Acá te van unos comandos de git fáciles para que consigas lo que necesitas.

Ya que configuraste el repositorio en tu escritorio y lo subiste al remoto en GitHub, querrás ir a tu laptop, abrir Git Bash, PowerShell o la herramienta de línea de comandos que prefieras.

Navega a la página de tu repositorio en GitHub y copia la ruta HTML o SSH haciendo clic en el botón "Code" arriba a la derecha de la página del repositorio. En la terminal, escribe lo siguiente, reemplazando la parte entre corchetas con la ruta HTML o SSH:

git clone <http/ssh path>

Después de esperar un poco, la terminal debería reportar que el repositorio se clonó correctamente desde el remoto. Ten en cuenta que puede que tengas que asegurarte de haber iniciado sesión en GitHub en tu laptop para que esto funcione.

Una vez que el repositorio esté clonado, usa cd para ir al directorio recién creado. Deberías estar en la rama principal por defecto, a menos que hayas hecho algo raro en la configuración de tu repositorio. Desde aquí, puedes empezar a crear nuevas ramas locales feature, bugfix, etc. para mantener tus ramas organizadas, así:

git checkout -b feature/some-new-feature

Después de hacer algunos cambios en scripts, escenas, prefabs, etc., puedes hacer lo siguiente para preparar los cambios nuevos para el commit, y luego hacer el commit con un mensaje:

git add . <-- prepara todos los cambios para el commit

git commit -m "your commit message here" <-- crea tu mensaje de commit

(Nota: Necesitas las comillas en la línea del commit).

Espera, ¡esto no significa que ya esté en tu repositorio remoto! El commit sigue siendo local, así que para subirlo al remoto, haz lo siguiente:

git push -u origin feature/some-new-feature

Ese comando subirá tu rama al repositorio remoto con seguimiento ascendente.

Con todo eso fuera del camino, vuelve a tu computadora de escritorio. Abre la herramienta de línea de comandos que prefieras en la carpeta de tu repositorio local y escribe lo siguiente:

git fetch <-- Revisa el repositorio remoto para ver si hay cambios nuevos

git switch feature/some-new-feature <-- cambia a la rama con seguimiento ascendente

De aquí en adelante, en la computadora que estés trabajando, deberías empezar tu trabajo con los siguientes comandos:

git fetch

git pull

Y terminar tu sesión de trabajo con estos comandos:

git add .

git commit -m "new commit message"

Esto mantendrá tu trabajo actualizado en el repositorio remoto y te permitirá obtener cambios descendentes de una computadora u otra. Además, ten en cuenta que suelo usar a lo loco el comando git status para revisar las cosas antes de hacer casi cualquier cosa.

Esta guía es bastante simple y no es una guía completa sobre cómo usar git. Si realmente quieres ser un crack con git, te sugiero que leas la guía de estrategia de ramificación GitFlow de GitHub y revises La Parábola de Git.

Familiarizarte con git es una excelente decisión. Es un estándar muy común en la industria y también te ayudará a familiarizarte con las técnicas generales de control de versiones. Si trabajas en equipo o con una empresa en un proyecto, necesitarás tener cierta familiaridad con git. Te puedo decir por experiencia que es mejor aprenderlo por tu cuenta que en el trabajo.