

Informe de proyecto GeoWalle

Raciel Simón Domenech Jean Carlo García Wong Diciembre,2023



1. Introducción

En este proyecto de programación llamado GeoWalle hemos adquirido diversos conocimientos sobre varias ramas de la programación en el lenguaje, por ejemplo: la compilación y el diseño de aplicaciones gráficas, aristas que nos son de vital importancia para tener un mejor desenvolvimiento en la carrera universitaria.

GeoWalle es de una gran importancia para nuestro desarrollo como programadores porque con él hemos podido ver cuales son los buenos hábitos que todo coder debe tener a la hora de realizar sus proyectos, nos ha servido como otra materia de estudio donde hemos aprendido mucho sobre los algoritmos utilizados en él y ha aumentado el amor hacia la carrera Ciencias de la Computación.

2. Requisitos para la ejecución

- Net 6.0
- Godot 4.1.1

3. Desarrollo

De forma general nuestro proyecto de programación que se nombra GeoWalle como ya habíamos mencionado anteriormente consta de dos partes, una llamada GeoWalle que contiene el motor gráfico de la aplicación y otra llamada Compiler que contiene el lenguaje de programación que utiliza la aplicación gráfica para implementar todas las funcionalidades que a continuación explicaremos.

3.1. GeoWalle

Esta parte del proyecto no es más que el motor gráfico del mismo como mencionamos anteriormente que para la realización de este usamos la plataforma Godot la cual nos facilitaba todas las cuestiones de pintar gráficos por todas las herramientas con las que cuenta la misma y además que funciona con el lenguaje de programación que aprendimos en este primer año de la carrera.

Esta aplicación visual consta de dos superclases llamadas CodeEditorScript y OutputConsole, la primera es un editor de código que de manera general es para ingresar código para después ser dibujado en pantalla. La segunda clase es una consola mediante la cual mostraremos errores en caso de haber o información útil para el usuario.

En el archivo CodeEditorScript.cs se implementan varias clases llamadas CodeEditorScript, Arc y Circle; estas dos últimas dan lugar a las herramientas matemáticas que pueden ser usadas en este proyecto.

La clase Circle que hace referencia a la circunferencia como es conocida matemáticamente consta de las propiedades tipo float centerx y center y que dan lugar a las coordenadas del centro de la circunferencia que sería un punto en el plano, una propiedad tipo float llamada radius que representa el radio de la misma y una última llamada segments utilizada para darle valor a los segmentos formado por los arcos de la circunferencia. Esta clase además consta con un método llamado Draw que me devuelve un array de tipo Line2D (viene implementada en Godot), método que se encarga de pintar esta figura geométrica.

La clase Arc hace referencia al arco de la circunferencia consta de las siguientes propiedades, centerx y centery de tipo float la cual hace alusión al centro de la circunferencia que le corresponde al arco sobre el cual se está trabajando, radius de tipo float que sería el radio de la circunferencia correspondiente a dicho arco, una propiedad de tipo int llamada segments y dos propiedades tipo float llamadas startangle y endangle que indicarían las posiciones de apertura y cierre que me determinan la amplitud del ángulo que forman los segmentos que originan al arco. Al igual que la clase anterior esta consta de un método Draw que devuelve un array de tipo Line2D que se va a encargar de pintar esta representación geométrica en el visual.

Por último y no por menos importante la clase CodeEditorScript que engloba a las dos explicadas anteriormente que no es más que el editor de código que tiene como propiedades una instancia de la clase OutputConsole que sería la consola correspondiente al editor actual, una instancia de la clase implementada por Godot Sprite2D llamada plane, una variable tipo bool CanPressF5 inicializada en true para saber si se puede ejecutar el código ingresado por el usuario y una variable tipo string llamada valuex con valor "null". Esta clase implementa el método con nombre Ready que es el encargado de ejecutar las

funcionalidades al inicializar el programa. Además se implementa el método `CalculateLinePoints` que se encarga de calcular el segmento representado por dos puntos y el método `CalculateRayPoints` que calcula el rayo formado por un punto hacia el infinito.

3.2. Compiler

Esta segunda parte del proyecto es la que contiene el lenguaje de programación y el proceso de análisis léxico, sintáctico y semántico de las líneas de código ingresadas por el usuario en la aplicación visual. Este compilador consta de varias clases que entre las más importantes se encuentran: `Tokens`, `Node`, `Parser` y `Error`.

La clase `Tokens` es la que almacena todo el lenguaje de programación implementado para este proyecto. La superclase `Node` consta de varias clases englobadas dentro de ella que nos facilitan el trabajo con las figuras geométricas en la parte gráfica y para de alguna forma almacenar la información que ingresa el usuario en el input de manera organizada y catalogada matemáticamente.

La clase `Error` la usamos únicamente para saber que tipo de error ocurrió a la hora de compilar el código ingresado por el usuario ya sea léxico, sintáctico o semántico para después mostrárselo al usuario en la aplicación visual.

La superclase `Parser` contiene una buena cantidad de métodos en ella que de manera general es la que se encarga de procesar el código que ingreso el usuario, interpretarlo y después ejecutarlo y en caso de existir algún error semántico se lanzan excepciones para advertir al usuario de que su código no está bien escrito.

4. Conclusiones

Este proyecto ha sido de gran utilidad como aprendizaje, nos ayudó a mejorar nuestro enfoque hacia la programación y a saber la realidad de como es el día a día de un programador. Además aprendí un montón de malos hábitos que deben ser evadidos por cualquier persona que tenga conocimiento de la programación. Espero que haya sido de utilidad este informe.

Índice

1. Introducción	1
2. Requisitos para la ejecución	1
3. Desarrollo	2
3.1. GeoWalleE	2
3.2. Compiler	3
4. Conclusiones	4