

E) Abgabe



Lade deinen Sourcecode mittels Push in ein GitHub-Repository hoch.

1. Vorbereiten des Repositories in PyCharm

1.1. Version Control aktivieren

Im Menu „VCS“ ist die Option „Enable Version Control Integration ...“. Dadurch wird Git für dieses Projekt eingerichtet.

Alle Dateien mit Sourcecode müssen zum Repository hinzugefügt werden. Wähle dazu die Dateien aus, mache einen Rechtsklick und wähle „Git“ ⇒ „Add“.

1.2. .gitignore

Verschiedene Dateien und Ordner sollen nicht auf GitHub hochgeladen werden. Dazu erstellst du eine Datei `.gitignore` (**Wichtig:** den Punkt am Anfang des Dateinamens nicht vergessen).

Kopiere die folgenden Einstellungen in diese Datei:

```
.idea/  
  
# dotenv  
.env  
  
# virtualenv  
.venv  
venv/  
ENV/  
  
# Installer logs  
pip-log.txt  
pip-delete-this-directory.txt  
  
# Unit test / coverage reports  
htmlcov/  
.tox/  
.coverage  
.coverage.*  
.cache  
nosetests.xml
```

```
coverage.xml
*.cover
.hypothesis/

# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
env/
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
*.egg-info/
.installed.cfg
*.egg

# VS Code
.history/
```

1.3. requirements.txt

Falls du zusätzliche Bibliotheken installiert hast, müssen diese in der Datei `requirements.txt` angegeben werden. Öffne das Terminal in PyCharm und gib diesen Befehl ein:

```
pip3 freeze > requirements.txt
```

Stelle sicher, dass die Datei `requirements.txt` in Git verwaltet wird (git → add).

2. GitHub-Repository

Erstelle ein neues, **privates** Repository auf GitHub. Lade deine Lehrperson als Mitarbeiter in dieses Repository ein. Sie muss mindestens die Rechte haben, um das Repository zu klonen.

Trage den Link zum GitHub-Repository in der Moodle-Aufgabe ein.

3. Commit, Push

3.1. Manage Remotss

In PyCharm musst du den Link zu deinem GitHub-Repository eintragen. Gehe dazu ins Menu „Git“ und wähle „Manage Remotes“.

3.2. Commit, Push

Führe einen Commit deiner neuesten Änderungen durch. Bevor du einen Push durchführst, solltest du immer mit Update Project allfällige Änderungen von GitHub herunterladen.

Bewertungskriterien

Punkte	Kriterium
1	Bot innerhalb der Zeit abgegeben
1	Bot registriert sich korrekt beim Clowder-Service
1	Bot öffnet einen Socket und empfängt Nachrichten
2	Bot kommuniziert korrekt mit der Arena
2	Der Bot verwendet eine Strategie für das Spiel

Es können auch Teilpunkte erteilt werden.

Zusatzpunkte

- 2 / 1 / 0.5 Punkte für die beste 3 Bots
- 2 Punkte: Beitrag zum Clowder-Service
- 2 Punkte: Beitrag zum Arena-Service
- 2 Punkte: Visualisierung des Spielablaufs

Die Punkte für die verschiedenen Services werden je nach Umfang und Qualität der Beiträge verteilt.

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**



Permanent link:
<https://wiki.bzz.ch/modul/m321/kitten/abgabe>

Last update: **2025/04/07 16:27**