



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.

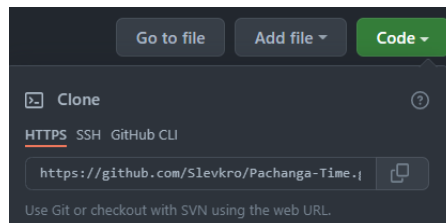
Grupo de Laboratorio: 01.

Sistema de control versiones.

Semestre 2022-1.

Fecha de entrega límite: 22/11/2022.

En 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 tenía ya el repositorio local vinculado al remoto.



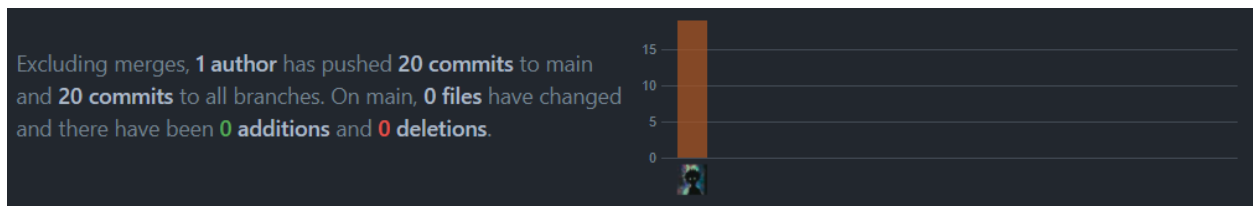
Si bien no hice el repositorio hasta tener un avance mínimo en el código, que fue alrededor del 5to día de primera semana, se tuvieron problemas para poder subir el repositorio a la página de GitHub relacionados con la cuota máxima que podía soportar el buffer con el que trabaja para subir los archivos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

Graph	Description	Date	Author	Commit
Uncommitted Changes (12)		22 Nov 2021 00:53	*	*
<code>main</code> <code>origin</code> <code>origin/HEAD</code>	Animacion del avatar y manuales/Bitacoras	22 Nov 2021 00:52	Shoto	1f595216
	Animacion del avatar	21 Nov 2021 14:53	Shoto	536d7b0d
	Jerarquia del avatar y su material	21 Nov 2021 01:43	Shoto	#122807
	Animaciones de baile, muñecos de nieve y detalles	20 Nov 2021 21:31	Shoto	3e21b121
	Guardado de camaras, arreglando luces puntuales	20 Nov 2021 17:23	Shoto	b4f06db5
	Show de luces y decoracion del piso	20 Nov 2021 02:04	Shoto	8c0abfd9
	Camaras al piso, areras y libre	18 Nov 2021 03:34	Shoto	23298b85
	Luces puntuales y direccional para ciclo dia-noche	17 Nov 2021 02:40	Shoto	d75d4cfa
	Animacion de Marceline y la roca del easter egg	16 Nov 2021 01:18	Shoto	74354db3
	Animacion del personaje sobre el agua y las nubes	15 Nov 2021 02:23	Shoto	ac9b78ff
	Separando objetos de animacion y luces puntuales	11 Nov 2021 01:19	Shoto	6b4a19d0
	Carga de librerias parte 3	9 Nov 2021 01:10	Shoto	e425bc78
	Carga de librerias parte 2	9 Nov 2021 01:10	Shoto	35e2c79e
	Carga inicial de librerias	9 Nov 2021 01:07	Shoto	7edac95f
	Carga de modelos y texturas	9 Nov 2021 00:14	Shoto	8c10fe05
	Carga del codigo hasta el skybox	9 Nov 2021 00:09	Shoto	e1b50451
	Agregando modelo de Shulba y inicio de bitacoras	8 Nov 2021 23:21	Shoto	3d5459d6
	Agregando gitignore	8 Nov 2021 00:58	Shoto	d4e0f620
	Carga de modelos y documentos iniciales	8 Nov 2021 00:41	Shoto	f7ceeb82
	Initial commit	5 Nov 2021 22:36	Brandon Cervantes	40d4cda7

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.



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.



```

Codigo > Pachanga time > Pachanga time > C++ Shader_light.cpp > Shader()
193 }
194 GLuint Shader::GetEyePositionLocation()
195 {
196     return uniformEyePosition;
197 }
198
201+ GLuint Shader::GetColorLocation()
202+ {
203+     return uniformColor;
204+ }
205+
199 void Shader::SetDirectionallight(DirectionalLight * dLi
206 void Shader::SetDirectionallight(DirectionalLight * dLi

```

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 debe 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>