

Git Básico

Outer games Studios 

Introducción

Dentro de esta guía nos enfocaremos en las funciones básicas de Git. No vamos a utilizar GitHub por el momento, sino que nos enfocaremos en el desarrollo local.

Estructura

Como se mencionó en la introducción, Git utiliza un directorio mágico llamado `.git/`. Este directorio vive dentro de nuestro proyecto/carpeta.

```
my_project/  
├─ assets/  
│   ├─ sprites/  
│   │   └─ char.jpg  
│   └─ music/  
├─ scripts/  
│   └─ char.gd  
└─ .git/
```

Supongamos que este es mi proyecto. El folder principal es `my_project`. Dentro de este folder tenemos varias cosas, como `assets` y `scripts`. Además de todo eso, tenemos un directorio especial, el directorio `.git/`. Este directorio es lo que permite a Git tomar esos *screenshots* de nuestro proyecto.

Estructura

Ahora vamos a definir algo. A cualquier folder que contenga este directorio `.git/` lo vamos a llamar un **repositorio**, o repo. Entonces cualquier proyecto que tengamos con `.git/` también lo podemos llamar como un repositorio.

Y otra cosa importante. Este repositorio está dentro de nuestra computadora, nadie más lo puede ver. Por eso decimos que es **local**, que solo está en nuestra computadora.

Entonces, a cualquier folder con un directorio `.git/` y que esté dentro de nuestra computadora lo llamamos un **repositorio local**.

Crear un repositorio local

Entonces, ¿cómo creo este repositorio local? Primero debes crear una carpeta nueva. Esa carpeta nueva se volverá nuestro repositorio. Ahora debes acceder a esa carpeta mediante la terminal. En el explorador de archivos puedes abrir seleccionar abrir una carpeta en la terminal. Puedes investigar información respecto a esto.

Una vez que estes dentro de la carpeta en la terminal, ejecuta el siguiente comando.

```
git init
```

Este comando simplemente crea el directorio `.git/` dentro de la carpeta en la que te encuentras.

Crear un repositorio

Con esto, ya tienes un repositorio creado. Recuerda, un repositorio es una carpeta con el directorio `.git/`.

Ahora podemos pasar a explorar las funcionalidades de Git.

Flujo

Simplificando un poco, Git sigue el siguiente flujo.

- Tienes cierto archivo que creaste, moviste, o modificaste dentro del repo.
- Preparas ese archivo. En inglés encuentras esto como *staging a file*.
- Finalmente, le dices a Git que tome un screenshot en el tiempo.

Esto es lo que vamos a aprender a hacer dentro de esta guía.

Modificar

Este es el paso más sencillo. Simplemente debes modificar algo dentro del repositorio. Esto quiere decir, añadir un archivo, crear un archivo, eliminar un archivo, modificar un archivo, etc. Todos estos son cambios que Git puede trackear.

Si estas siguiendo esta guía, inténtalo. Crea un archivo (pueden ser un script de Python), y agrega algo.

Staging

El siguiente paso es preparar los archivos. En el paso anterior se hicieron modificaciones. Este paso se basa en decirle a Git a que quieres que le tome *screenshot*. No tiene sentido tomarle foto a algo que no cambió, por lo que solo le dices a Git que prepare lo que cambiaste. Esto se hace con el siguiente comando:

```
git add [file-name]
```

También puedes hacerlo para varios archivos, como `git add file_1.py README.md data.csv`.

Como es muy común querer preparar todos los archivos que cambiaron, puedes utilizar este comando:

```
git add .
```

Staging

Si estás siguiendo esta guía, también intentalo. Ejecuta cualquiera de los comandos para preparar todos los archivos que acabas de añadir/modificar al repo.

Commit

Este es el último paso (en un repositorio local). Ahora debemos de decirle a Git que le tome un screenshot a estos cambios. Para esto utilizamos el comando:

```
git commit -m "message"
```

En la parte de `message` pones un comentario que diga que es lo que hace tu cambio. Es buena práctica poner mensajes descriptivos pero concisos. Simplemente dí lo que tu cambio hace.

Si estás siguiendo la guía, intentalo.

Flujo

Volvamos a ver el flujo. El flujo es el siguiente: `modify > stage/add > commit`. Primero modificamos todo lo que tengamos que cambiar. Luego preparamos los archivos y finalmente tomamos el screenshot.

Aunque no sea obvio, Git ya esta trackeando todos estos archivos y esta creando diferentes versiones.

Recap

- Un repositorio contiene un directorio `.git/`
- Un repositorio es local si está en mi computadora
- Un repositorio se inicializa con `git init`
- Primero modifico algo
- Segundo preparo archivos con `git add .`
- Tercero tomo screenshot con `git commit -m "message"`

¡Gracias!

Siguiente parte: GitHub Básico