A picture containing text, porcelain

Description automatically generated

**Universidad Nacional Autónoma de México.**

**Facultad de Ingeniería.**

**Laboratorio de Computación Gráfica e interacción Humano Computadora.**

**Nombre del alumno: Cervantes Rubí Brandon.**

**Número de Cuenta: 316136741.**

**Grupo de Laboratorio: 01.**

**Sistema de control versiones.**

**Semestre 2022-1.**

**Fecha de entrega límite: 22/11/2022.**

Graphical user interface, application

Description automatically generatedEn este proyecto decidí implementar las herramientas de Git para trabajar localmente y GitHub para subir las versiones a un repositorio remoto. Lo primero que se hizo fue crear un repositorio remoto con mi cuenta de Github el cual se clono localmente con el comando git clone y el enlace HTTPS que ofrece la plataforma. Hasta este momento no hubo problemas y se tenia ya el repositorio local vinculado al remoto.

Si bien no hice el repositorio hasta tener un avance mínimo en el código, que fue alrededor del 5to día de primera semana, se tuvieron problemas para poder subir el repositorio a la página de GitHub relacionados con la cuota máxima que podía soportar el buffer con el que trabaja para subir los archivos por lo cual daba un error al momento de intentar subir los archivos.

Para solucionarlo lo primero que hice fue intentar subir parte por parte, puesto que eran alrededor de 500 archivos contando el código base del proyecto con las librerías include, glfw, lib, etc. Con esta técnica subí con éxito las primeras carpetas, pero eventualmente me desesperé. Trate también agregando un archivo .gitignore para ignorar extensiones de los archivos que no necesitaba que se subieran, pero me tope con que la mayoría de los archivos eran código en C++ y no podía ignorar todo este código porque si lo hacía iba a terminar sin subir nada, ya muy desesperado decidí eliminar los últimos dos commits que manchaban un poco el repositorio lo cual fue un error ya que me lleve la carpeta del proyecto volviendo al repositorio que clone vacío. En esos momentos casi se me sale el corazón, pero afortunadamente tenía un respaldo del código que se había trabajado el día anterior, lástima que no pudiese decir lo mismo para los archivos editados de modelos en blender que ya se habían trabajado.

El último intento consistió en incrementar el buffer con el que trabaja git push para subir todos los archivos y haciendo esto pude subir todos los archivos en un solo intento. Ya con todo en orden me ayude de una extensión GitHub para visual estudio code para ir subiendo el código la cual permite hacer los comandos que componen todo el flujo de trabajo de git; git add, git commit, git pull, git push, etc. de forma grafica seleccionando los archivos que se van a subir presionando algún icono, agregando el comentario en un recuadro y de más elementos visuales. Además de que la complemente con otra extensión llamada gitGraph que muestra la línea del tiempo de todas las versiones que se han subido. Como solo trabaje en una rama, la línea del tiempo luce de forma lineal. Este resumen del desarrollo muestra ademas el numero de archivos con cambios que no se han subido, la fecha en la que se hizo, el usuario que lo hizo y el id del commit por si se necesita en alguna operacion para regresar entre las versiones por ejemplo.

Text

Description automatically generated

Esta línea del tiempo también la podemos ver en el portal de github pero con un formato diferente, un poco menos legible que en la extensión comentada pero ayuda bastante para darse una idea de cómo está constituido el proyecto. Algo que me llamo la atención es que además de esta línea del tiempo de desarrollo, en el apartado de Insights se pueden ver las estadísticas del proyecto pudiendo cambiar el periodo de tiempo entre días, meses y semanas.

Graphical user interface, application

Description automatically generated

En este caso fui el único contribuidor, pero si se trabajara en equipo se podrían ver los demás en la gráfica de barras, así como la cantidad de commits que han hecho al proyecto lo cual considero que es una buena adición para fomentar el desarrollo. Otra de las estadísticas que me parecieron interesantes es la de los commits, los cuales se pueden ver una grafica que se selecciona por semana y puedes ver la cantidad de commits que se hicieron en esa semana. A continuación, coloco los commits de la ultima semana la cual fue la mas constante en cuanto a trabajo del proyecto. Y por otro lado ya en la implementación me ayudó mucho la herramienta para comparar cambios para ver si se subía a la versión estable del repositorio o si aún no estaba terminado, por ejemplo, y así que se quedara hasta completarlo.

Graphical user interface, chart, line chart

Description automatically generated

Graphical user interface

Description automatically generated

En conclusión, trabajar con esta herramienta de control de versiones trajo sus ventajas las cuales son el poder llevar un avance controlado del proyecto con la tranquilidad de que si de repente deja de funcionar en alguna versión se pueda regresar a una versión estable y comparar para ver cual fue el problema, aunque se sebe de manejar con cuidado el regresar entre las versiones o de plano eliminar commits ya que se podría llegar a perder el avance de las cosas.

*Enlace al repositorio:* [*https://github.com/Slevkro/Pachanga-Time*](https://github.com/Slevkro/Pachanga-Time)