

Hausaufgabe 1

Die Hausaufgaben müssen von jedem Studierenden einzeln bearbeitet und abgegeben werden. Für die Hausaufgabe sind die aktuellen Informationen vom Blog <https://seblog.cs.uni-kassel.de/ws2122/programming-and-modelling/> zu berücksichtigen.

Abgabefrist ist der 04.11.2021 - 23:59 Uhr

Vorbereitung

Für die Abgabe der Hausaufgaben wird ein Git-Repository genutzt. Dieses wird von jedem Studierenden selbst angelegt und ist ausschließlich für den jeweiligen Studierenden sowie für die Betreuer sichtbar.

Nicht oder zu spät gepushte (Teil-)Abgaben werden mit 0 Punkten bewertet.

Zunächst muss ein Account auf <https://github.com/> angelegt werden (es kann auch ein bereits vorhandener Account genutzt werden). Da Git auch in der späteren Arbeitswelt ein essentielles Tool darstellt, nutze diese Anmeldung für dich und sieh von einem Spaß-Account mit Nutzernamen wie "XxXEdg3L0rdXxX" ab.

Ohne gültigen Account kann der folgende Schritt nicht durchgeführt werden.

Melde dich nun unter <https://classroom.github.com/a/oYjoWHvA> für die Hausaufgabe 1 an. Folge den dort geschilderten Anweisungen, um Zugriff auf das erstellte Repository zu erhalten. Deine Abgaben lädst du, wie in der Übung gezeigt, darin hoch. Es werden nur Text (.txt), Markdown (.md) oder PDF (.pdf) als Dateiformat in den jeweiligen Aufgaben verlangt. Andere Formate werden nicht akzeptiert und folglich mit 0 Punkten gewertet.

Eine Registrierung für die Hausaufgaben ist zwingend notwendig.

Falls du dich noch nicht in dem Google Form (<https://forms.gle/9osQTwveBVMDFNKx8>) angemeldet hast, hole dies jetzt unbedingt nach. Deine Hausaufgaben können ohne Anmeldung nicht bewertet werden!

Ein Abweichen von diesem Schema führt zu einer Nichtbewertung.

Aufgabe 1 - Git (10P)

In dieser Aufgabe sollen die Grundkonzepte "Commit", "Branch", "Checkout", "Push" und "Merge" des Git-Workflows erlernt werden. Im Verlauf der Aufgabe wird ein **Mergekonflikt** verursacht, welcher aufgelöst werden soll. Führe die folgenden Schritte auf dem Repository, welches du in der **Vorbereitung** erstellt hast, aus.

1. Erstelle einen Ordner mit dem Namen "task1". Erstelle innerhalb dieses Ordners eine Datei mit dem Namen "me.txt". Committe und pushe diese Änderung.
2. Erstelle einen Branch mit dem Namen "dev".
3. Führe danach einen Checkout auf den "main"-Branch durch (Wechsle zurück auf den "main"-Branch).
4. Schreibe den folgenden Satz in die Datei:
"Hallo mein Name ist <Emailadresse> <Matrikelnummer> <Nachname>". Committe und pushe die Änderung.
5. Führe einen Checkout auf den "dev"-Branch durch.
6. Schreibe folgenden Satz in die (nun wieder leere) Datei "me.txt": "Hallo mein Name ist albert@uni-kassel.de 4242 Zündorf". Committe und pushe anschließend diese Änderung.
7. Führe einen Checkout auf den "main"-Branch durch.
8. Merge nun den "dev"-Branch in den "main"-Branch (dies sollte einen Konflikt hervorrufen).
9. Committe die Änderung mit dem vorhandenen Konflikt in der Datei (Dies ist normalerweise nicht üblich und ein No-go, aber es verdeutlicht in diesem Fall das Problem).
10. Behebe nun den Mergekonflikt. In der Datei soll am Ende wieder folgender Satz stehen: "Hallo mein Name ist <Emailadresse> <Matrikelnummer> <Nachname>". Committe und pushe abschließend diese Änderung.

Nach Bearbeitung der Aufgabe sollte das Dateisystem des Repositorys wie folgt aussehen.

```
pmws2122-assignment-1-<GitHub-Username>/
├── .git/...
├── .github/..
├── task1/
│   └── me.txt
```

Der Git-Baum sollte mindestens 4 Commits beinhalten. Bei der Bewertung wird vor allem darauf geachtet, dass der Mergekonflikt unaufgelöst committet wurde und dessen Lösung erst im Nachhinein geschieht.

Zur Bearbeitung darf ein beliebiges Git-Tool genutzt werden.

Aufgabe 2 - Abstrakt vs. Konkret (8P)

Diese Aufgabe beschäftigt sich mit dem Unterschied zwischen abstrakten und konkreten Bezeichnungen sowie den Begriffen "Abstrakt" und "Konkret". Betrachte dafür das Wimmelbild aus Abbildung 1. Sollte das Bild zu klein sein, kann die Originaldatei aus der Quelle genutzt werden.



Abbildung 1: Orangerie Wimmelbild

1. Erstelle eine Tabelle mit den Spalten "Abstrakt" und "Konkret". Trage nun **fünf** Beispielpaare in die Tabelle ein. Alle Paare müssen in Abbildung 1 zu sehen sein.
2. Definiere auf der Grundlage aus Aufgabenteil 1 mit eigenen Worten die Begriffe "Abstrakt" und "Konkret".
3. Definiere den Begriff "Beispiel" mit eigenen Worten und stelle hierbei einen Bezug zu den Definitionen aus Aufgabenteil 2 her.

Lege die erstellte(n) Datei(en) in einen neuen Ordner mit dem Namen "task2" in deinem Repository ab. Committe und pushe die Änderungen abschließend auf den [main-Branch](#).

Aufgabe 3 - Textuelle Szenarien (12P)

Erstelle **drei** textuelle Szenarien zu konkreten Spielsituationen des Spiels "Nine men's morris". Die Szenarien sollen in Englisch verfasst sein. Die Regeln des Spiels findest du auf der nächsten Seite vor dem Anhang.

Hinweis: Ein textuelles Szenario besteht IMMER aus einem Titel, einer Startsituation, einer Aktion sowie einer Endsituation. Diese vier Teile sollten sichtbar (durch Farbe und/oder Absatz) voneinander getrennt sein.

Als Hilfestellung liegt ein einfaches Beispielszenario vor, das **nicht** für die Hausaufgabe verwendet/kopiert werden darf:

Title: Moving a man

Start: Alice and Bob are playing "Nine men's morris".
Alice and Bob have already placed all 9 men.
Bob's men are placed at b2, c5, d1, d6, d7, e4, e5, g4, g7.
Alice's men are placed at a7, a1, b4, c4, d3, d2, f6, f4, g1.
It is Alice's turn.

Action: Alice moves a man from the position b4 to a4.

Result: The new position of the moved man is a4. Alice formed a mill and needs to remove an enemy man from the game board.

In der folgenden Hausaufgabe können die Szenarien einiger Studierenden zitiert werden. Solltest du die Verwendung deiner eigenen Szenarien nicht billigen, merke dies bitte in einer Datei namens [README.md](#) an!

Lege die erstellte(n) Datei(en) in einem neuen Ordner mit dem Namen "task3" in deinem Repository ab. Committe und pushe die Änderungen abschließend auf den [main](#)-Branch.

Spielregeln

This semester we create nine men's morris or also known as mill.
(https://en.wikipedia.org/wiki/Nine_men%27s_morris)

The board consists of a grid with twenty-four intersections or points. Each player has nine pieces, or "men", usually coloured black and white. Players try to form 'mills'—three of their own men lined horizontally or vertically—allowing a player to remove an opponent's man from the game. A player wins by reducing the opponent to two pieces (where they could no longer form mills and thus be unable to win), or by leaving them without a legal move.

The game proceeds in three phases:

1. Placing men on vacant points
2. Moving men to adjacent points
3. (optional phase) Moving men to any vacant point when the player has been reduced to three men

Phase 1: Placing pieces

The game begins with an empty board. The players determine who plays first, then take turns placing their men one per play on empty points. If a player is able to place three of their pieces on contiguous points in a straight line, vertically or horizontally, they have formed a mill and may remove one of their opponent's pieces from the board and the game, with the caveat that a piece in an opponent's mill can only be removed if no other pieces are available. After all men have been placed, phase two begins.

Phase 2: Moving pieces

Players continue to alternate moves, this time moving a man to an adjacent point. A piece may not "jump" another piece. Players continue to try to form mills and remove their opponent's pieces as in phase one. A player can "break" a mill by moving one of his pieces out of an existing mill, then moving it back to form the same mill a second time (or any number of times), each time removing one of his opponent's men. The act of removing an opponent's man is sometimes called "pounding" the opponent. When one player has been reduced to three men, phase three begins.

Phase 3: "Flying"

When a player is reduced to three pieces, there is no longer a limitation on that player of moving to only adjacent points: The player's men may "fly" (or "hop", or "jump") from any point to any vacant point.

Anhang

Es folgt eine Auflistung hilfreicher Webseiten und weiterer Erklärungen zu den Themen dieser Hausaufgabe. Die Links sind als Startpunkt zur selbstständigen Recherche gedacht. Das Durcharbeiten der folgenden Quellen ist kein bewerteter Anteil der Hausaufgaben.

Git

- Download: <https://git-scm.com/downloads>
 - Hilfestellung für Linux:
<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
 - Hilfestellung für Mac:
<https://gist.github.com/derhuerst/1b15ff4652a867391f03#file-mac-md>
- Documentation: <https://book.git-scm.com/doc>
 - What is Version Control?:
<https://book.git-scm.com/video/what-is-version-control>
 - Pro Git: <https://book.git-scm.com/book/en/v2>
- A Visual Git Reference: <http://marklodato.github.io/visual-git-guide/index-en.html>
- Git Cheat Sheet:
<https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf>

GitHub

- GitHub: <https://github.com/>
- Getting started with GitHub:
<https://help.github.com/en/categories/getting-started-with-github>
- Beispiel für ein GitHub-Repository, schaut euch z. B. Dateien, Commits, Branches, Issues und Pull requests an: <https://github.com/deepfakes/faceswap>