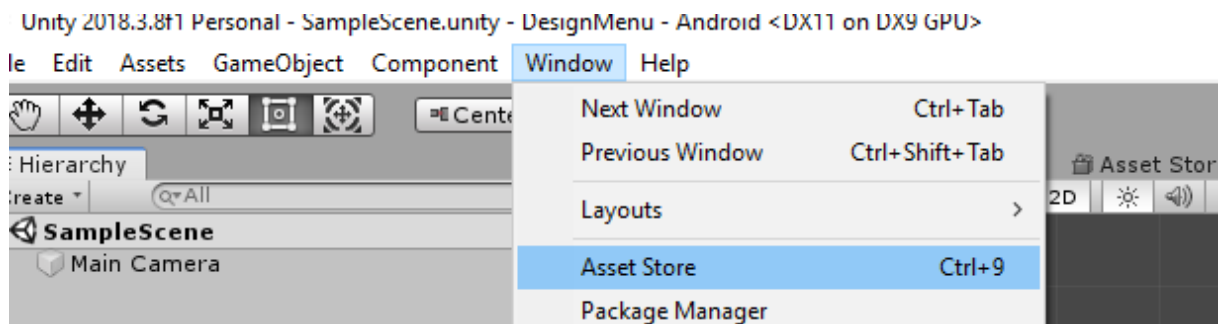
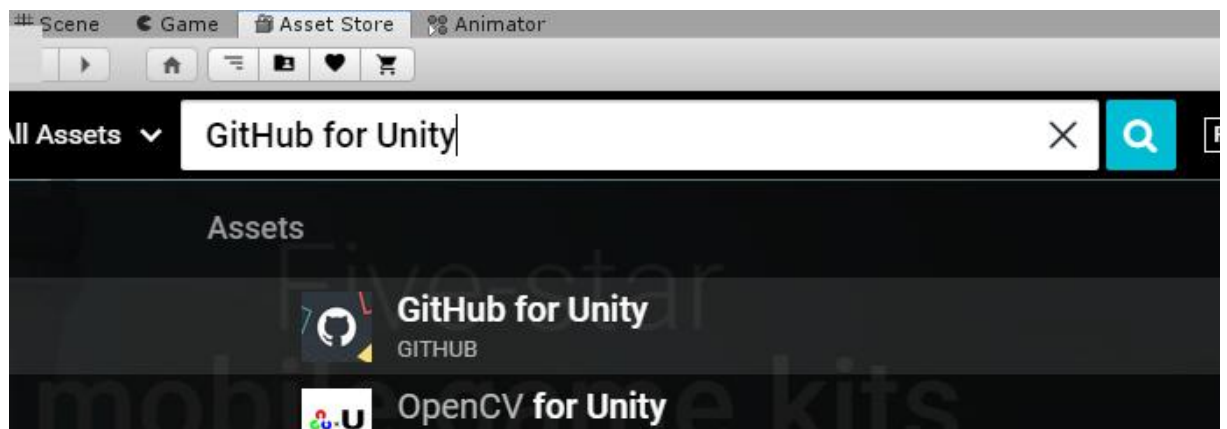


## GitHub for Unity – Schritt für Schritt

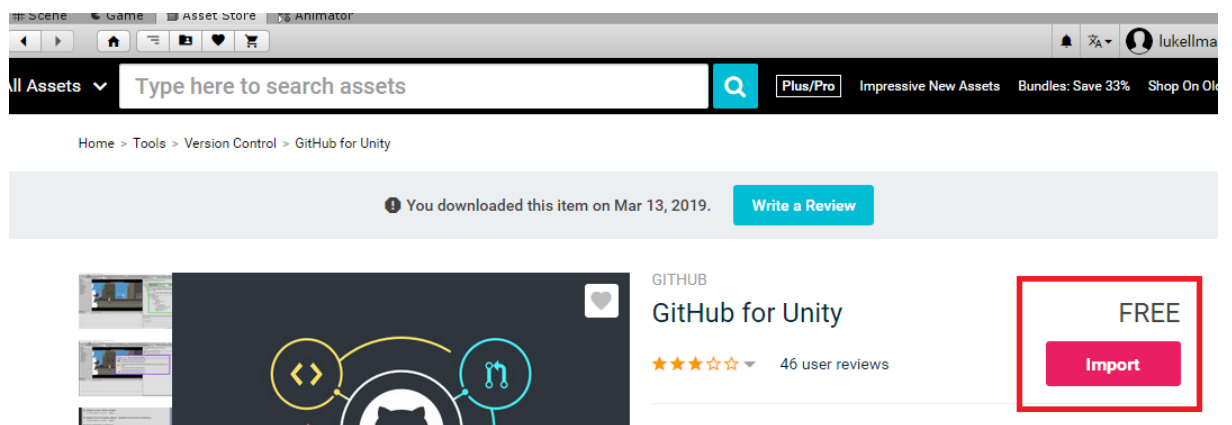
1. Ihr erstellt ein neues Unity-Projekt (2D-Template) (ihr könnt noch unter File > Build Settings die Platform auf Android switchen (dann könnt ihr auch Unity Remote nutzen)).
2. Ihr klickt unter Window > AssetStore an



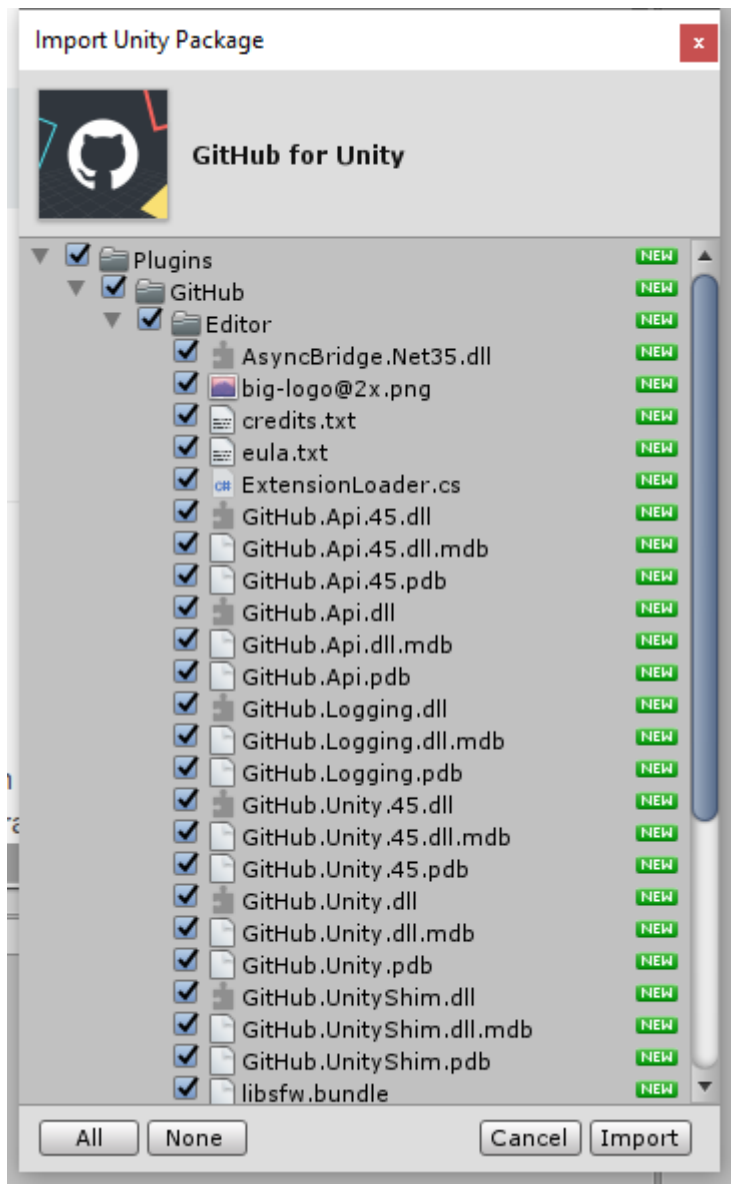
und sucht nach „GitHub for Unity“.



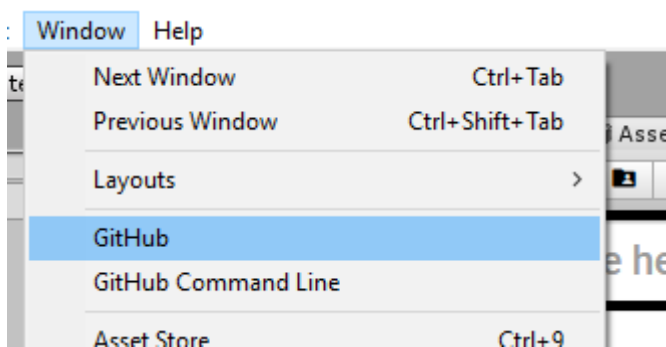
3. Jetzt hier auf Download (hab ich schon gemacht) und dann Import klicken.



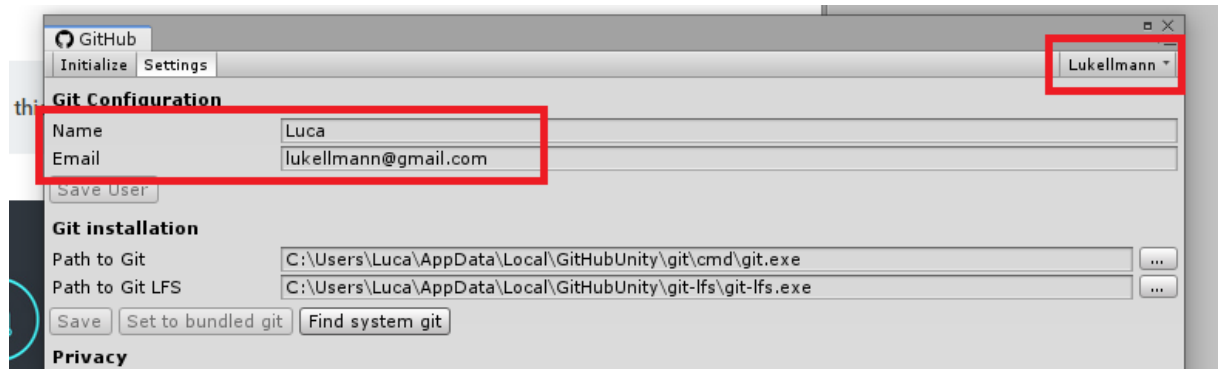
In dem Dialog, der sich jetzt öffnet alles auswählen und importieren.



4. Jetzt unter Window > GitHub GitHub öffnen.



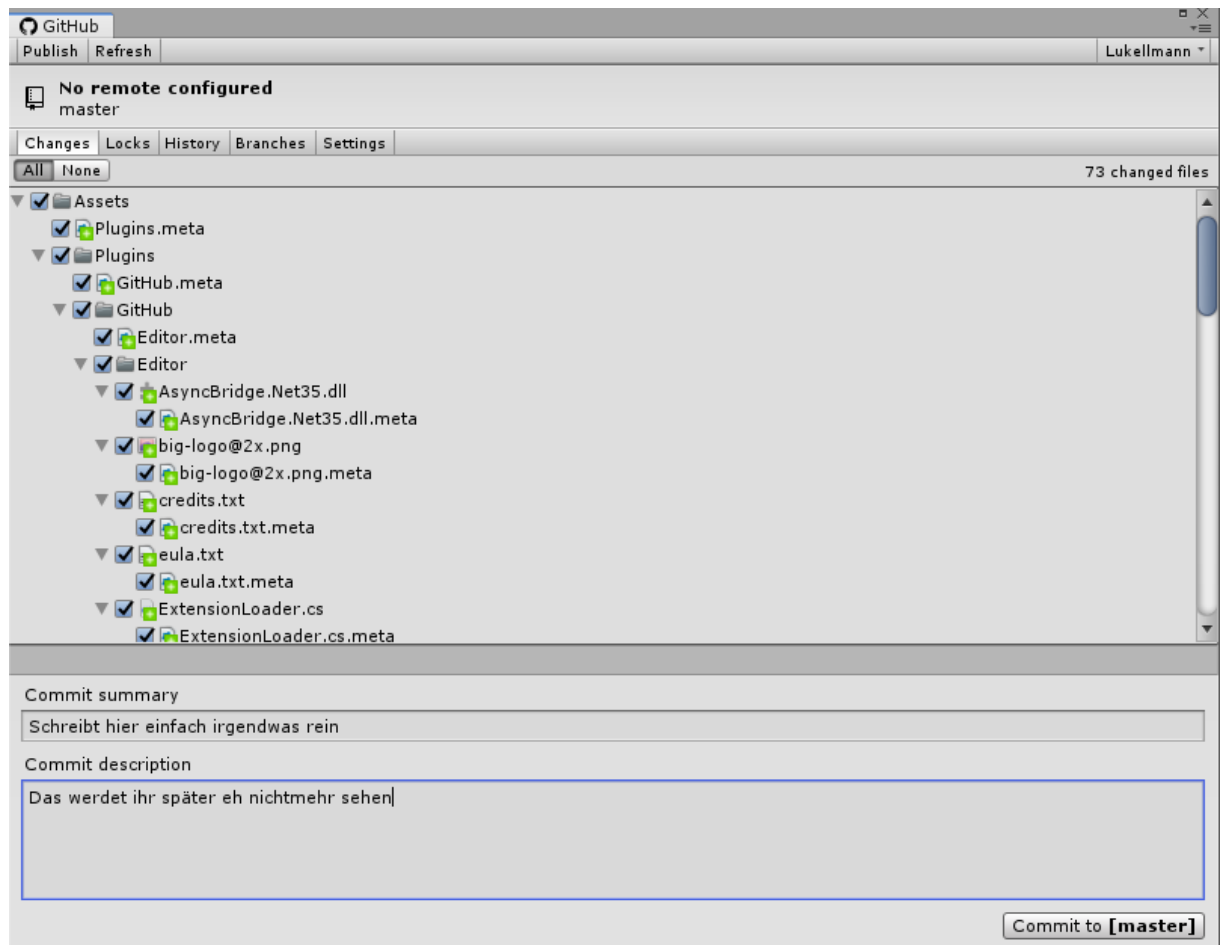
5. Im GitHub-Fenster erstmal oben rechts mit eurem GitHub-Account anmelden (ich bin schon angemeldet) und dann nochmal bei Settings euren Namen und eure Email angeben.



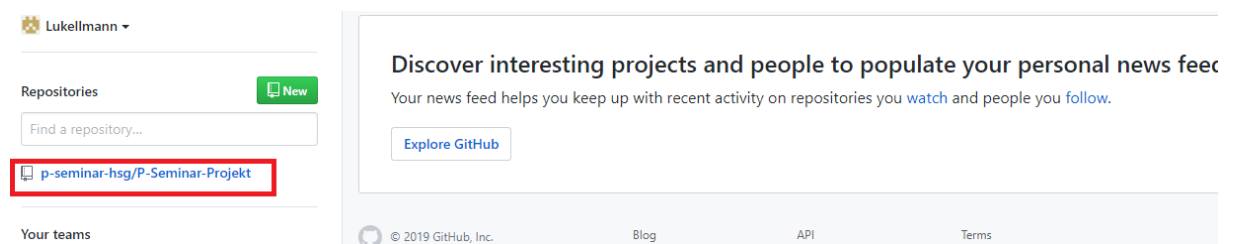
6. Jetzt unter Initialize auf „Initialize a git repository for this project“ klicken um ein lokales Repository für das Projekt zu erstellen. Sollte der Vorgang abbrechen und wieder der gleiche Screen sichtbar sein, einfach das Projekt schließen (notfalls mit dem Task-Manager falls sich Unity nicht mehr schließen lässt), den PC neustarten und dann nochmal probieren (hat zumindest bei mir immer funktioniert).



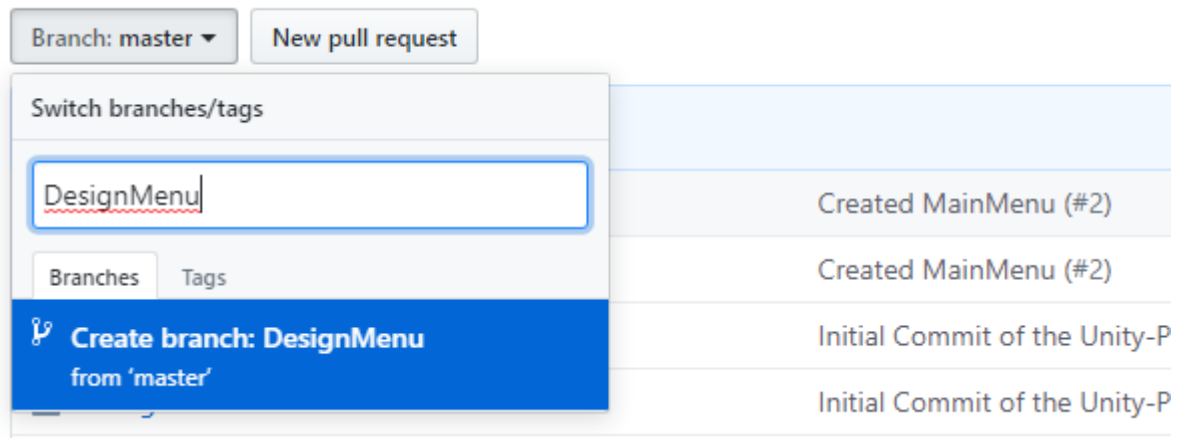
7. Unter Changes alle auswählen, eine Commit summary und description eingeben (ist hier egal, werdet ihr später nichtmehr sehen) und dann auf „Commit to [master]“ klicken, sodass nach dem Commit die Liste der Changes leer ist.



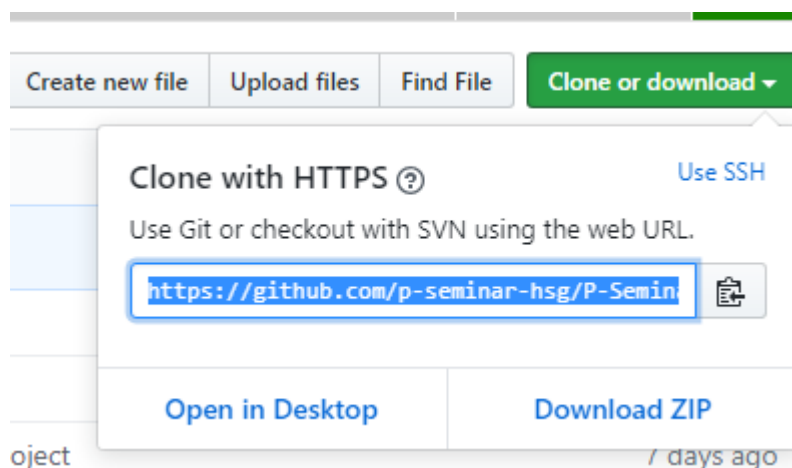
8. Jetzt geht ihr im Browser auf github.com, meldet euch mit eurem GitHub-Account an und geht auf das Repository „P-Seminar-Projekt“.



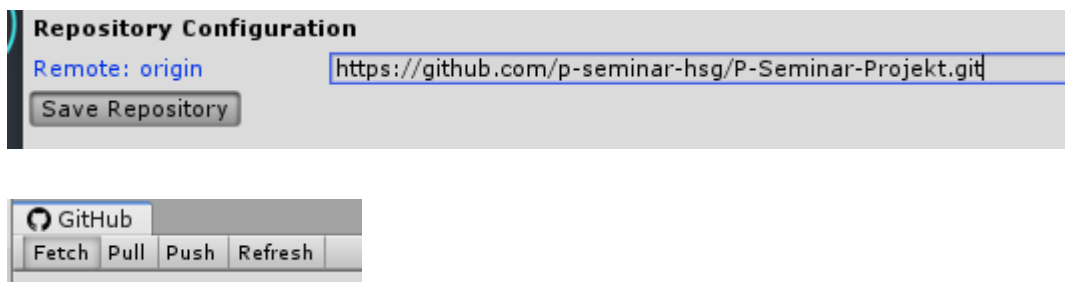
9. In diesem Repository erstellt ihr jetzt einen neuen Branch indem ihr hier einen (sinnvollen) Namen eingibt und auf „Create branch“ klickt.



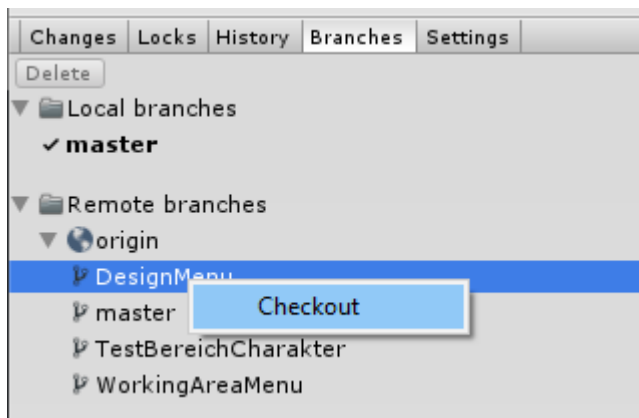
10. Jetzt holt ihr euch die URL des Repositories indem ihr auf „Clone or download“ drückt und die dann erscheinende Adresse kopiert.



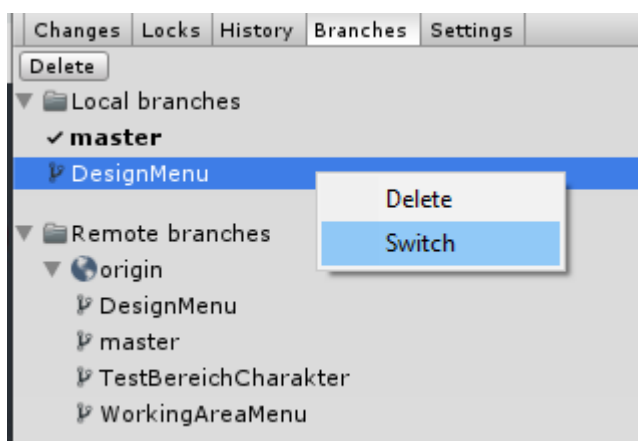
11. Zurück in Unity geht ihr im GitHub-Fenster wieder auf Settings, fügt unter „Remote: origin“ die URL ein, drückt auf „Save Repository“ und dann auf „Fetch“ (oberste Leiste).



12. Wenn ihr auf Branches geht, sollten euch euch unter „Remote branches“ die Branches unseres Repositories angezeigt werden. Hier macht ihr jetzt einen Rechtsklick auf den vorhin von euch erstellten Branch und drückt auf „Checkout“.

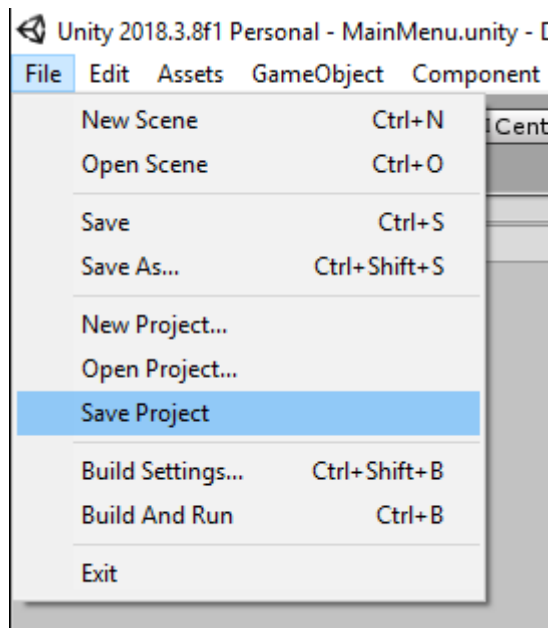


13. Jetzt erscheint auch unter „Local branches“ der von euch erstellte Branch. Auf diesen macht ihr wieder einen Rechtsklick und wählt „Switch“ aus (Falls das nicht funktioniert müsst ihr evtl. nochmal unter Changes neue Changes zum lokalen master-Branch committen (siehe Schritt 7)). Achtung: Wenn ihr zu dem anderen Branch wechselt, werden die Änderungen am Projekt, die ihr evtl. gemacht habt (z.B. neue Scrpts erstellt), NICHT übernommen (sind allerdings noch im lokalen master-Branch enthalten), da in dem neuen Branch nur die Änderungen enthalten sind, die vorher auf GitHub hochgeladen wurden. D.h. ihr müsst diese Änderungen halt nochmal in Unity reinziehen.

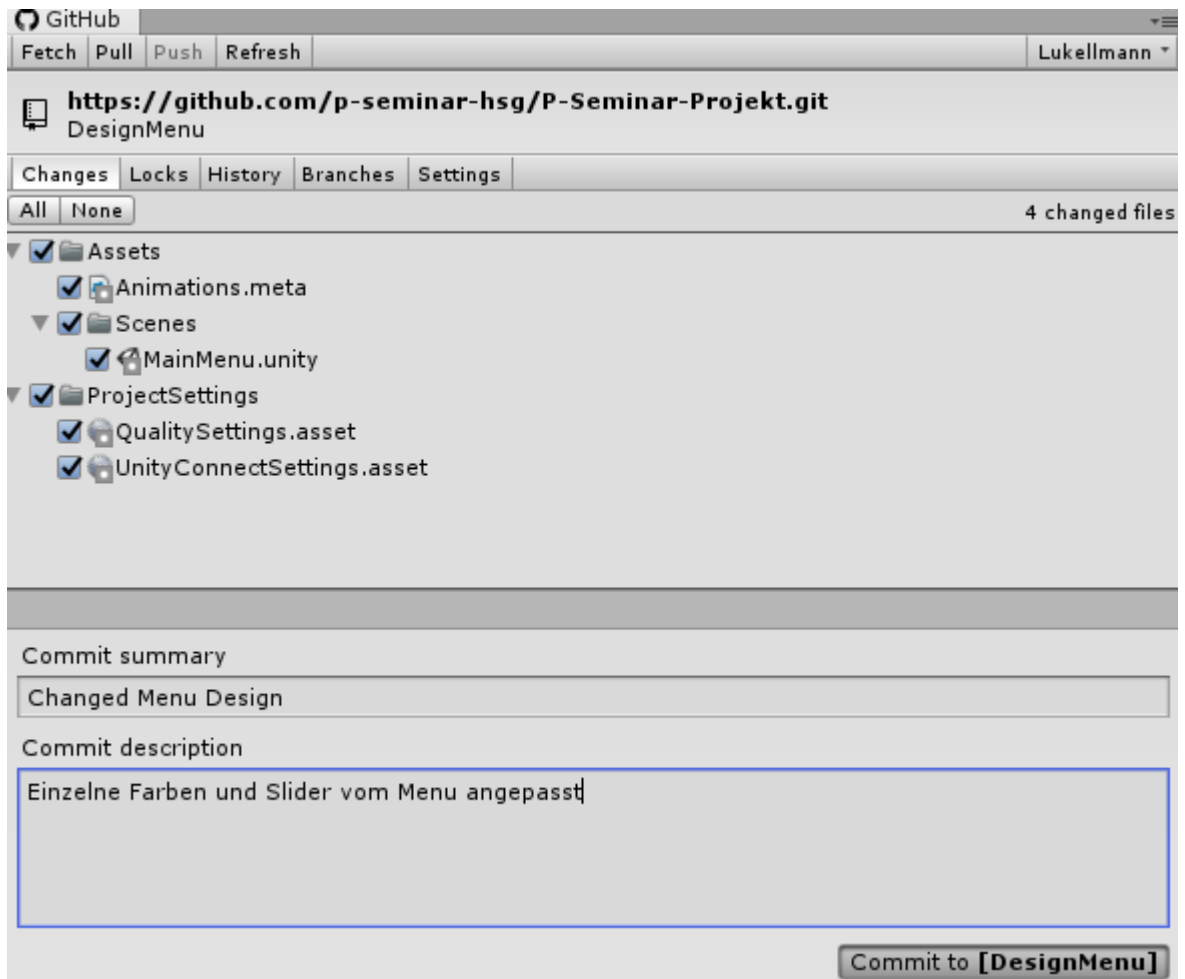


14. Nachdem ihr den Branch gewechselt habt, solltet ihr jetzt den Stand des Projekts aus dem master-Branch unseres Repositories in Unity haben. Jetzt könnt ihr am Projekt arbeiten.

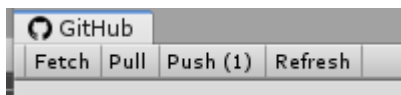
15. Wenn ihr dann damit fertig seid, unter File > Save (aktuelle Scene speichern) und Save Project anklicken um alles zu speichern, damit auch alle Änderungen unter Changes im GitHub-Fenster erscheinen.



16. Im GitHub-Fenster wieder alle Changes zum aktuellen Branch committen (siehe Schritt 7), diesmal aber bitte eine sinnvolle Commit summary und description wählen, da diese auch später sichtbar sein werden.

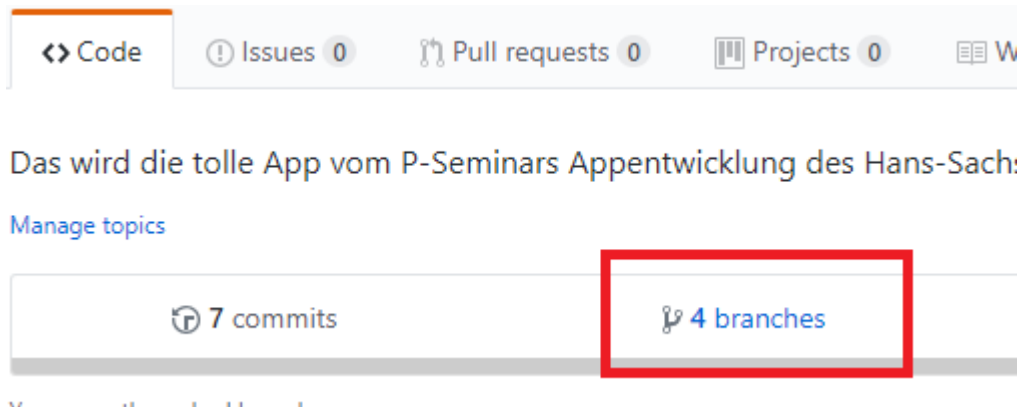


17. Jetzt dürfte in der oberen Leiste der Button „Push“ nichtmehr ausgegraut sein und dahinter sollte eine (1) stehen. Drückt jetzt auf Push. Sollte dieser Vorgang nicht funktionieren (es erscheint KEINE Meldung „Branch pushed“), schließt das Projekt, startet den PC neu und versucht es nochmal (hat bei mir auch funktioniert).

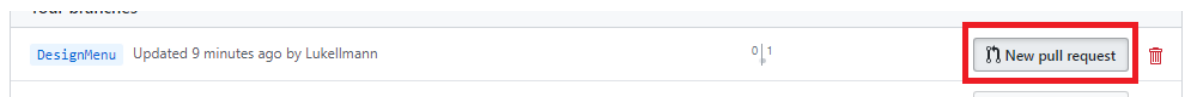




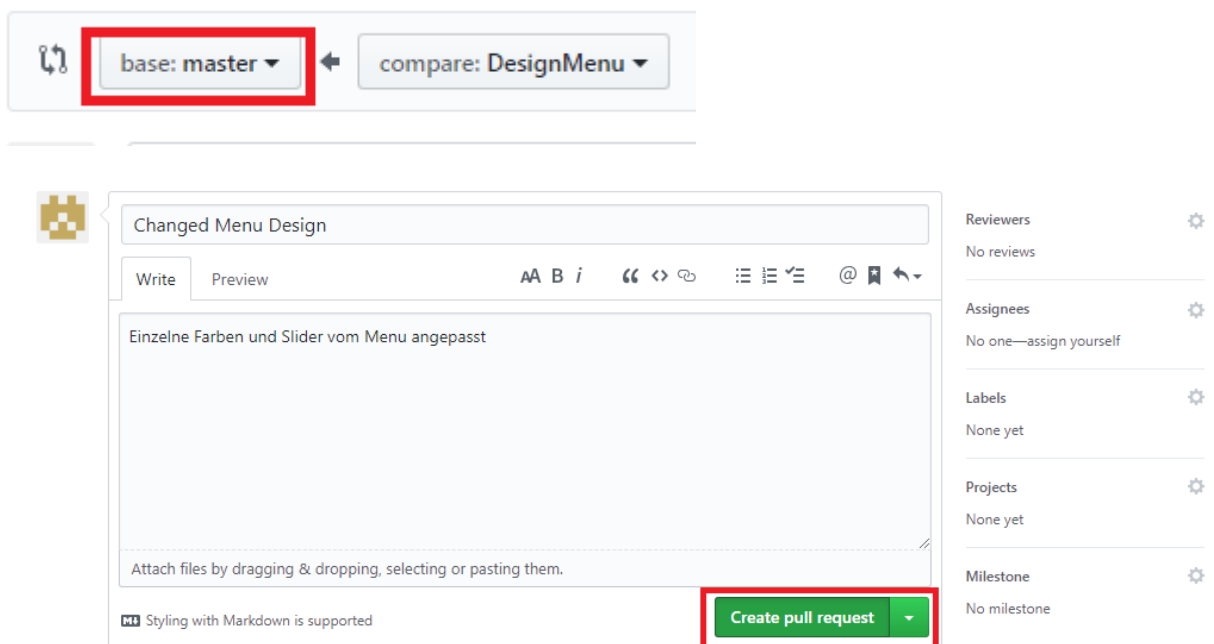
18. Wenn ihr den Branch nun erfolgreich pushen konntet, geht ihr wieder in euren Browser und geht im GitHub-Repository auf den Reiter „branches“.



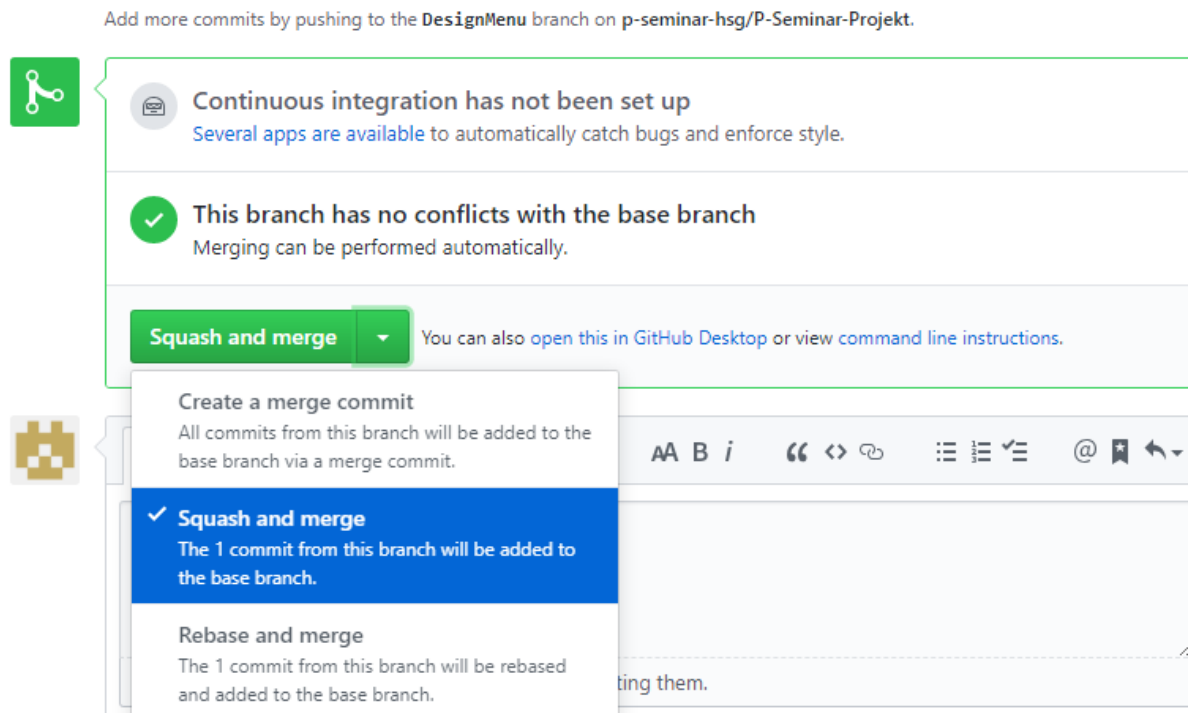
19. Jetzt klickt ihr hinter eurem Branch, mit dem ihr gearbeitet habt auf „New pull request“.



20. Hier könnt ihr nochmal die Beschreibung ändern und klickt dann auf „Create pull request“. Achtete auch darauf, dass als base der master-Branch ausgewählt ist. Wenn es Löschungen gab, werden euch diese im nächsten Schritt angezeigt. Schaut euch diese bitt durch, damit ihr nicht versehentlich etwas unbeabsichtigt löscht.



21. Im nächsten Schritt wählt ihr „Squash and merge“ aus und bestätigt das.



22. Jetzt ist der neue Inhalt von eurem Branch auch im master-Branch enthalten, d.h. ihr könnt euren Branch löschen und wieder einen neuen öffnen, wenn ihr das nächste Mal am Projekt arbeitet (so habt ihr immer die aktuellste Version des Projekts). Das in Schritt 1 erstellte Unity-Projekt könnt ihr übrigens weiterhin benutzen, ihr müsst nur wieder auf einen evtl. neu erstellten Branch wechseln. Drückt immer, wenn ihr einen neuen Branch erstellt habt auf „Fetch“, dann aktualisiert dich die Liste der Branches.