

Python - Programmmentwurf 2021

Michael Möbius

Voraussetzung

- Jeder muss 4 von 6 Übungsaufgaben gemacht und abgegeben haben

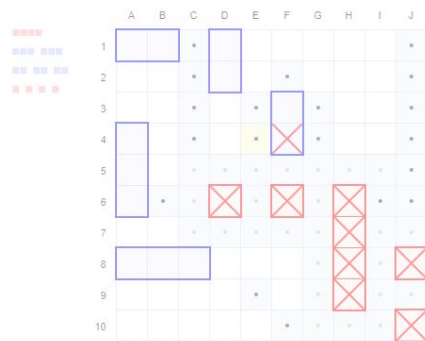
Gruppenarbeit

- Gruppen von 3-4 Studenten
- Jeder Student muss seinen Teil beitragen
- Gruppen:

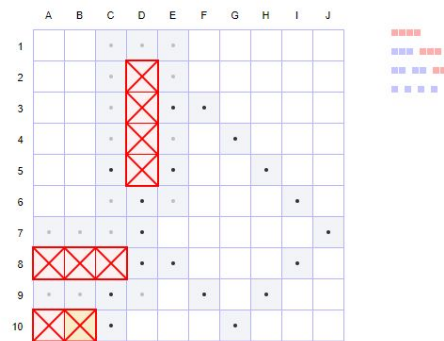
<https://docs.google.com/spreadsheets/d/1TyKvTH7b7eDuXlalkeEVEMptT5kfBe9CoFoRXHCTJAY/edit?usp=sharing>

Schiffeversenken

- https://de.wikipedia.org/wiki/Schiffe_versenken
- 10x10 Felder
- Regeln aus Wikipedia einhalten



Dein Spielfeld



Spielfeld des Gegners
(anderer Spieler)

Regeln (Aus Wikipedia)

Vorbereitung - Platzierung der Schiffe

1. Die Schiffe dürfen nicht aneinander stoßen.
2. Die Schiffe dürfen nicht über Eck gebaut sein oder Ausbuchtungen besitzen.
3. Die Schiffe dürfen auch am Rand liegen.
4. Die Schiffe dürfen nicht diagonal aufgestellt werden.
5. Jeder verfügt über insgesamt zehn Schiffe (in Klammern die Größe):
 - ein Schlachtschiff (5 Kästchen)
 - zwei Kreuzer (je 4 Kästchen)
 - drei Zerstörer (je 3 Kästchen)
 - vier U-Boote (je 2 Kästchen)

Aufgabe - Schiffeversenken

- Zwei Spieler an einem PC
- Spiel aus Datei einlesen und in Datei abspeichern
 - Format kann frei gewählt werden
- Spiel Benutzerinterface
 - Jede Gruppe erstellt eigene Ein- und Ausgabe über **Terminal/Konsole**
 - Anzeige als Feld in der Konsole
- Wenn möglich objektorientiert programmieren
- Automatisierte Tests
- Kleine Bibliotheken können nachinstalliert werden (z.B. für Dateizugriffe,.. keine Game-Engine)
- Lauffähig Windows und Linux (Pfade für Dateien, Konsole clear)

Automatisierte Tests

- Es wird **pro Datei** eine Testabdeckung von mindestens 75% gefordert
 - Selbst überprüfen mit “Coverage” Tool
 - Test müssen sinnvoll sein (validieren)
- Tests müssen selbst entwickelt werden
 - Tests fließen in die Bewertung ein

Aufgabe - Schiffeversenken - 4 Personen

- Gruppen mit 4 Studenten entwickeln zusätzlich einen Computer gegner
- Muss sich auch an Regeln halten
- Muss nicht intelligent sein

Zeitraum

- 6 Wochen Zeit
- 16.03.2021 - 27.04.2021 23:59
- Abgabe über Moodle.
 - Bei Problemen an dhbw@michaelmoebius.de
 - Als Anhang in der Mail (zip)

Umsetzung in Python

- Aktuelle Python version verwenden
- Python mindestens 3.7
- Pylint mindestens 2.4.4

Versionsverwaltung

- Es sollte git verwendet werden
 - **Privates** repository (kein Zugriff für andere Studenten!)
 - Nur lauffähiger Code sollte committed werden
 - Kostenlos z.b. bei Github

Lint

- Alle Lint warnungen sollen behoben werden (Auch kommentare)
- Keine Ausnahmen außer den hier genannten
- Falschmeldungen müssen erklärt werden (Wenn berechtigt, kein Punktabzug)
- Pylint
 - Zeilenlänge 160
 - z.B. mit .pylintrc Datei
 - Keine Dokumentation in Tests nötig
 - In test Dateien:

```
# pylint: disable=C
```
 - In tests darf auf “protected members” zugegriffen werden
 - test.py: W0212: Access to a protected member _board_arr of a client class (protected-access)
 - ```
pylint: disable=protected-access
```

# Dokumentation

- Siehe Bewertung.doc

# Bewertung

- Ihr erstellt eine eigene Abschätzung zu euren Projekt anhand der Tabelle
- Wird mit 40% in die Note Programmieren eingerechnet (Maximal 40 Punkte)
- Die anderen 60 Punkte kommen von Maus/Staudacher

# Tipps

- Pylint über Konsole ausführen
- Keine weiteren Pylint Deaktivierungen verwenden
- Tests sind in dieser Arbeit sehr hoch bewertet
- Achtet auf das schließen von Dateien beim Lesen/Schreiben
- Exceptions bei Dateizugriffen abfangen
- Verwendet innerhalb von Klassen und Methoden den “\_” für private Felder/Methoden