

# Reporte Tarea 1: Beauchefville

Modelación y Computación Gráfica  
CC3501-1

Alumno: Felipe Morales  
Fecha: 10 de Mayo de 2021

## Introducción

Este documento presenta una revisión de los contenidos utilizados y el funcionamiento del programa para esta primera entrega del curso CC3501-1 Modelación y Computación Gráfica. Se entregarán los debidos argumentos para el uso de las distintas unidades del curso y , así, entregar una perspectiva de la implementación. Algunos supuestos son:

- Se considera a Jill de Resident Evil como personaje principal y no a Hinata pues no encuentre alguna textura que se acomodará al problema.
- La escena es estática en cuanto al que el fondo no se traslada.
- Existen modelos de poderes y vida pero no se implementó funcionalidad en estos por lo que están bloqueados por un obstáculo. Decisión hecha para modelar objetos más complejos y dar estética al juego.
- Se utilizan curvas de Bezier para modelar la trayectoria de los estudiantes y zombies.

## Desarrollo e implementación

Se decide hacer uso de Modelación Jerárquica dado que se busca construir una escena la cual presenta subconjuntos tales como fondo básico que contempla la calle, sus líneas y el pasto. Del mismo modo podemos tener otro subconjunto el cual presenta los elementos con texturas tales como los árboles, veredas y tienda. Por último tenemos aquel subconjunto de los personajes que contemplan a Jill, los estudiantes y los zombies.

El propósito de tener estos distintos subconjuntos es que se comportan de distintas formas en el estilo del juego por lo que se aprovecha el uso de grafos y las transformaciones por niveles.

El primer subconjunto consta de una profundidad de 4 niveles considerando las hojas. El segundo subconjunto presenta una profundidad de 7 niveles donde cabe resaltar la presencia de árboles los cuales tienen un nodo para transforma de shearing. En este mismo se encuentra también el grafo para realizar la vereda en su respectiva locación.

El grafo corresponde a autoría del estudiante con altura superior a 7 niveles con la posibilidad de extenderlo a un grafo más grande que contenga a cada subconjunto como hijos. Encontramos modelos con partes móviles como árboles o figuras que se mueven de forma estética. El conjunto de árboles presenta cierta complejidad así como un brazo robótico.

A continuación se presenta los distintos diagramas para cada escena con sus respectivas transformaciones:

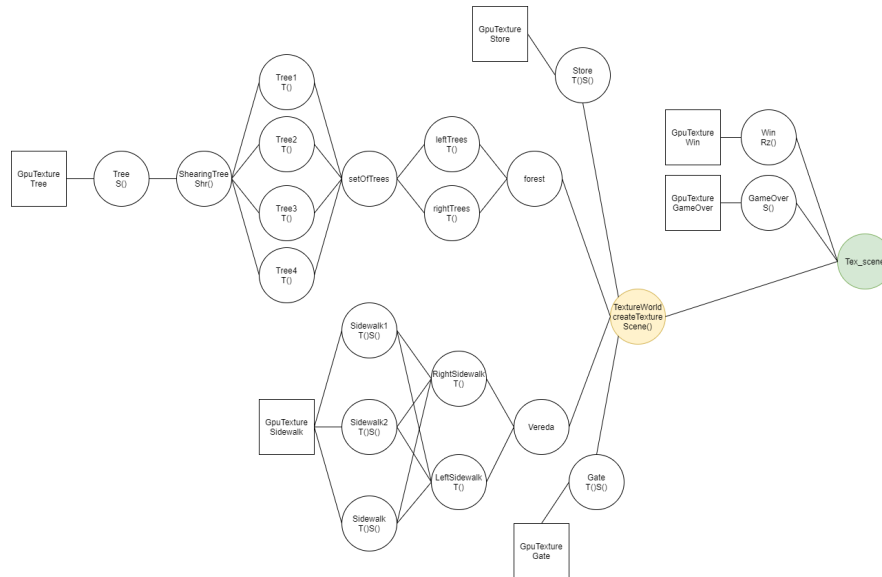


Imagen 1: Modelación jerárquica para escena de texturas

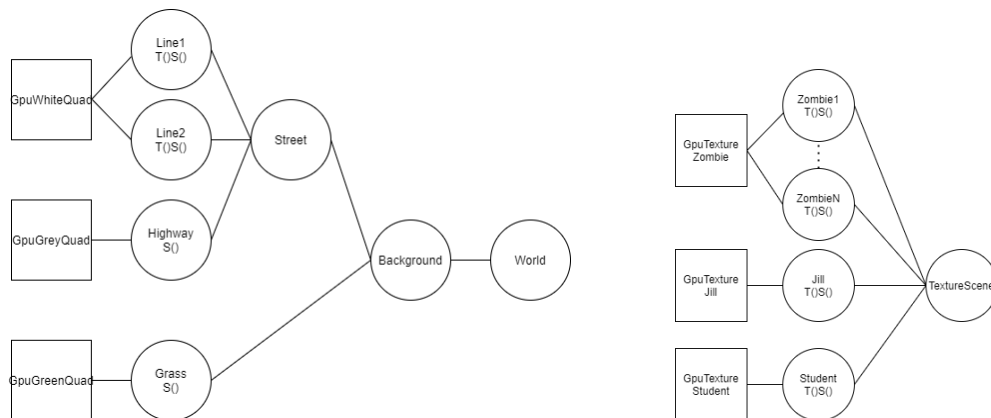


Imagen 2 y 3: MJ para escena con figuras geométricas y personajes del juego respectivamente

Para la sección de OPENGGL se utilizan VAOs, VBOS y EBOs eficientemente, reutilizando vértices, buffers y texturas cuando es posible. Dado que no presentamos algún caso de uso donde se modifiquen los vértices solo usamos GL\_STATIC.

Para la sección de shaders contamos con 4 de estos pues tenemos 4 funcionalidades distintas. En primer lugar usamos 2 shaders básicos, sin modificaciones, de transformaciones los cuales se encargan de los modelos con y sin texturas. Los restantes son de autoría propia y sirven para las transformaciones y el coloreado de las figuras con o sin texturas, sin embargo son simples. Uno es para ver quién está contagiado (los muestra verdes) y el otro hace titilar el color de los power ups (que no tienen funcionalidad en el juego).

En los modelos geométricos utilizamos una buena parte de cuadriláteros debido a que tienen un excelente manejo en ámbitos de esta tarea. Además, se incluyen 2 figuras con cierto grado de complejidad, las cuales corresponden a los power ups, donde se modelaron los vértices para realizar las respectivas figuras.

Respecto a las traslaciones, están presentes en gran parte de los modelos como los personajes o figuras pues tenemos que moverlas. A juicio propio se hace un uso correcto de traslaciones, escalamientos y rotaciones. Componiendo estas transformaciones correctamente. Además, se utilizan todas las transformaciones vistas en cátedra (las rotaciones respecto al eje X e Y no se utilizan dada las dimensiones del problema 2D pero si se utilizan las rotaciones en Z).

Se hizo uso de una gran cantidad de texturas sobre cuadriláteros, lo que me parece totalmente lógico dada las dimensiones y complejidad de la tarea, no encontré ningún motivo ni momento en el que fuera necesario usar otra figura para usar texturas. Se hace uso de mipmaps para evitar el anti-aliasing presentado al dibujar los árboles con GL\_LINEAR\_MIPMAP\_LINEAR con lo que se obtuvo un resultado más atractivo para la vista. Para repetir la textura en las veredas, evitando que la textura de aquella se estire verticalmente y se contraiga horizontalmente, se utiliza GL\_REPEAT obteniendo mejores resultados.

El objetivo base del enunciado se cumple logrando implementar todas las funcionalidades mecánicas y lógicas del juego. Así mismo, no existe problema para las entradas del usuario cumpliendo todas las funciones requeridas y se implementa la mecánica de correr. Por último, se considera que la información visual es completa y correcta para el contexto del problema.

## Instrucciones de ejecución

Este programa recibe 4 argumentos los cuales son: cantidad de zombies Z, cantidad de humanos H, personajes que entrarán a la escena cada T segundos y la probabilidad P de que alguien infectado se transforme en zombie (esta probabilidad debe ser un número en el intervalo [0,1]). Por lo que el programa se ejecuta en la terminal de la siguiente forma:

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

El control del juego consiste en que el usuario puede mover a Jill con las teclas WASD, puede correr con el shift izquierdo y puede hacer uso de los lentes especiales para ver a aquellos infectados manteniendo apretada la barra espaciadora.

No se hace uso de librerías adicionales mas las de el curso contenidas en la carpeta gráfica.

## Resultados

El juego consiste en llegar a la tienda del extremo tratando de evitar a los zombies y estudiantes infectados, si chocamos con alguno de ellos se pierde en el juego. Se construye una escena con árboles moviéndose simulando alguna corriente de aire. Se logran algunos power ups para agregar estética a la escena donde estos titilan como en otros juegos invitando al jugador a alcanzarlos, lamentablemente en esta versión son inalcanzables.

Todo aquel personaje infectado se verá verde bajo los lentes especiales que se usan con la barra espaciadora. Además, cada zombie presenta una trayectoria diferente en cada caso ya que se hizo curvas de bezier con puntos al azar respecto a ciertas coordenadas por cada instancia.

A continuación se visualizan distintas instancias del juego con las distintas funcionalidades.



Imagen 4: Instancias del juego

## Autoevaluación

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