

# Praktische Informatik 1

## Versionsverwaltung mit git

Thomas Röfer

Cyber-Physical Systems  
Deutsches Forschungszentrum für  
Künstliche Intelligenz

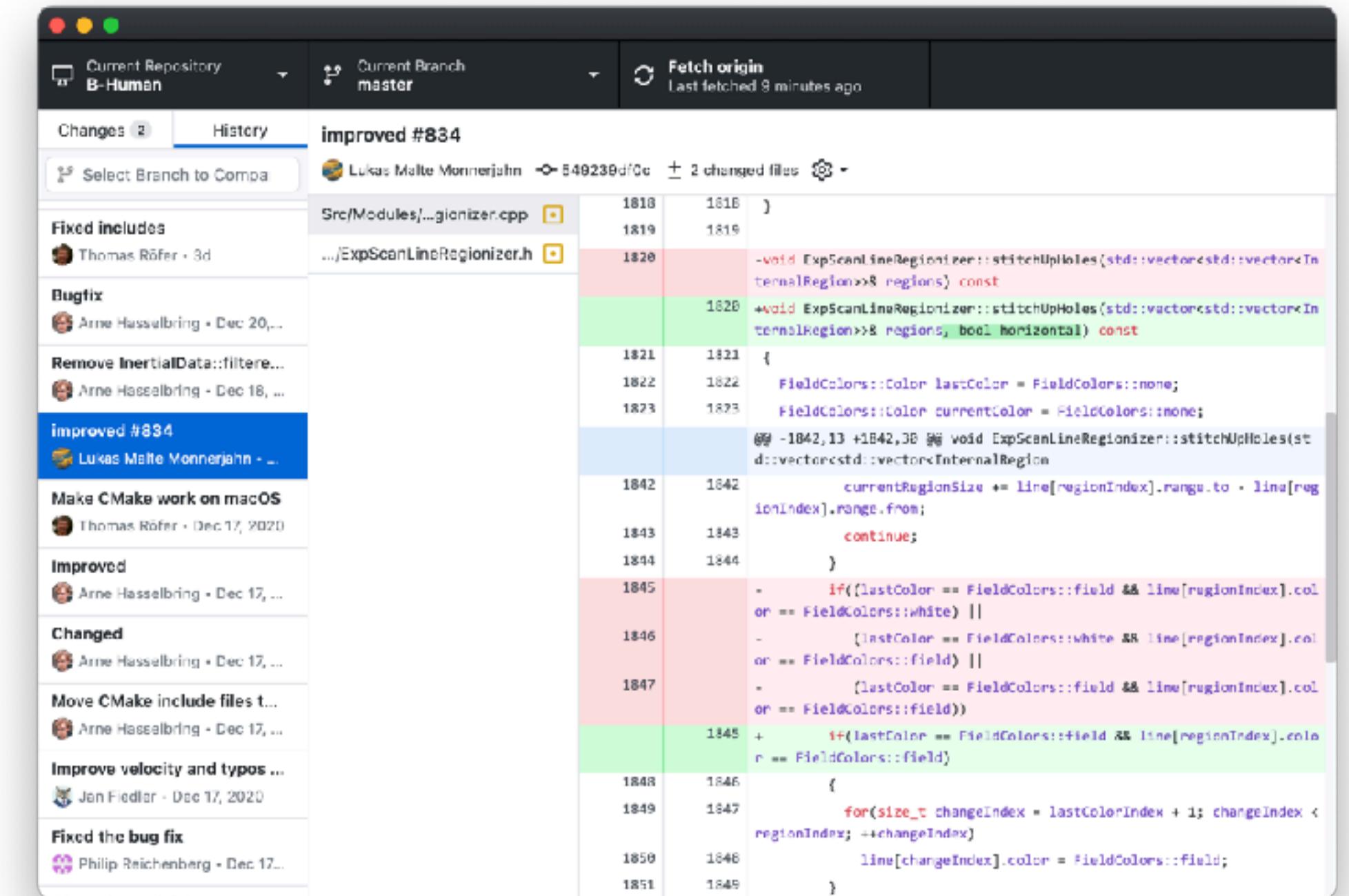
Multisensorische Interaktive Systeme  
Fachbereich 3, Universität Bremen



# Versionsverwaltung mit git

- Änderungshistorie verwalten
  - Wer hat was wann wie verändert?
  - Vergisst nie etwas, d.h. alle alten Versionen der Dateien sind noch da!

- Entwicklung wurde von Linus Torwalds initiiert
- Zusammenarbeit über das  GitLab der Informatik
  - **gitlab.informatik.uni-bremen.de** (Nicht **gitlab.com**!)
- Alle Abgaben passieren ab nun per GitLab

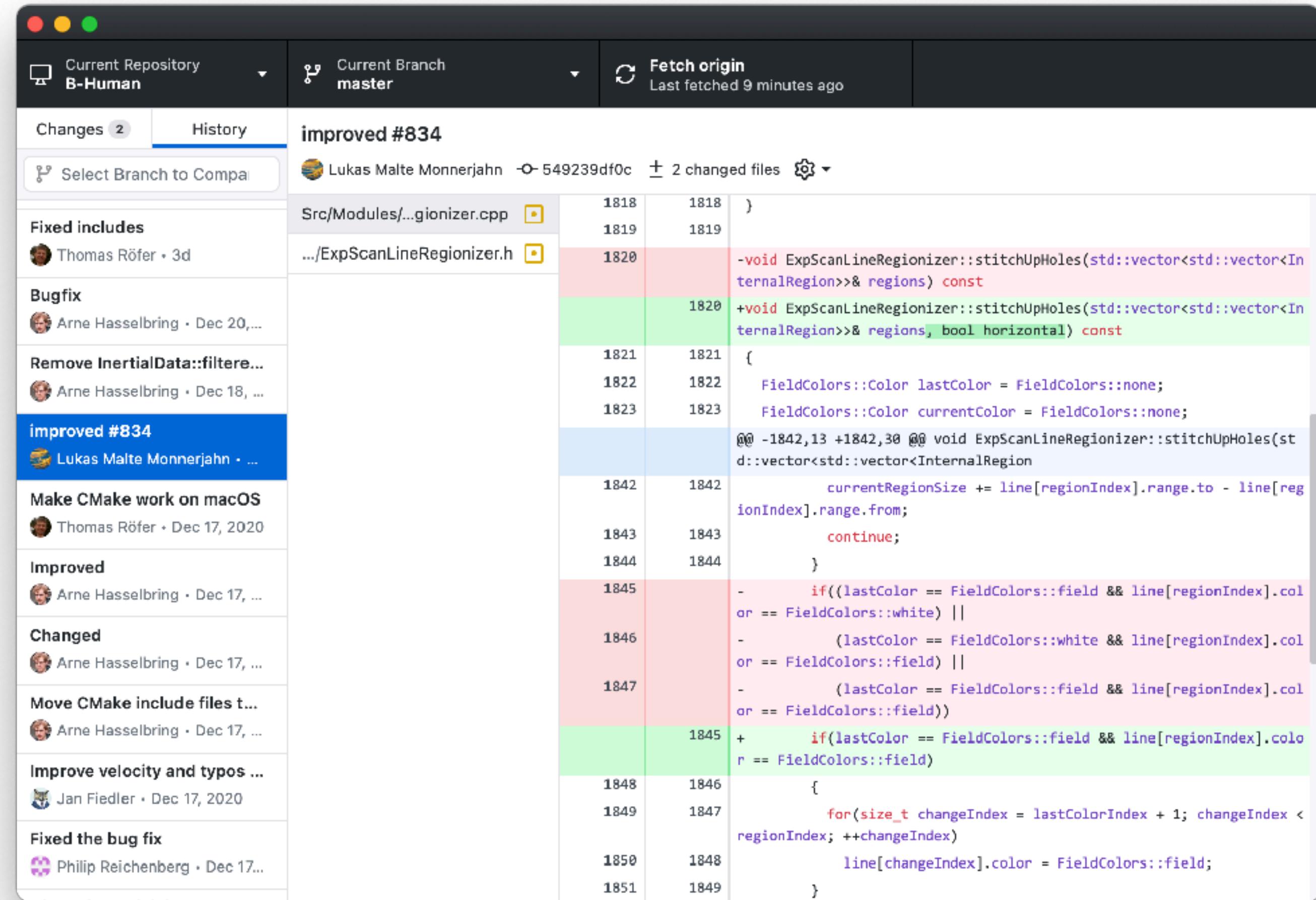


The screenshot shows a Git commit history for a repository named 'B-Human'. The current branch is 'master' and the last fetch was 9 minutes ago. The 'History' tab is selected, showing a list of commits. The commit 'improved #834' is highlighted. The diff view shows a block of code with color-coded changes: red for deleted lines, green for added lines, and blue for modified lines. The code is related to 'ExpScanLineRegionizer' and 'stitchUpHoles'.



Von Krd (Foto), Sprat (Zuschnitt), CC BY-SA 4.0, Wikipedia

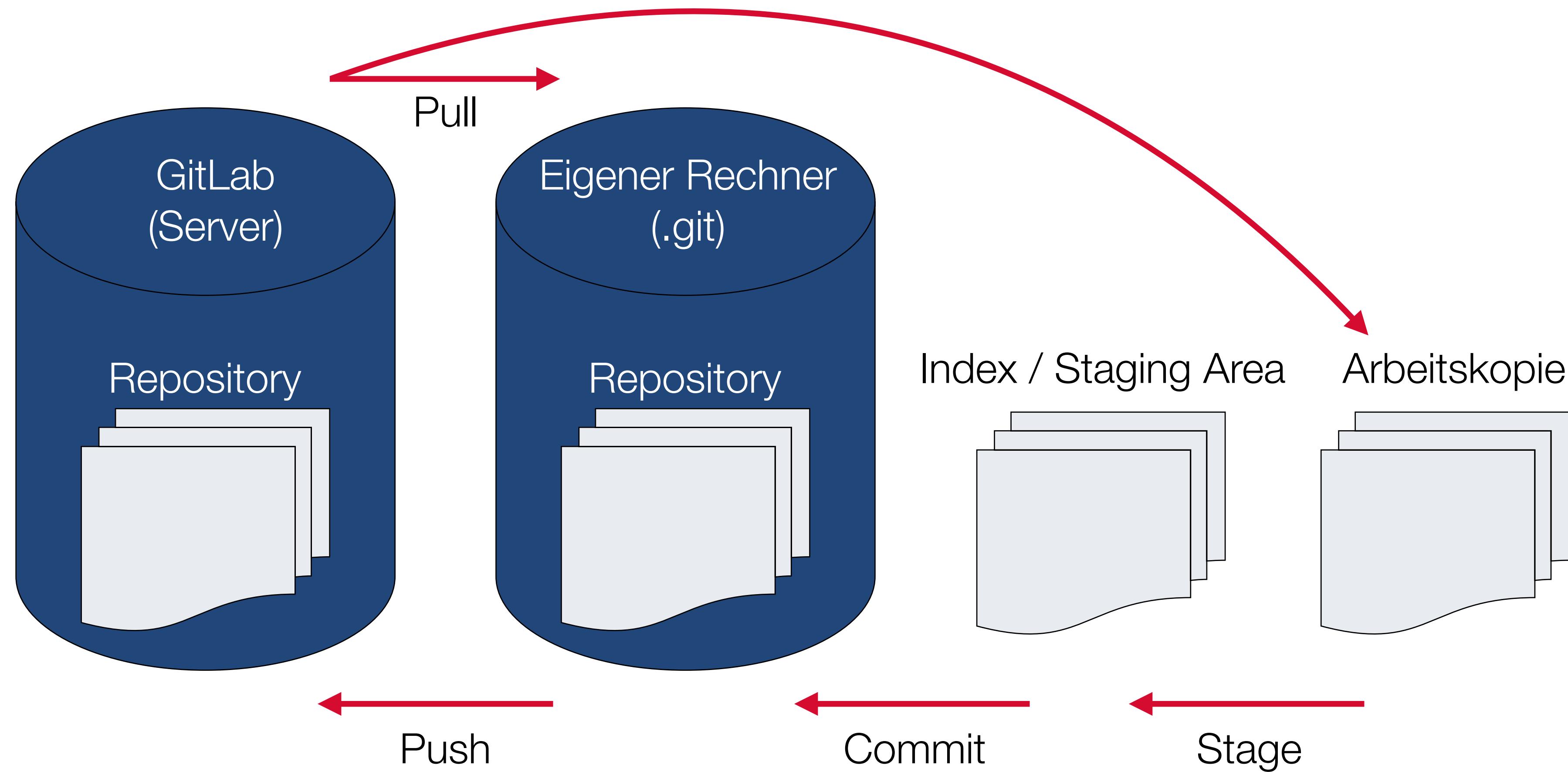
# Entwickeln mit GitHub Desktop und GitLab: Demo



The screenshot shows the GitHub Desktop application interface. The top bar displays the current repository as 'B-Human', the current branch as 'master', and the status of the 'Fetch origin' operation. The main window is focused on pull request #834, which has been improved. The commit message is 'improved #834' by Lukas Malte Monnerjahn. The diff view shows changes in two files: 'Src/Modules/...gionizer.cpp' and '.../ExpScanLineRegionizer.h'. The changes are color-coded: red for deleted code, green for added code, and grey for modified code. The code changes involve modifications to the 'stitchUpHoles' function, specifically handling internal regions and field colors.

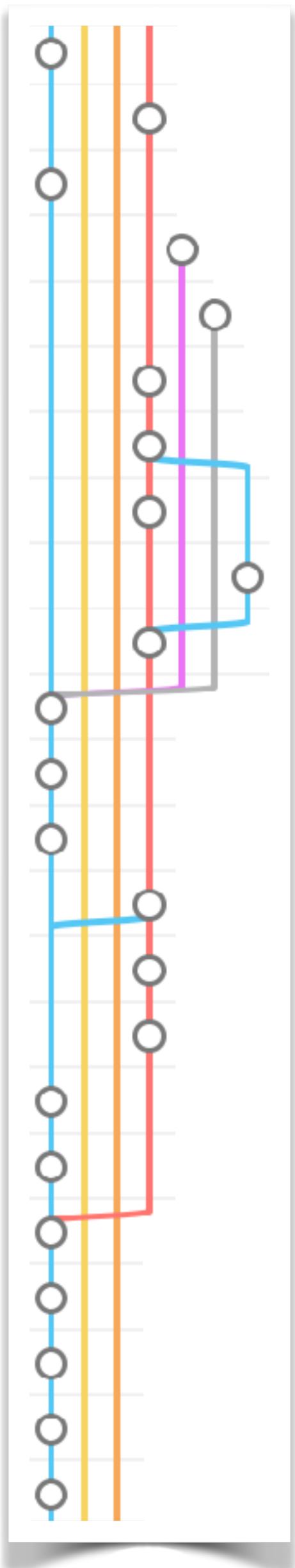
```
1818 1818 )  
1819 1819  
1820 -void ExpScanLineRegionizer::stitchUpHoles(std::vector<std::vector<In  
1820 +void ExpScanLineRegionizer::stitchUpHoles(std::vector<std::vector<In  
1821 1821 {  
1822 1822     FieldColors::Color lastColor = FieldColors::none;  
1823 1823     FieldColors::Color currentColor = FieldColors::none;  
1842 1842     currentRegionSize += line[regionIndex].range.to - line[reg  
1843 1843         ionIndex].range.from;  
1844 1844         continue;  
1845 -         if((lastColor == FieldColors::field && line[regionIndex].col  
1846 -             or == FieldColors::white) ||  
1847 -             (lastColor == FieldColors::white && line[regionIndex].col  
1848 1845     +         if(lastColor == FieldColors::field && line[regionIndex].col  
1849 1846         or == FieldColors::field)  
1849 1847         {  
1850 1848             for(size_t changeIndex = lastColorIndex + 1; changeIndex <  
1851 1849                 regionIndex; ++changeIndex)  
1848             line[changeIndex].color = FieldColors::field;  
1849         }  
1849     }  
1849 }  
1849 }
```

# Überblick



# Begriffe

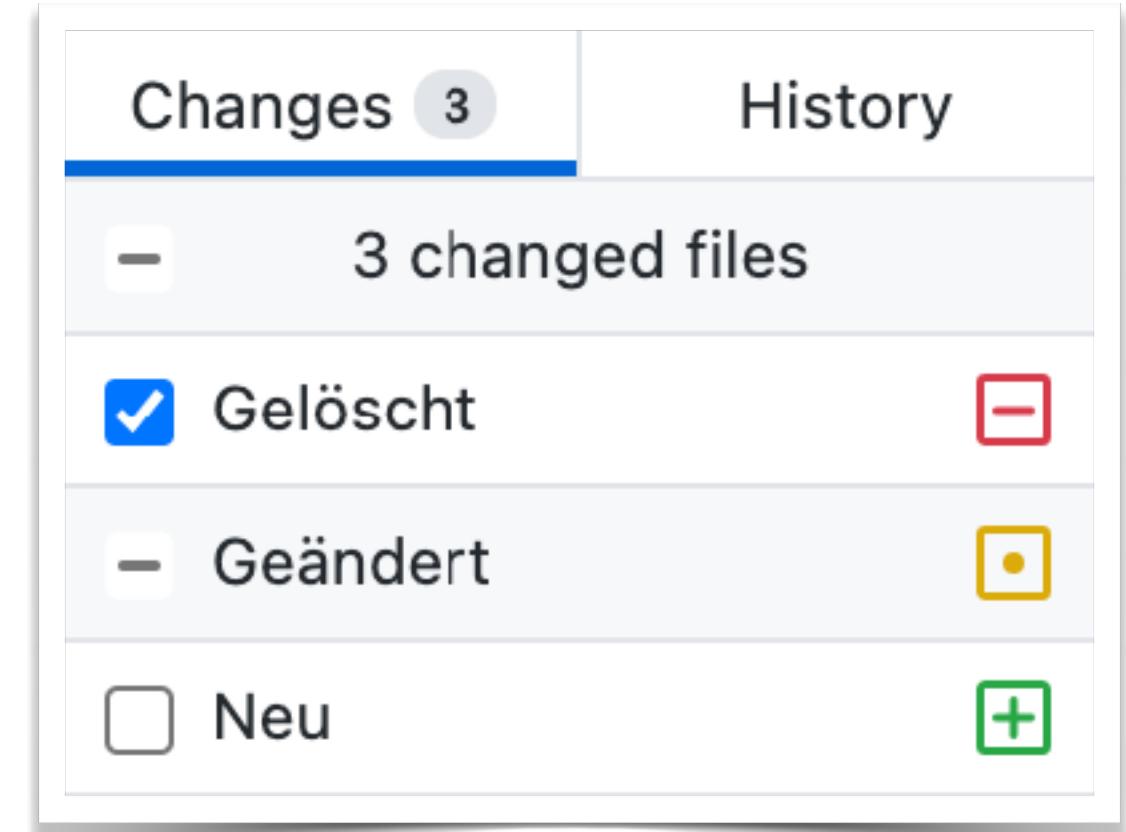
- **Arbeitskopie**: Dateien, an denen tatsächlich gearbeitet wird
- **Commit**: Zustand aller unter Versionsverwaltung stehender Dateien zu einem bestimmten Zeitpunkt, der explizit festgehalten wurde
  - Jeder Commit kennt seine Vorgänger-Commits → Historie
- **Branch**: Ein benannter Commit
  - Wir nutzen nur **main** und **origin/main**
- **Repository**: Eine Ansammlung von Commits mit der Markierung, welcher lokale Branch der aktuelle ist (**HEAD**)
- **Index (Staging Area)**: Explizit zusammengestellter Zustand von Dateien, aus dem der nächste Commit erzeugt werden kann



# Commit vorbereiten: Staging

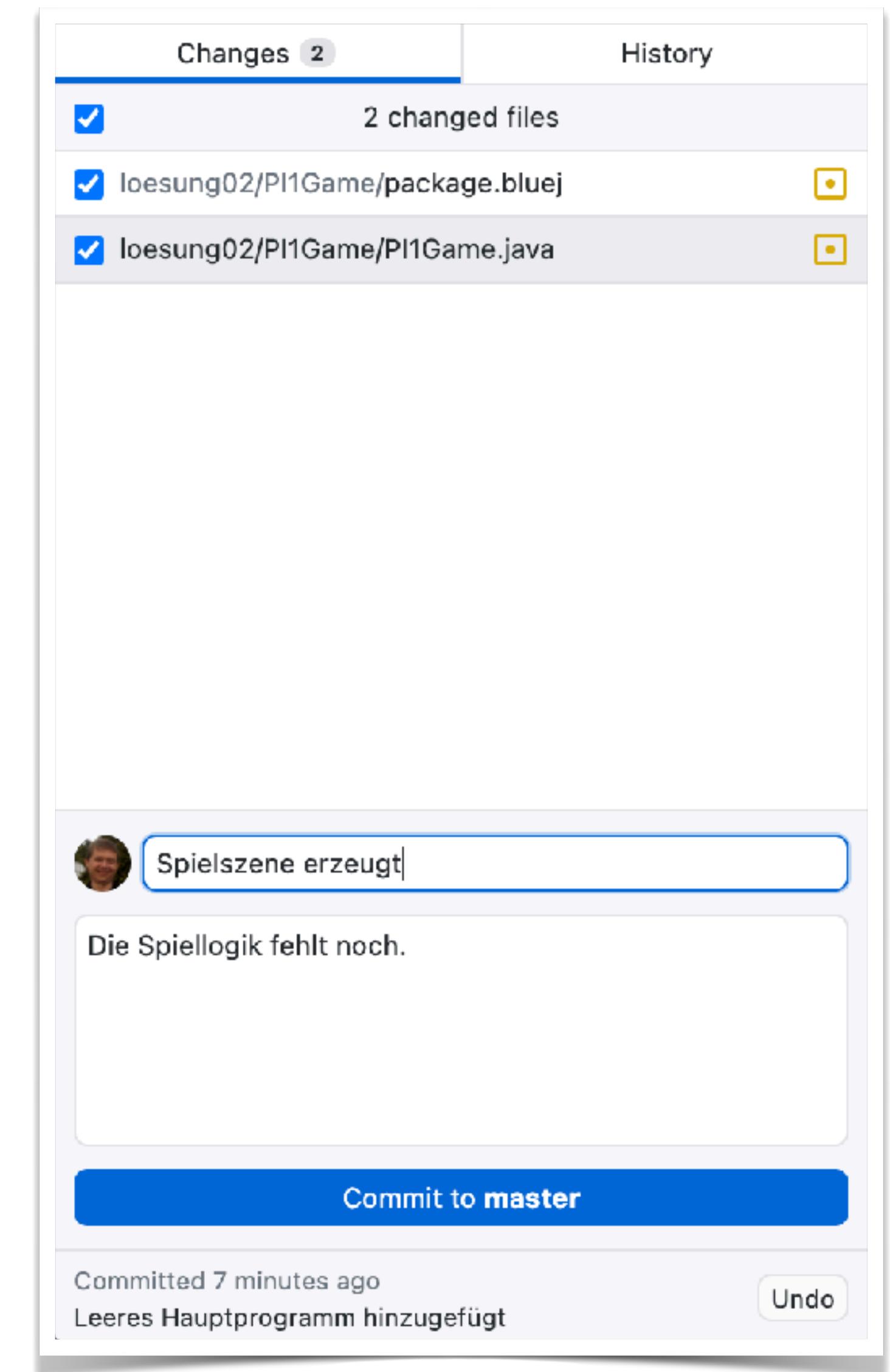
- Kartenreiter **Changes**

-    Dateien sind neu, geändert oder gelöscht
-    Änderungen werden (nicht/teilweise) Teil des nächsten Commits (**git add <Datei>** / **git reset <Datei>**)
- Einzelne Abschnitte von Textdateien lassen sich in der Ansicht rechts durch Anklicken der Zeilennummern markieren (Achtung: Für **geänderte** Zeilen muss Löschung **und** Einfügung markiert werden)
- Kontextmenü der Dateien
  - **Discard Changes...**: Datei wieder auf Zustand aus letztem Commit zurücksetzen (**git checkout -- <Datei>**)
  - **Ignore File**: Datei soll nicht unter Versionsverwaltung stehen (oft: generierte Dateien)
  - **Ignore All .<Endung> Files**: Alle Dateien mit dieser Endung ignorieren
- Ignorieren erzeugt/erweitert Datei **.gitignore** im Hauptverzeichnis, die auch Teil eines Commits werden sollte



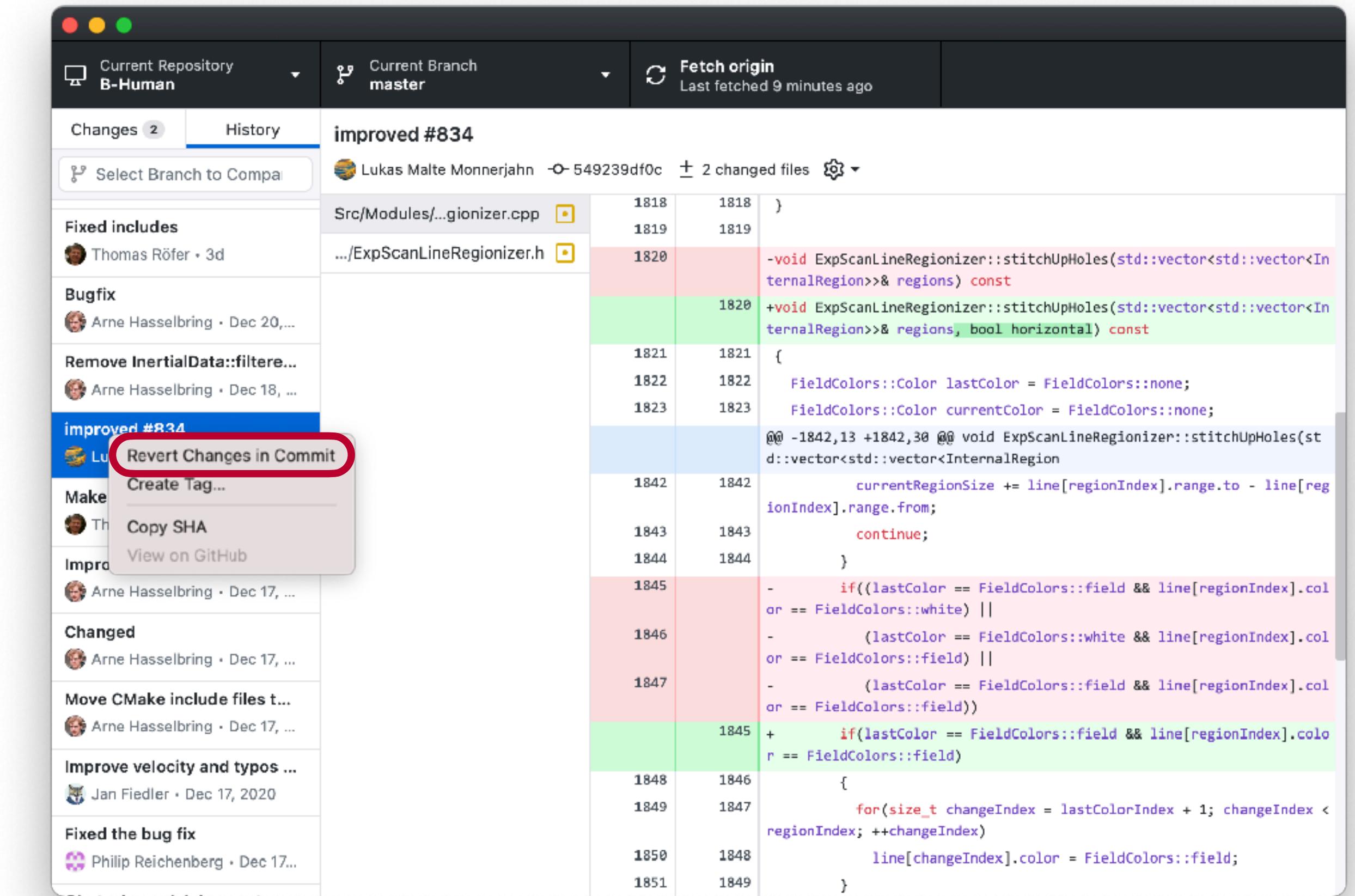
# Commit (git commit -m "<beschreibung>")

- Commit-Nachricht
  - Kurzbeschreibung für Bedeutung des Commits in einer Zeile (**ist ein Muss!**)
  - Zusätzliche/ausführlichere Informationen in weiterem Eingabefeld
- Commit landet erst einmal nur im Repository auf dem eigenen Rechner, **nicht auf dem Server!**
- **Undo** entfernt letzten Commit, wenn er noch nicht auf den Server übertragen wurde
  - Die Änderungen bleiben in der Arbeitskopie erhalten



# Historie (git log)

- Änderungsverlauf des Repositories
  - Beschreibung der Änderung
  - Wer hat geändert (mit Bild von [gravatar.com](https://gravatar.com))?
  - Was wurde geändert?



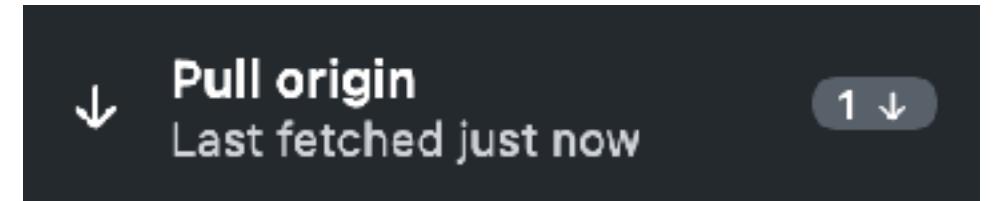
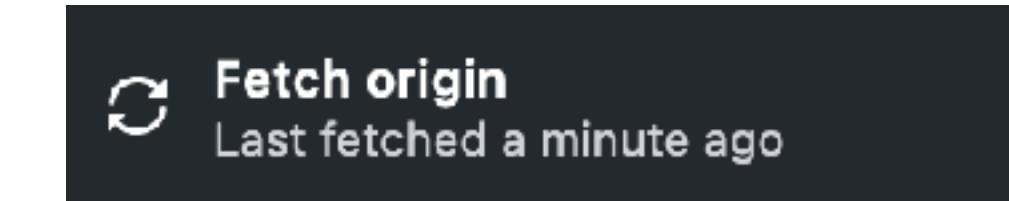
Current Repository: B-Human | Current Branch: master | Fetch origin: Last fetched 9 minutes ago

improved #834

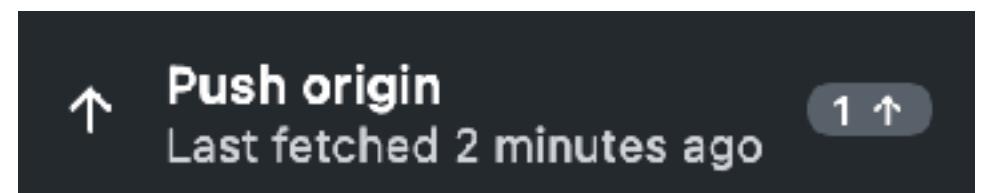
Lukas Malte Monnerjahn -O- 549239df0c ± 2 changed files

File	Line 1818	Line 1819	Line 1820	Line 1821	Line 1822	Line 1823	Line 1824	Line 1825	Line 1826	Line 1827	Line 1828	Line 1829	Line 1830	Line 1831	Line 1832	Line 1833	Line 1834	Line 1835	Line 1836	Line 1837	Line 1838	Line 1839	Line 1840	Line 1841	Line 1842	Line 1843	Line 1844	Line 1845	Line 1846	Line 1847	Line 1848	Line 1849	Line 1850	Line 1851																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
Src/Modules/...gionizer.cpp	1818	1818	1820	1821	1822	1823	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607	2608	2609	2610</td

## Änderungen vom Server holen



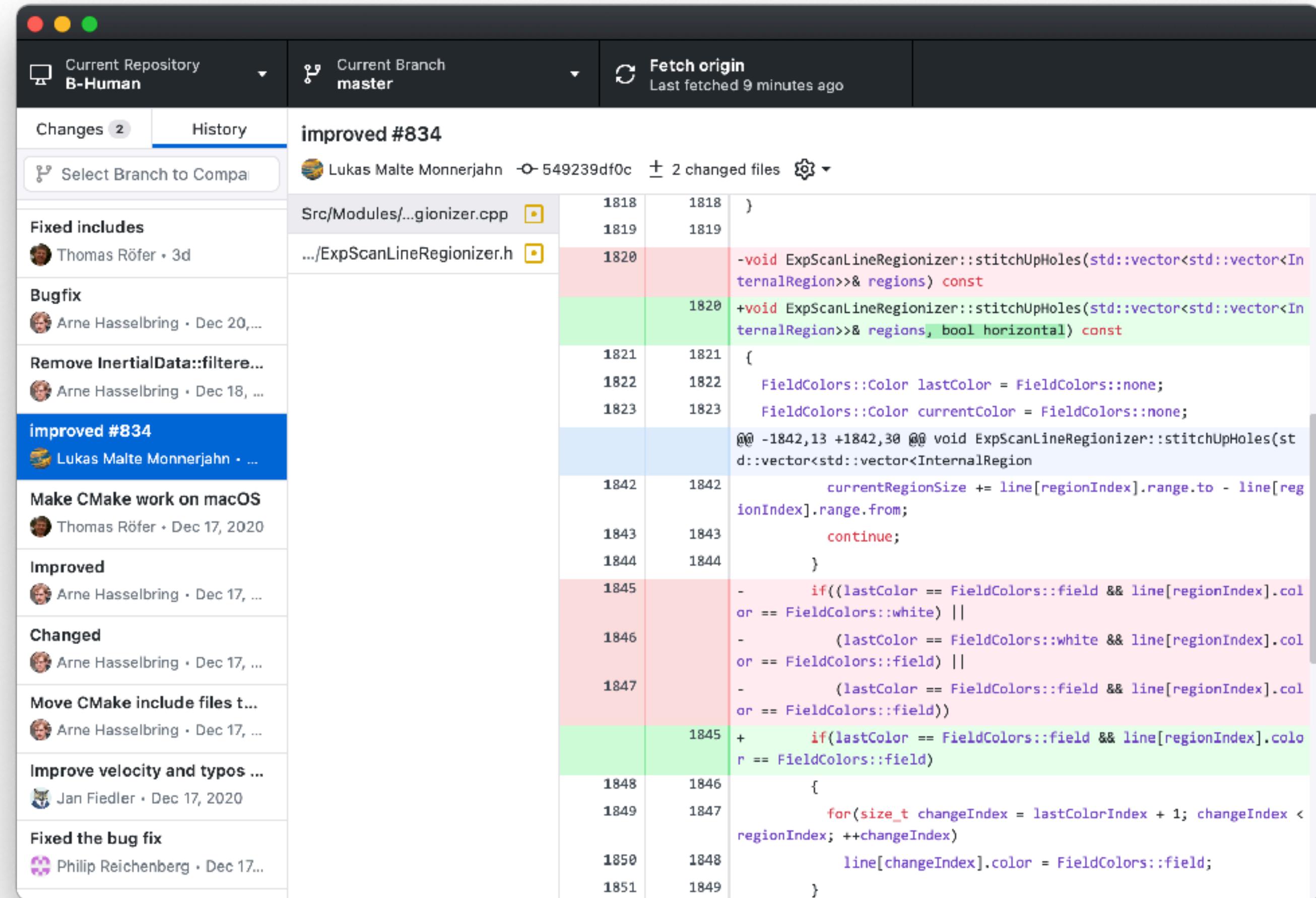
- **Fetch**: Holt neue Commits vom Server und integriert sie ins lokale Repository (**git fetch**)
- **Pull**: Wie **Fetch**, aber integriert neue Commits auch in die Arbeitskopie (**git pull**)
  - GitLab Desktop sichert vorher automatisch geänderte Dateien in der Arbeitskopie und stellt sie nachher wieder her (**git pull --autostash**)
  - **Pull** kann **Konflikte** erzeugen!
- Gibt es lokal und auf dem Server neue Commits, wird ein **Merge Commit** erstellt, der zwei Eltern-Commits hat
  - Eventuell aufgelöste Konflikte werden Teil des **Merge Commits**



## Änderungen auf Server übertragen (git push)

- Spielt alle lokalen Commits auf den Server, die dort noch fehlen
- Geht nur, wenn alle Commits des Servers lokal vorhanden sind
- Ablauf
  - (**Fetch** und) **Pull**, um alle Commits des Servers zu holen
  - **Testen**, ob diese zusammen mit den lokalen Änderungen funktionieren (keine Fehler auf den Server spielen)!
  - **Push**, wenn inzwischen keine neuen Commits auf dem Server sind

# Konflikte: Demo



The screenshot shows a GitHub pull request interface for a repository named 'B-Human' on the 'master' branch. The pull request is titled 'improved #834' and is authored by Lukas Malte Monnerjahn. The commit hash is 549239df0c, and it has 2 changed files. The code editor displays a conflict in the file 'Src/Modules/...gionizer.cpp'. The conflict highlights lines 1820 and 1821 in red (from the top branch) and green (from the bottom branch). The code block shows a merge conflict between two changes to the same line of code.

```

1818 1818 }
1819 1819
1820 1820 -void ExpScanLineRegionizer::stitchUpHoles(std::vector<std::vector<In
1821 1821 +void ExpScanLineRegionizer::stitchUpHoles(std::vector<std::vector<In
1822 1822 ternalRegion>>& regions) const
1823 1823 +void ExpScanLineRegionizer::stitchUpHoles(std::vector<std::vector<In
1824 1824 ternalRegion>>& regions, bool horizontal) const
1825 1825 {
1826 1826     FieldColors::Color lastColor = FieldColors::none;
1827 1827     FieldColors::Color currentColor = FieldColors::none;
1828 1828     @@ -1842,13 +1842,30 @@ void ExpScanLineRegionizer::stitchUpHoles(st
1829 1829         currentRegionSize += line[regionIndex].range.to - line[reg
1830 1830         ionIndex].range.from;
1831 1831         continue;
1832 1832     }
1833 1833 -     if((lastColor == FieldColors::field && line[regionIndex].col
1834 1834 or == FieldColors::white) ||
1835 1835 -     (lastColor == FieldColors::white && line[regionIndex].col
1836 1836 or == FieldColors::field) ||
1837 1837 -     (lastColor == FieldColors::field && line[regionIndex].col
1838 1838 or == FieldColors::field))
1839 1839 +     if(lastColor == FieldColors::field && line[regionIndex].col
1840 1840 or == FieldColors::field)
1841 1841     {
1842 1842         for(size_t changeIndex = lastColorIndex + 1; changeIndex <
1843 1843             regionIndex; ++changeIndex)
1844 1844             line[changeIndex].color = FieldColors::field;
1845 1845     }

```

# Konflikte vermeiden

- Nur Quelldateien unter Versionsverwaltung stellen
- Generierte Dateien ignorieren
  - In GitLab im Hauptverzeichnis des Repositories Datei **.gitignore** erstellen, Inhalt der **gitignore.txt** aus Stud.IP einfügen und commit-ten
- **BlueJ** vor Arbeit mit **git** beenden
  - Schreibt Projektdatei einheitlich neu
  - Laufendes **BlueJ** bemerkt Änderungen durch **Pull** vielleicht nicht
- **Discard** für **package.bluej** ausführen, wenn Klassenanordnung nicht geändert wurde

```
# Temporäre BlueJ-Dateien
*.class
*.ctxt
**/doc/*.html
**/doc/*.js
**/doc/*.zip
**/doc/element-list
**/doc/jquery
**/doc/resources
**/doc/stylesheets.css
**/doc/logfile.txt
**/doc/package-list
**/doc/legal

# Temporäre TeX-Dateien
*.aux
*.bbl
*.blk
*.fdb_latexmk
*.fls
*.log
*.synctex.gz
*.toc

# Versteckte Dateien (aber .gitignore behalten)
*.
!.gitignore
```

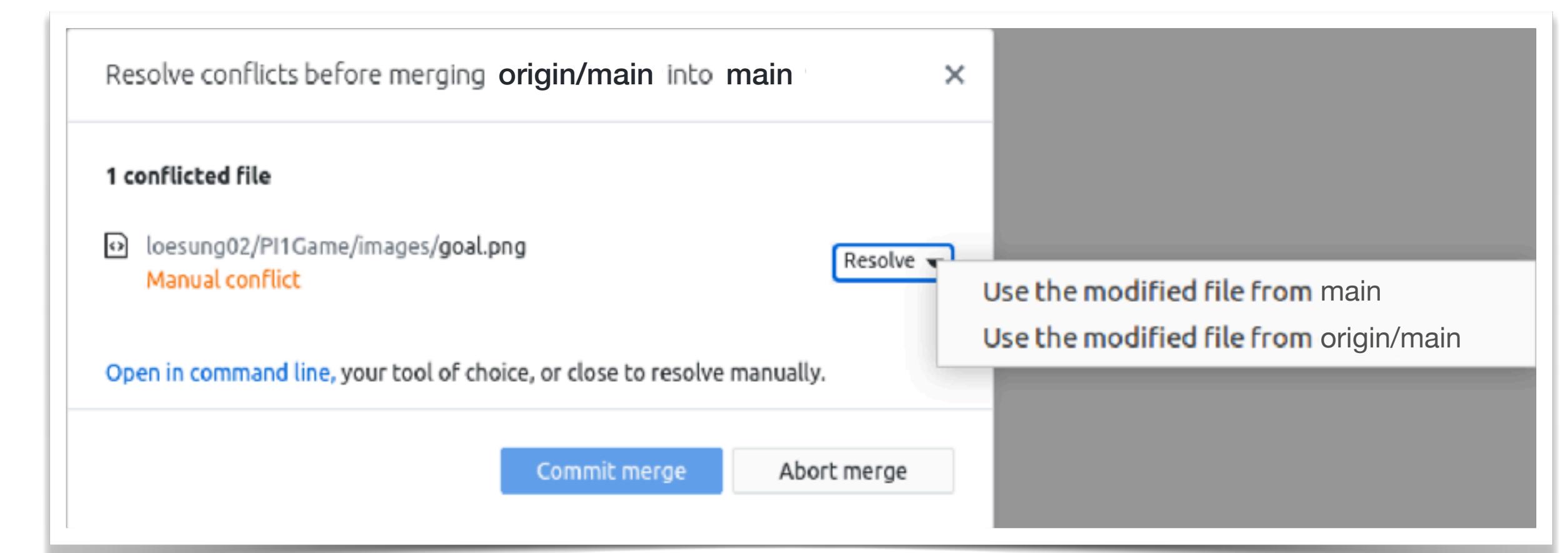
# Konflikte auflösen

- Textdateien: Datei mit Konflikten in Texteditor öffnen
  - Text zwischen Konfliktmarkierungen geeignet kombinieren
  - Konfliktmarkierungen entfernen

<<<<< HEAD  
 final int key = getNextKey();  
 if (key == VK\_RIGHT) {  
     player.setRotation(0);  
     :  
 } else {  
     playSound("error");  
     continue;  
 }=====switch (getNextKey()) {  
     case VK\_RIGHT -> {  
         player.setRotation(0);  
         :  
     } default -> {  
         playSound("error");  
 }>>>>> a1d60520c29b360f1822a0bb922fd3998f9bb421

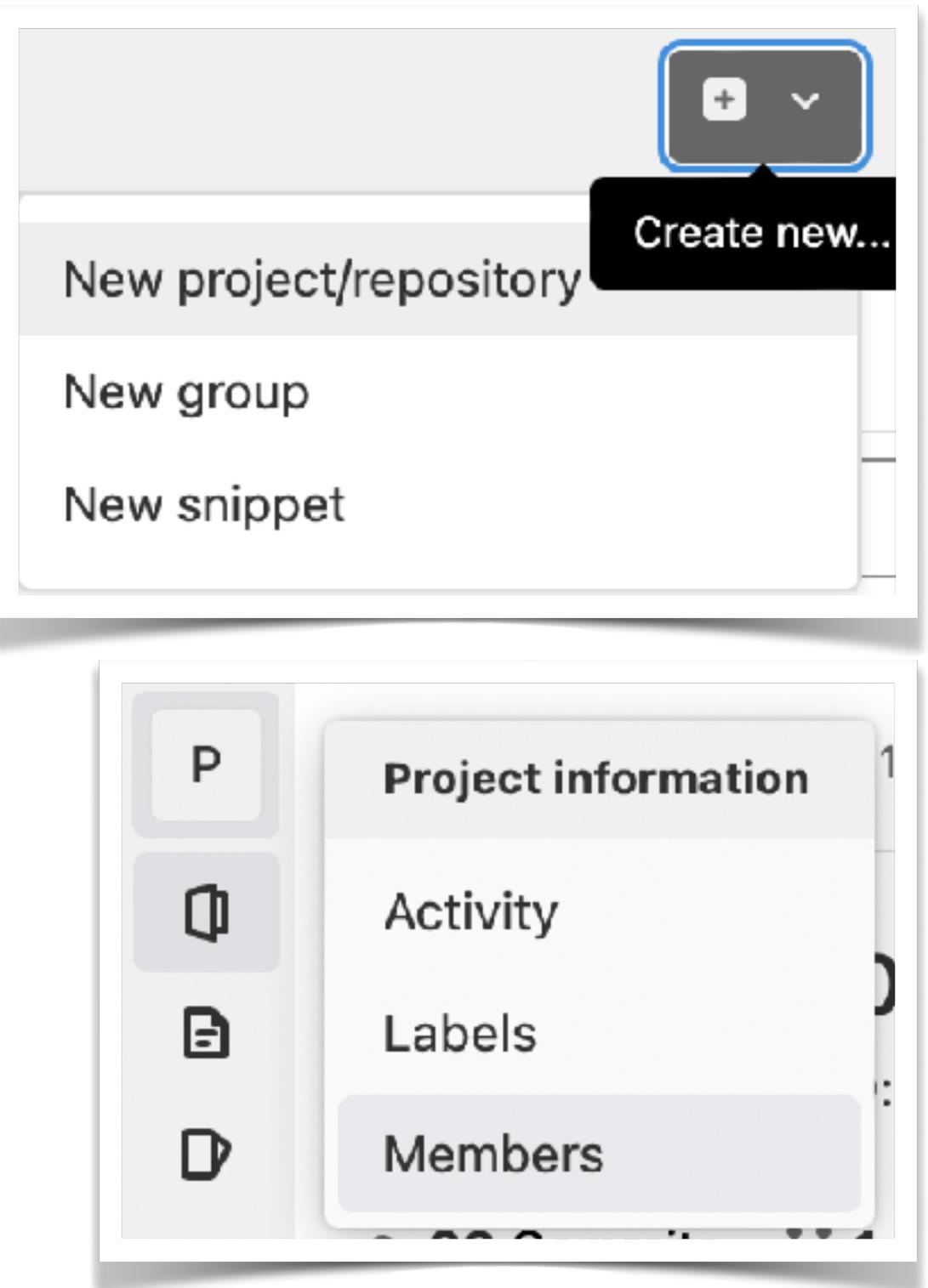
- Andere Dateien: In Menü **Resolve** eigene Version (**main**) oder die vom Server (**origin/main**) auswählen

- Danach **Commit Merge** auswählen

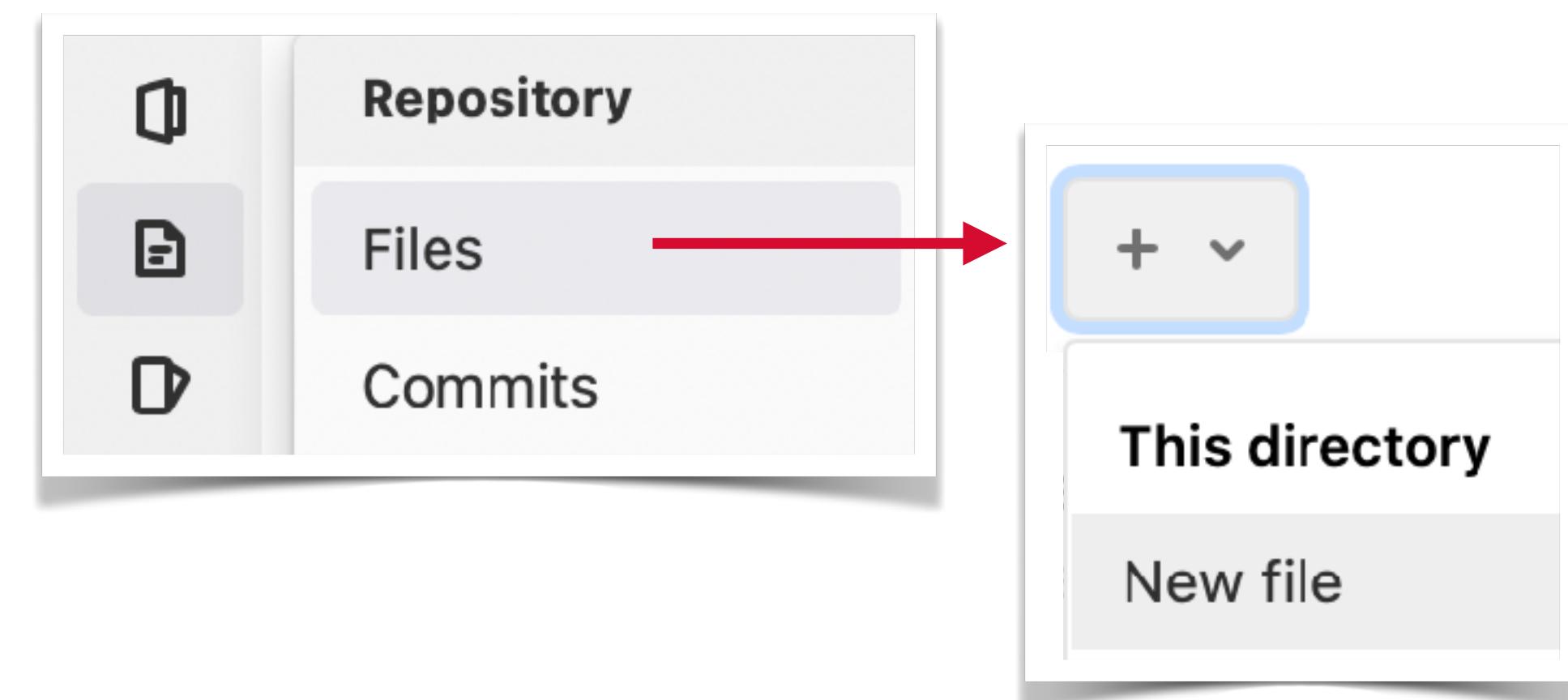


# Repository einrichten (git clone <Adresse>)

- Neues Projekt **pi1-2022** erzeugen und eure Tutor:in als **Developer** einladen
- In GitLab die Datei **.gitignore** erstellen, Inhalt der **gitignore.txt** aus Stud.IP einfügen und commit-ten
- Repository mit Git(Hub Desktop) von **HTTPS**-Adresse klonen



oder



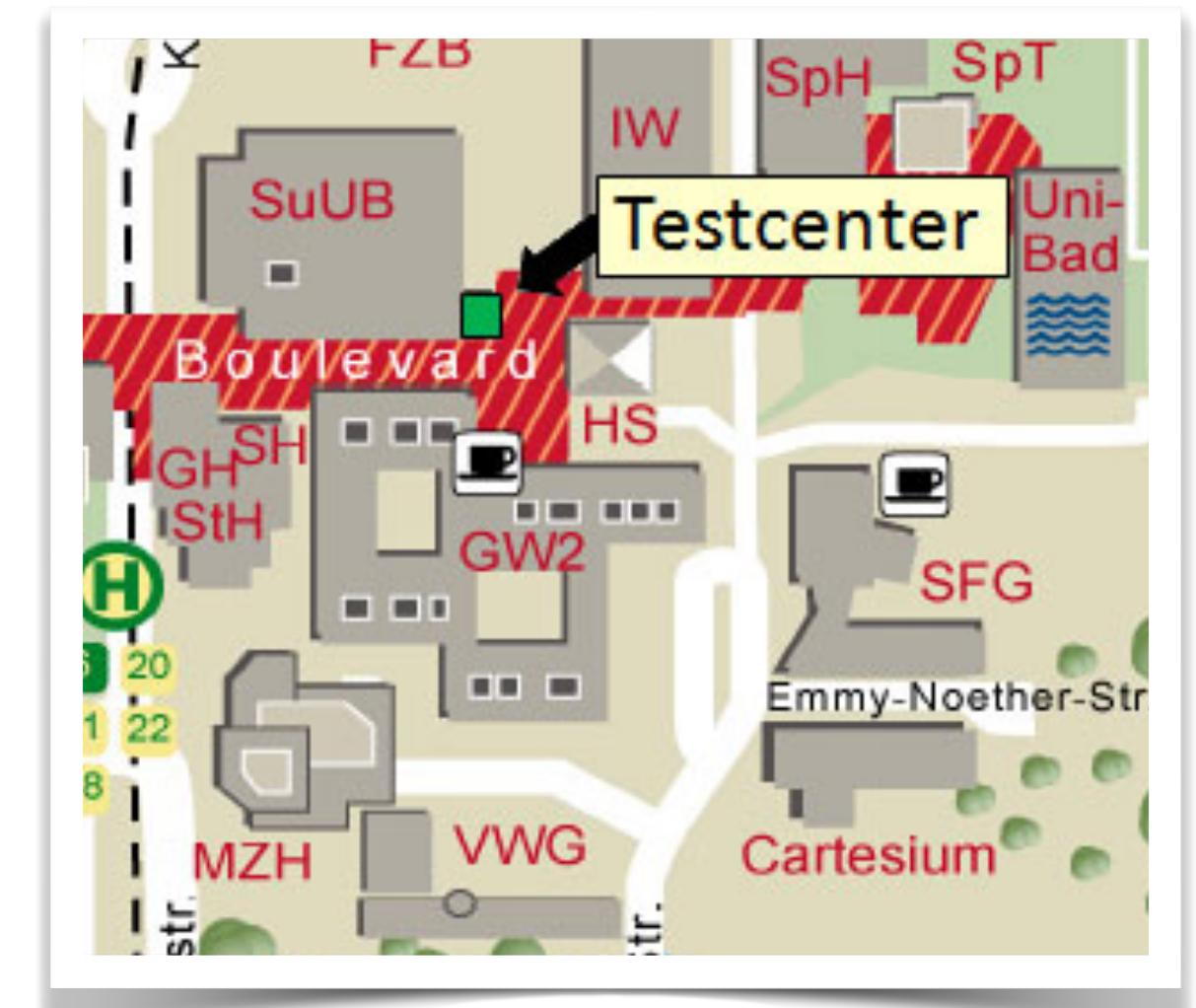
## Zusammenfassung der Konzepte

- **Versionsverwaltung**
- **Arbeitskopie, Index, Repository** (lokal und auf Server)
- **Commit**
- **Pull** und **Push**
- **Konflikte** vermeiden und auflösen

Auf git's!

## Übungsblatt 7 (Übungsklausur)

- 12./14.12, jeweils 10:00, 12:00, 14:00, 16:00, Einlass **15 Min.** vorher
- Einzelbearbeitung am Rechner (1¾ Stunden)
  - Programmieraufgaben: 100%
  - Fragenkatalog: 20%
- Anmeldung unter **Terminvergabe** im Stud.IP bis 04.12 (bevorzugt überlappend mit eigenem Tutoriumstermin)
  - Keine Anmeldung → keine Klausur → keine Punkte für Übungsblatt 7
  - Beachtet <https://www.uni-bremen.de/zmml/kompetenzbereiche/e-assessment/>



## Übungsklausur: Programmieraufgaben

- Einrichtung über **install.cmd** auf dem Desktop, erzeugt Projekt **Uebungsklausur2022** auf dem Desktop
- Aufgaben werden in BlueJ über **Neue Klasse...** erzeugt
  - Kommentare erhalten, damit Zuordnung zu Aufgaben bei automatischer Korrektur möglich
  - Jede Aufgabe nur einmal abgeben (Doppelte entfernen)
- Nutzen **getX()**, **getY()** und **setRotation()** aus **GameObject**
- Jede Aufgabe hat eine **main()**-Methode zum Testen

## Übungsklausur: Programmieraufgaben

- **Aufgabe 1 (50P)**: Attribute, Konstruktor, Verzweigung (ähnlich zu Übungsblatt 3)
- **Aufgabe 2 (25P)**: Aufgabe 1 + Suchschleife, Array, String (ähnlich zu Übungsblatt 5)
- **Aufgabe 3 (25P)**: Aufgabe 2 + sich selbst Lösung überlegen
- **Eigener Code** kann von **Aufgabe n** zu **Aufgabe n+1** kopiert werden (d.h. vorgegebenen Code **nicht** ersetzen)
- Alles, was nicht kompiliert, gibt dabei **0 Punkte**