

Solución Propuesta

La solución propuesta cumple con todos los requisitos presentes en el enunciado. Se tiene un escenario que mezcla formas simples para crear objetos más complejos (árboles, tienda) y un fondo con textura. Como centro del juego, tenemos la clase Controller, que maneja el control por parte del usuario y mantiene variables clave para el funcionamiento del juego. Los elementos dinámicos del juego se definen en 3 clases de Python:

- Player, que es el personaje principal del juego, consta de variables para el movimiento del objeto-jugador por la escena, variables necesarias para el sistema de contagios, y su modelo de texturas. Además tiene los métodos necesarios para actualizar su posición, su estado de contagio-zombie, y para detectar colisiones.
- Human, que representa tanto a humanos como a zombies en la escena, consta de variables para el movimiento del objeto, variables necesarias para el sistema de contagios, y un modelo de texturas intercambiable que incluye las texturas de humanos y zombies. Además, tiene los métodos necesarios para actualizar su posición, su estado de contagio-zombie, detectar colisiones, y removerse de la escena.
- Aura, que es el checkpoint para considerar una partida ganada, consta de variables para determinar su posición. Además, tiene un método para detectar una colisión con el jugador.

Se crea en el comienzo del programa, la cantidad de zombies y humanos que pueden llegar a ser utilizados, y se agregan a una las listas de zombiesOut y humansOut respectivamente, para indicar que no están en escena. Luego, al comenzar a iterar el juego, según los parámetros dados en un comienzo, se agregan zombies y humanos a las listas zombiesIn y humansIn respectivamente, y al grafo de escena para ser visualizados en pantalla.

Cada una cantidad de segundos especificadas en un comienzo, se llama a un método de la clase Human y Player, para que en caso de estar contagiados (infected=1), con probabilidad P (seteada en un comienzo) pasen a ser zombies. En el caso de que el player pase a ser zombie, se cambia la variable gameover del controlador, para desplegar dos imágenes por haber perdido, mientras tanto el juego reinicia sus parámetros para poder volver a jugar sin tener que volver a abrir el programa luego de presionar enter. Para el caso en el que el player colisione con el objeto Aura, se modifica también la variable gameover del controlador, pero con el valor opuesto, de tal forma que se desplieguen dos imágenes por haber ganado, y de igual forma que al perder, el juego se reinicia al apretar la tecla enter.

Todo humano que entra en escena tiene un 50% de probabilidades de entrar contagiado. Esta probabilidad puede ser modificada en la ejecución del programa, pero si no se especifica, queda en 50% por defecto.

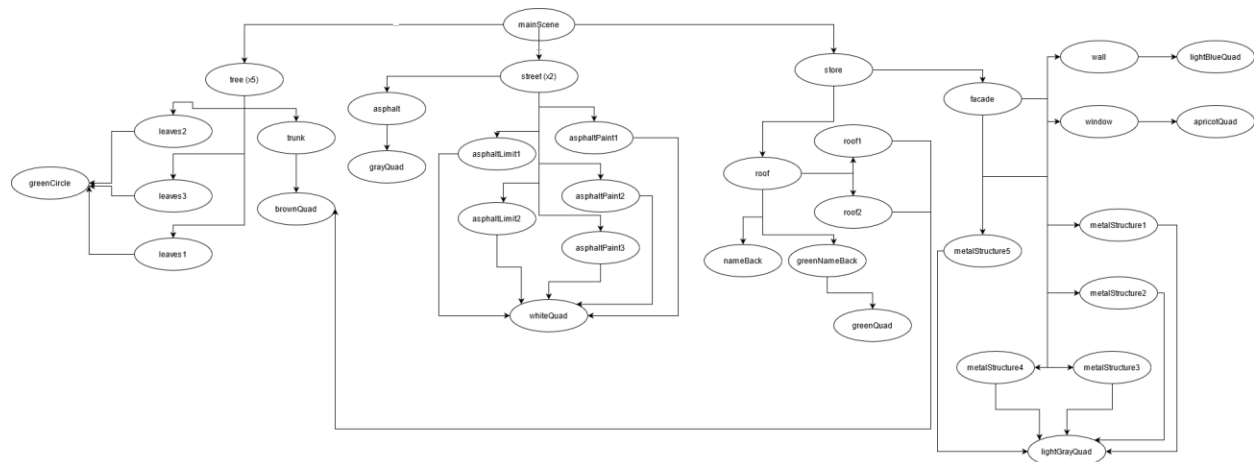
Debido a que el método de las colisiones es en base a listas que se verifican constantemente, el programa es muy ineficiente. Para intentar reducir esa ineficiencia, sólo los humanos que no se han convertido en zombies, y el player son quienes verifican si chocan con los zombies de la lista zombiesIn; y en caso de hacerlo, pasan a ser zombies inmediatamente. Además, los humanos no contagiados y el

player verifican constantemente si han chocado con un humano contagiado (incluyendo el mismo player); y en caso de hacerlo, pasan a estar contagiados automáticamente.

Visualmente, un humano y un zombie se diferencian por su textura; lógicamente, por su variable zombie. Aún así, para mantener una cantidad fija de humanos y zombies, la clase Human tiene una variable wasHuman, que indica si en un comienzo se creó siendo un humano o un zombie; de esta manera, al salir de escena, un humano que se convirtió en zombie, vuelve a ser humano, y al momento de volver a entrar a escena se vuelve a calcular de manera aleatoria si entra contagiado o no.

Al usar las gafas, se agrega a la escena los shapes de los humanos contagiados, pero se dibujan con un shader que los sombrea de rojo. Por otro lado, se agregó una funcionalidad “infectedMode” para que en el momento en que el player se contagie, el fondo se dibuje con un shader que le agrega color rojo a las figuras, de esta manera el player puede saber que está contagiado y puede perder en cualquier momento.

Sin entrar en muchos más detalles de implementación o códigos creados, se muestra la versión resumida del grafo de escena de la escena principal, creado con múltiples formas y transformaciones:



Cabe destacar que la versión completa del grafo tiene 5 nodos árbol y 2 nodos calle, los cuales tienen los mismos nodos hijos, pero tienen distintas transformaciones de traslación.

Para los demás elementos se usaron figuras simples con texturas, por lo que no están presentes en el grafo principal. Por otro lado, los círculos donde comienza el player, si bien no son texturas, se dibujaron con líneas, por lo cual no se agregaron al grafo de escena principal.

Instrucciones de Ejecución

Teniendo las librerías del curso instaladas y estando en la carpeta donde se encuentra `t1.py` y sus archivos relacionados, se ejecuta el archivo con la sintaxis `"t1.py Z H T P I V"`, siendo I y V parámetros opcionales. Como indica el enunciado de la tarea, Z y H indican, respectivamente, la cantidad de zombies y humanos que entran a la escena cada T segundos, P la probabilidad de que un humano contagiado se transforme en zombie, I la probabilidad de que un humano entre contagiado a escena, y V la velocidad de desplazamiento del player. Cabe destacar que tanto P como I deben ser valores entre 0 y 1, ambos incluidos.

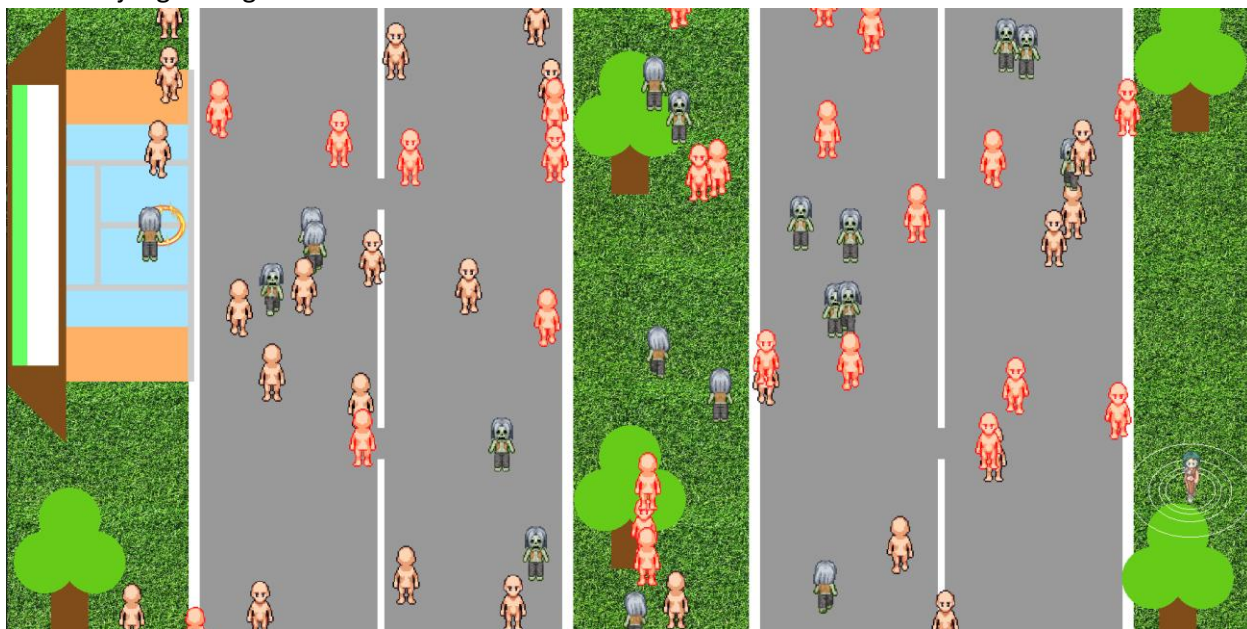
Una vez entra en ejecución el programa, el objeto player aparecerá en la zona derecha de la ventana, y se podrá desplazar con las teclas W (arriba), S (abajo), A (izquierda), D (derecha). La idea será llevar al player hasta la izquierda de la pantalla, donde se encuentra el objeto Aura girando en la puerta de la tienda. Con la barra espaciadora se podrá "ver a través de las gafas", para identificar a los humanos contagiados. Con la tecla 1 se podrá activar el modo "infectedMode" para reconocer cuando el player es infectado. Una vez que el player colisiona con el aura o pasa a ser zombie, el juego termina y se puede presionar la tecla enter para reiniciarlo. En cualquier momento se puede presionar la tecla escape para cerrar el juego.

Resultados

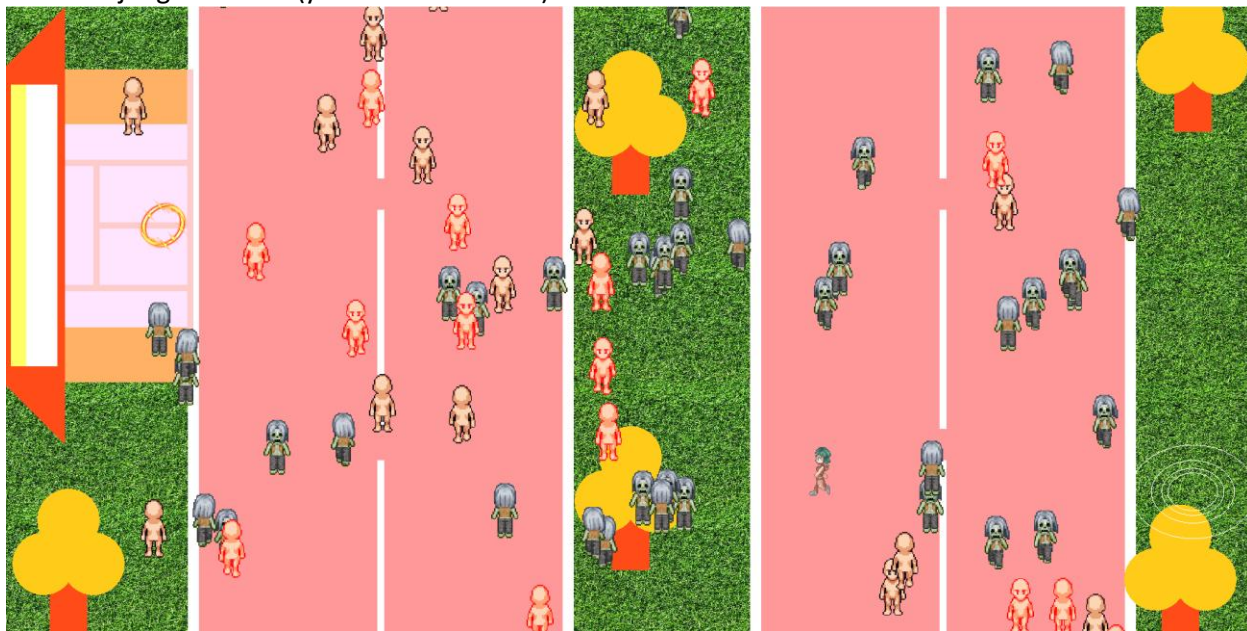
Modo normal de juego:



Modo de juego con gafas:



Modo de juego infected (y estando infectado):



Autoevaluación

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