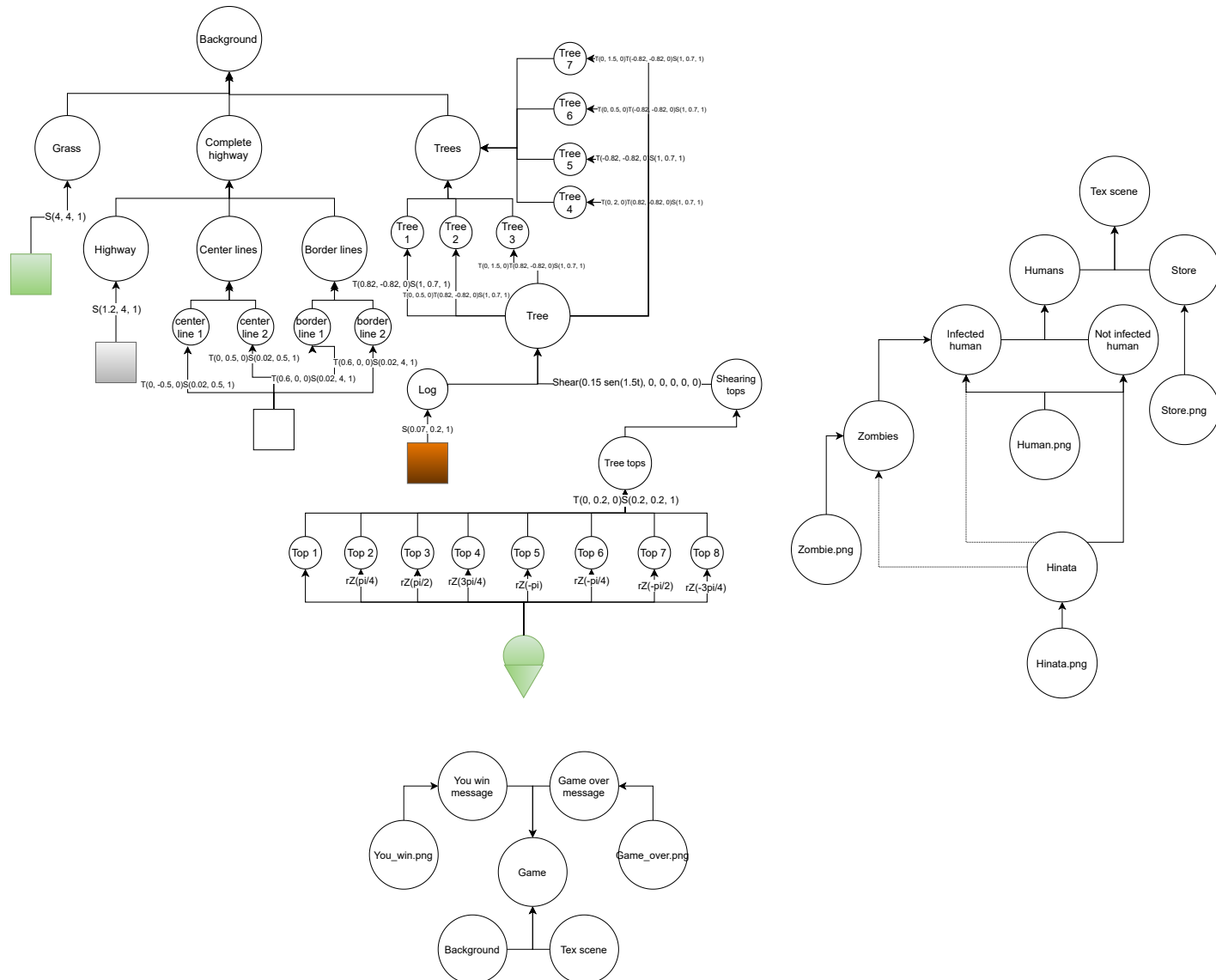


Solución Propuesta

Se utilizan dos grafos de escena para elementos del fondo *Background* y elementos con texturas *Tex scene*, unidos en un solo grafo global llamado *Game*, que incluye los mensajes de *you win* y *game over*, enlazados a sus respectivas texturas.



Por otro lado, se utilizan clases para humanos, zombies, jugador (Hinata) y la tienda. Estos objetos interactúan entre ellos (cambian estados) y con el grafo de escena para generar efectos en pantalla: Contagio entre humanos, movimiento de entidades, por decir algunos ejemplos.

Vale decir que los humanos en pantalla no se transforman en zombies, pero si se pueden infectar al contacto con un zombie. De todas formas, el jugador pierde si toca un zombie o si se contagia y se vuelve zombie.

Grafo background

3 componentes principales: Un nodo para el pasto, otro para la carretera y un último para los árboles. El pasto no consta de mayor complejidad: Cuadrado verde escalado. La carretera posee líneas en el centro y bordes, en un fondo gris.

El nodo de árboles posee 7 árboles distribuidos en la escena. Cada árbol posee un tronco y copas hechas a partir de una figura nueva.

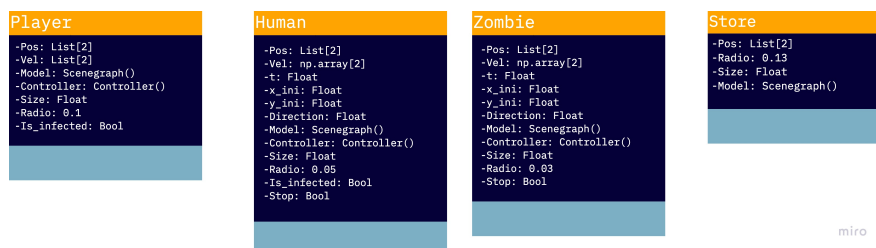
La nueva figura es semejante a un cono de helado (ver tree tops en la figura). Se crea a partir de un círculo trasladado, cuyos vértices se enlazan a un punto fuera del círculo. A estas copas se les aplica un shearing sobre los vértices durante la ejecución, para generar un efecto de viento. Los árboles son dibujados con *GL_DYNAMIC_DRAW*, mientras los demás nodos con *GL_STATIC_DRAW*.

Grafo de texturas y clases controladoras

Se consideran los humanos y zombies, donde originalmente todos son humanos pero con distintos niveles de infección: Zombies son humanos infectados gravemente. Se dibujan con *GL_DYNAMIC_DRAW*.

Hinata es en principio un humano que puede infectarse o puede volverse zombie. En este último caso, el juego termina y arroja un mensaje de *game over*. Es dibujado con *GL_STREAM_DRAW*.

La tienda es un modelo estático y su rol es indicar la victoria si el jugador colisione con esta. Como es estática, se dibuja con *GL_STATIC_DRAW*. Las clases se resumen en el siguiente diagrama:



Se definen funciones para los requerimientos del enunciado: Spawn de humanos y zombies

que agregan nuevos nodos al grafo de escena cada T segundos, detector si Hinata llega a la tienda (gana), toca un zombie o al estar contagiado se convierte en zombie (pierde), desplegar mensaje de victoria y derrota, detectar humanos contagiados y cambiarlos de nodos, funciones trigonométricas en movimiento de entidades (varían en sentido, posición y velocidad, aleatoriamente).

Se revisan constantemente las entidades para detectar colisiones y cambiar estados, como: Contacto entre contagiados y Hinata o no contagiados (cambia estado a infectado), humanos con zombies (que solo los infectan), zombies con Hinata y Hinata con la tienda.

Por último, se borran los modelos de zombies y humanos que se encuentran fuera de la pantalla.

Shaders

Se implementan 4 shaders según lo que se requiera: Un shader con transformaciones para el background, un shader de texturas con transformaciones para entidades, tienda y victoria/-derrota, un shader que amplifica el verde en coords. de texturas para contagiados y un shader con transparencia al finalizar el juego. Con la tecla *escape* se cierra la ventana de juego.

Instrucciones de Ejecución

El programa se inicia al ejecutar el siguiente comando en la ventana de comandos:

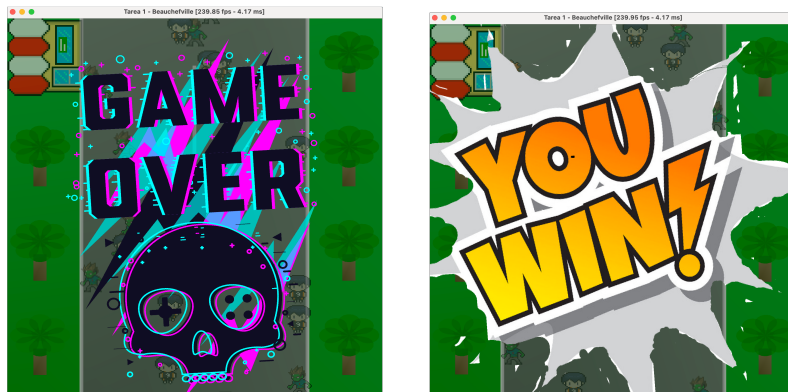
```
python survival.py Z H T P
```

Con Z y H el número de zombies y humanos que aparecen en pantalla, T tiempo de spawn de entidades y tiempo en que un contagiado pueda volverse zombie, y P probabilidad de volverse zombie si se está contagiado o que un humano aparezca contagiado en pantalla.

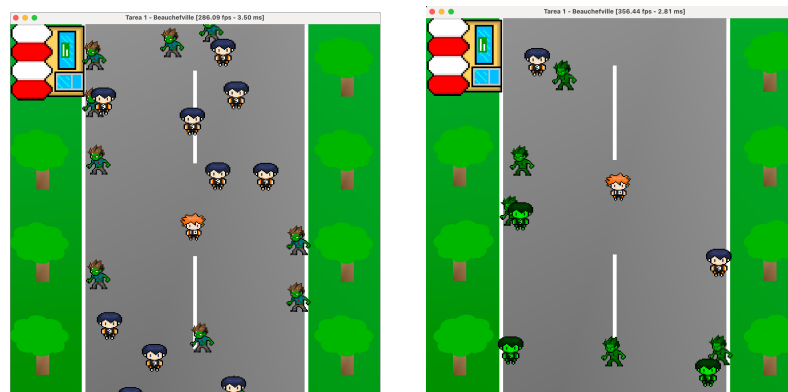
El personaje se mueve con las teclas WASD y con la barra espaciadora se pueden observar las entidades contagiadas, que cambian a color verde.

Resultados

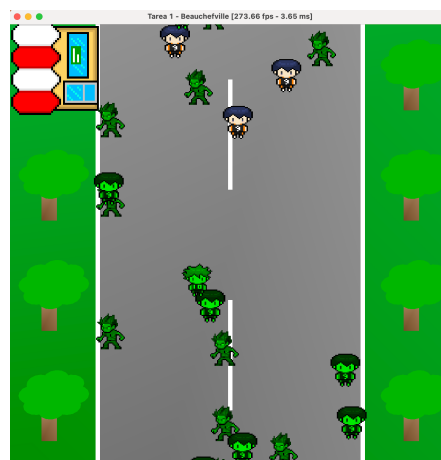
Se obtienen los siguientes screenshots de algunas de las funcionalidades importantes del programa:



Mensajes de derrota y victoria. Se aprecian las transparencias al terminar. Se agregó además un visualizador de estado del juego que indica los fps actuales.



Vista general del programa. Al presionar la tecla espaciadora, se pueden visualizar las entidades contagiadas (los zombies son contagios más graves). Notar además el movimiento de los árboles entre ambas figuras.



Hinata toca un humano infectado y se contagia

Autoevaluación

Criterio-Puntaje	0	1	2	3
Evaluación general			x	
OpenGL				x
Shaders			x	
Modelos geométricos			x	
Transformaciones				x
Texturas			x	
Modelación jerárquica				x
Curvas	x			
Funcionalidades mecánicas o lógica de juego			x	
Entradas o Control de usuario			x	
Visualización de estado del programa			x	