencia de una plataforma, con 0 su ausencia, y con una x si es falsa. El número de filas es arbitrario.

También se pide no rellenar ninguna fila con solo 0, puesto que impediría que el gato avance y termine el nivel.

Si no se entrega un archivo de plataformas, se usa el ejemplo de enunciado por defecto, que se encuentra en el archivo `ex_structure.csv` y se notifica en la consola.

2.2. Controles

El gato se controla usando las teclas **A**, **W**, **D** y la barra espaciadora.

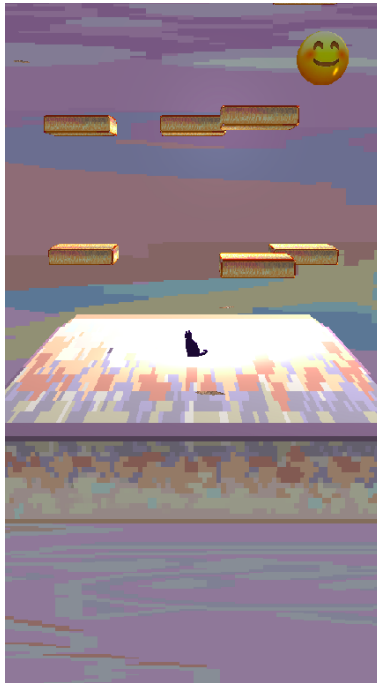
La teclas **A** y **B** modifican el ángulo a partir del cuál se define la dirección del movimiento. **A** gira al personaje en sentido horario en su eje z, la tecla **D** lo rota en sentido antihorario.

Presionar y mantener la tecla **W** hace que el personaje acelere y se deplace en la dirección a la que apunta y desacelera al soltarla.

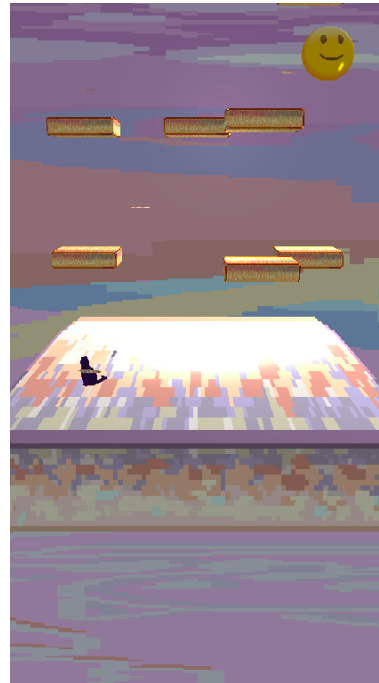
Los saltos se realizan presionando la barra espaciadora una vez. Tienen un alcance fijo. Sólo pueden realizarse estando sobre suelo, no existen saltos múltiples en el aire.

Además, se puede cambiar la vista del escenario al cambiar de cámara con las teclas **B**, **N** y **M**. La tecla **B** cambia a cámara frontal, la tecla **N** a vista aérea y la tecla **M** a vista en primera persona.

3. Resultados



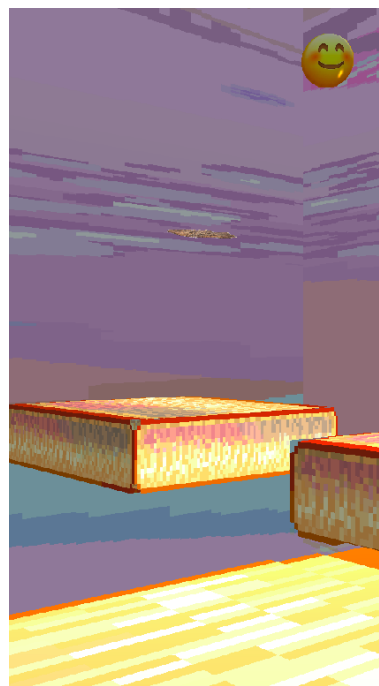
(a) Gato sobre base. Vista frontal.



(b) Gato perdiendo una vida



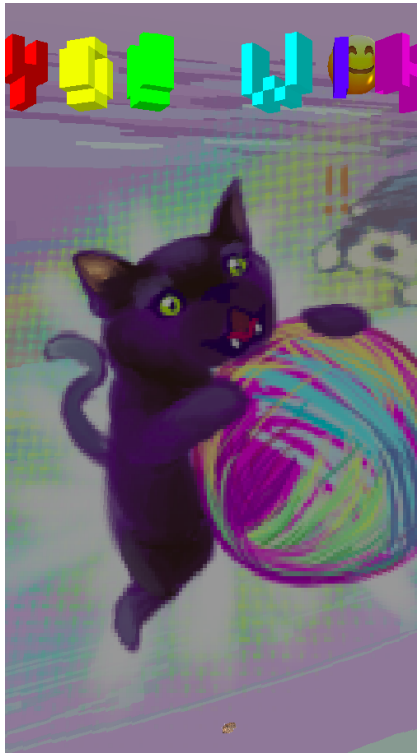
(c) Gato cercano a la lana. Vista aérea.



(d) Cámara en primera persona

Figura 3: Interacciones y Plataformas

Capturas de imágenes varias que ilustran varias partes del programa.



(a) Pantalla de victoria



(b) Pantallas finales

Figura 4: Pantallas Juego Finalizado

En las imágenes de arriba se observan las pantallas de término de juego. También se pueden apreciar las letras durante una animación de giro.