

Reporte de Proyecto Grupal 01

Juego Tri-morph

Armando Hernández*, Mariana Hinojosa*, Yuridia Montelongo* y Norberto Avtitia*
Ingeniería en Tecnologías de la Información
Universidad Politécnica de Victoria

Resumen—En este documento se reporta la manera en que se llevó a cabo el desarrollo del juego Tri-morph que se encuentra disponible en la plataforma Android en su versión original. Este consiste en la caída de una cantidad indefinida de triángulos los cuáles poseen un tipo de comportamiento, color y forma de perder si chocan con el triángulo principal, el cual tiene libre movimiento en cada una de sus aristas y que es el principal objeto del juego. Con base a conocimientos generales de matemáticas y primitivas propias de la librería OpenGL en el lenguaje C++ se completó el juego exitosamente.

I. INTRODUCCIÓN

El objetivo principal de esta práctica es recrear el juego Tri-morph originalmente para dispositivos Android con el lenguaje C++ y las primitivas para graficación que ofrece la librería de OpenGL. Esto se logra entendiendo por completo cómo es que funciona el juego.

El juego Tri-morph es una aplicación de entretenimiento que simula la caída de un sinnúmero de triángulos a lo largo de toda una ventana o pantalla, cada triángulo que cae pertenece a una categoría, existen seis categorías. Cada categoría representa un comportamiento diferente, un color y una velocidad. Dependiendo de la categoría del triángulo será los bonus que se agregarán al score en la pantalla (algunas categorías no generan bonus). La razón de perder o ganar en este juego es que existe un triángulo con libertad de movimiento en sus aristas en donde el usuario tiene que desplazarlo a donde sea necesario para evitar perder en el juego. Cada triángulo que cae puede ser motivo de fracaso si es que este toca al triángulo principal, pero cada categoría tiene su punto de choque y no siempre es en algunos de los lados del triángulo, puede ser en su centro, en alguna arista, etc.

En las siguientes capturas de pantalla se muestra la vista del juego en su versión original, el cuál se intenta emular con las herramientas que se están abarcando en este curso. El inciso (a) muestra el menu principal el cuál no se abarcará en esta práctica pero que sin embargo se debe de dar una vista general del proyecto completo. El inciso (b) muestra la pantalla de inicio del juego que es un contador que decrementa para empezar a jugar y empiezan a caer los triángulos.

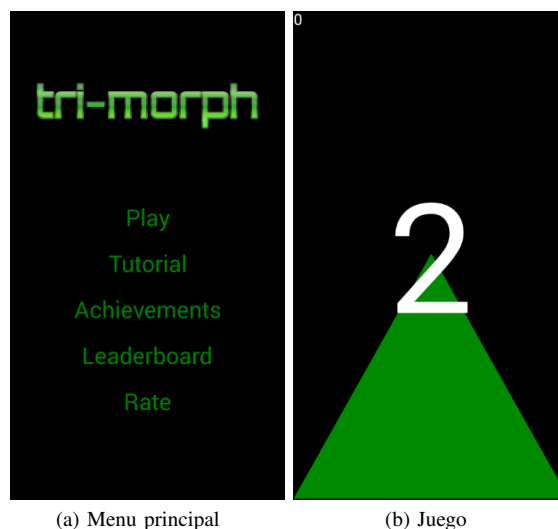


Figura 01. Capturas de pantalla del juego en su versión original.

En cuanto al desarrollo de la simulación se tuvo que indagar en una gran cantidad de temas relacionados a los comportamientos de los triángulos en graficación por computadora. Sabemos que un triángulo se utiliza para identificar un polígono compuesto por 3 lados. Esta figura geométrica se logra a partir de la unión de tres rectas que se interceptan en tres puntos desalineados. Cada uno de estos puntos donde las rectas se unen recibe el nombre de vértice, mientras que los segmentos que se pueden apreciar en la figura reciben el nombre de lados.[1]

Teniendo como punto de partida los conceptos bien definidos de nuestra figura principal que es el triángulo, se tomó en cuenta las partes con las que este es conformado que son sus aristas y vértices.

Una arista es el segmento de recta que limita la cara, también conocida como lado, de una figura plana. [2]

De esta manera nos lleva a definir el concepto de vértice que es el punto en el cual se encuentran las dos semirrectas de un ángulo, o el punto de intersección de dos lados de una figura plana.[3]

Por otra parte, se tiene que al producir un choque entre algún triángulo pequeño y el triángulo principal se obtiene una colisión. Una colisión es un fenómeno físico que implica que si dos elementos son arrojados a una velocidad x en un

mismo espacio, los mismos colisionarán o chocarán de manera violenta porque no puede darse que ambos ocupen el mismo espacio al mismo tiempo.

La colisión puede darse con dos objetos en movimiento como también cuando uno de los dos objetos está en movimiento y el otro no (por ejemplo, cuando un auto se estrella o colisiona contra una pared o contra otro automóvil que se encuentra parado y estacionado). Siempre la colisión generará algún tipo de daño o de alteración en la estructura de los elementos que chocan. [4]

Al obtener un choque entre ambas figuras se termina el juego y por lo tanto todo tipo de movimiento en la pantalla se detiene.

II. DESARROLLO EXPERIMENTAL

Para poder desarrollar de forma práctica este proyecto se indagó en diferentes tópicos necesarios para la elaboración de una solución optimizada y objetiva para este problema. De este manera y después de analizar cada uno de los retos que este juego tenía para poder ser resuelto, se inició el proceso de estructurar la manera lógica del algoritmo en el lenguaje C++.

Se empezó por la parte principal del juego que es el triángulo principal el que será sometido a cambios durante toda la ejecución ya que sus vértices son la interacción principal de el dispositivo con el usuario. Para esto, se llegó a la idea de controlar el movimiento del mouse mediante una función en la cual se van actualizando las coordenadas del vértice que el usuario haya seleccionado guardadas en un array dinámico de tipo objeto de la clase llamada "Point".

Al tener el movimiento del triángulo verde se decidió realizar una clase llamada "Triangle", ya que en el juego existen diferentes triángulos pequeños cayendo con diferentes comportamientos, colores y velocidades, por lo que la mejor solución a este reto fue el crear esta clase. Mediante una función "Timer" controlamos el que vayan cayendo esos triángulos calculando sus nuevas posiciones en y actualizandolas en su objeto correspondiente y de esta manera mandarlas a re-dibujar. Todo funciona de manera automática gracias a esta función. En otra función "Timer" se controla el color de cada uno de los triángulos pequeños y las velocidades en las que cae cada uno.

Uno de los mayores problemas que se tuvo en el desarrollo de la aplicación fue que los triángulos pequeños al tocar triángulo principal tienen diferentes comportamientos, ya sea que se termina el juego, se toman puntos extras en la puntuación por lo que necesitábamos encontrar una manera de saber cuando tocaban al triángulo principal por lo que después de realizar varias investigaciones se optó que la mejor manera de conseguir lo antes mencionado era utilizando la fórmula de la ecuación de la recta y de esta manera cada que se actualizan las coordenadas de los triángulos pequeños se está verificando si pertenece a cualquiera de las tres aristas del triángulo principal, al saber que pertenece a cualquiera de las rectas se da por hecho que el triángulo pequeño ya tocó el triángulo principal.[5]

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$y - y_1 = m(x - x_1)$$

Figura 02. Ecuación de la recta.

Después de saber que el triángulo principal ha sido tocado por alguno de los triángulos pequeños se realiza las operaciones necesarias en la puntuación dependiendo del comportamiento del triángulo que haya colisionado de manera que al finalizar el juego se realiza una condición, la cual va de la mano del comportamiento de los triángulos que finalizan el juego al tener una colisión con el triángulo principal.

Al perder en el juego en algún choque entre un triángulo pequeño y el triángulo principal se muestra la puntuación obtenida en todo el tiempo que el usuario pudo mantener el juego en ejecución.

III. RESULTADOS

En el transcurso de esta práctica se fueron generando una serie de evidencias que comprueban el desarrollo de esta práctica y que a su vez permiten dar a conocer cómo se visualiza el algoritmo puesto en práctica de todo el juego.

En la Figura 05 se visualiza el diagrama de flujo general de las actividades que realiza el algoritmo para poder ejecutar el juego satisfactoriamente.

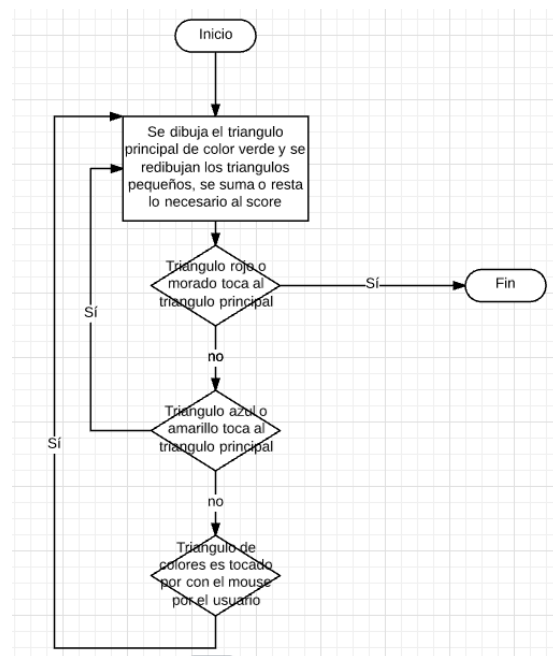


Figura 03. Diagrama de flujo del programa

Gracias a este diagrama nos podemos dar cuenta de la secuencia que sigue la ejecución del programa.

En la Figura 04 se muestra las instrucciones más básicas de la aplicación para darle una idea de lo que se trata.

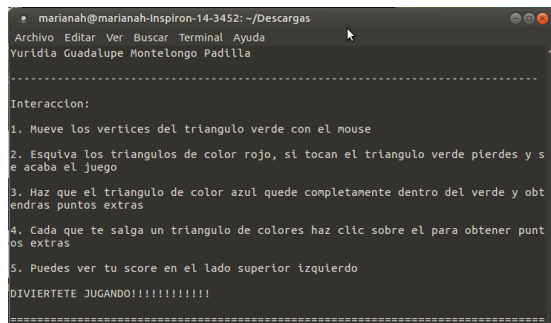


Figura 04. Instrucciones del juego.

En la Figura 05 y 06 se muestran evidencias gráficas de lo que es el juego y todos los elementos que cumplen con una función fundamental en el cumplimiento de su objetivo.

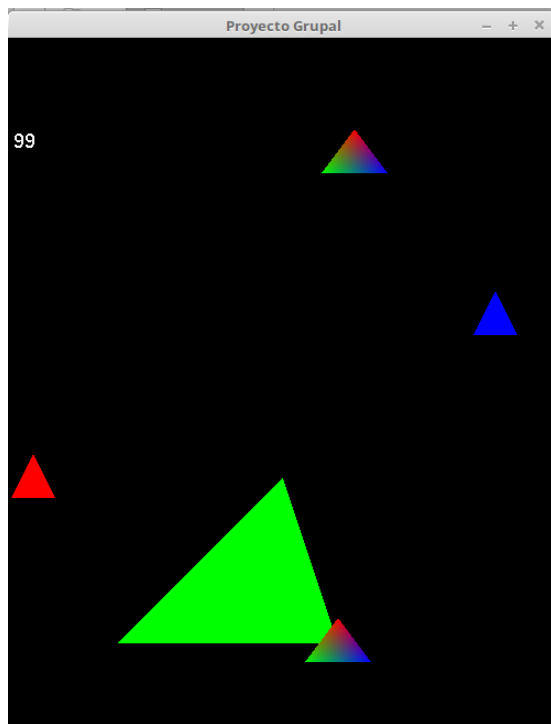


Figura 05. Screenshot de la aplicación.

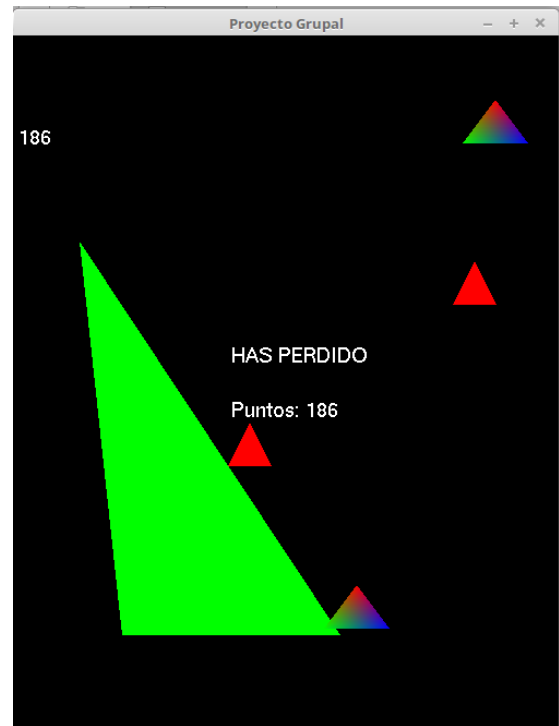


Figura 06. Screenshot de la aplicación.

La aplicación ofrece información completa y la funcionalidad principal que se pretendía cumplir con el desarrollo de este proyecto.

IV. CONCLUSIÓN

Podemos concluir que la aplicación realizada cumple con todos los objetivos planteados desde un principio, la cual esta basada en el juego existente de llamado "Tri-Morph", desde mover el triangulo principal, hasta cada uno de los comportamientos de cada triangulo pequeño que cae. Y dejando en claro que todo se realizo solo mediante el uso de librerias OpenGL, proporcionadas anteriormente por nuestro catedratico.

V. BIBLIOGRAFÍA

- [1] Julián Pérez Porto y Ana Gardey. Triángulo. 2010. <https://definicion.de/triangulo/> Consultado el 19 de Octubre del 2017.
- [2] Wikipedia. Arista (geometría). 2017. [https://es.wikipedia.org/wiki/Arista_\(geometr\)](https://es.wikipedia.org/wiki/Arista_(geometr)) Consultado el 19 de Octubre del 2017.
- [3] Diccionario matemático. Vértice. 2017. <http://www.mathematicsdictionary.com/spanish/vmd/full/v/vertexinplanegeometry.htm> Consultado el 19 de Octubre del 2017.
- [4] Definición ABC. Definición de colisión. 2007. <https://www.definicionabc.com/general/colision.php> Consultado el 20 de Octubre del 2017.
- [5] Profesor en Línea. Ecuación de la recta. 2015. http://www.profesorenlinea.com.mx/geometria/Recta_Ecuacion_de.html Consultado el 21 de Octubre del 2017.