Station404 - Verlassene Raumstation

Inhaltsverzeichnis

1. Projektübersicht

Station404 ist ein konsolenbasiertes Spiel, das in Python entwickelt wurde. Das Spiel simuliert die Erkundung einer verlassenen Raumstation, wobei der Spieler verschiedene Bereiche scannen muss, während er mechanischen Fallen ausweicht.

2. Bibliotheken und Versionen

- Python 3.13.1
- Pylint 3.0.3
- Coverage 7.4.1
- MyPy 1.8.0

3. Architektur

Das Projekt folgt einer klaren Trennung von Aufgaben in verschiedene Dateien:

- main.py: Haupteinstiegspunkt und Spielschleife
- gamemap.py: Verwaltung des Spielbretts und Zellenlogik
- logic.py: Spiellogik und Regelimplementierung

4. Funktionsbeschreibung

main.py

- main_menu(): Hauptmenüfunktion zum Starten des Spiels oder Anzeigen der Anleitung
- main(): Hauptfunktion für den Gameplay-Loop und zur Verwaltung der User-Inputs

gamemap.py

- Map: Klasse zur Verwaltung des Spielbretts
  - **init**(): Initialisiert ein leeres Spielbrett
  - make_map(size): Erstellt ein quadratisches Spielbrett der angegebenen Größe
  - print_map(grid): Zeigt das Spielbrett mit Koordinaten und Rahmen an

logic.py

- place_traps(spaceship): Platziert zufällig Fallen auf dem Spielbrett
- scan_field(x, y, spaceship, solution): Scannt ein Feld und zeigt die Anzahl benachbarter Fallen an
- mark_field(x, y, spaceship): Markiert ein Feld mit einem X
- get_traps(spaceship): Gibt eine Liste der Koordinaten aller Fallen zurück
- get_neighbor_traps(spaceship, x, y): Zählt die Anzahl der Fallen in benachbarten Feldern
- check_win(spaceship, solution): Überprüft, ob alle Fallen korrekt markiert wurden
- welcome(): Zeigt den Willkommensbildschirm mit ASCII-Art an
- instructions(): Zeigt die Spielanleitung und Regeln an
- clear_terminal(): Leert das terminal

5. Benutzeroberfläche

Das Spiel wird vollständig über die Konsole gesteuert und bietet folgende Elemente:

Spielbrett

- Quadratisches Raster zur Anzeige des Spielfelds
- Rahmen aus ASCII-Zeichen für bessere Übersichtlichkeit

Spielsymbole

- [0] : Unerforschtes Feld
- [?] : Gescanntes Feld mit Zahl (1-8) für benachbarte Fallen

- [ ] : Gescanntes Feld ohne benachbarte Fallen
- [X] : Markiertes Feld

Benutzerinteraktionen

- 1 Anzeigen des Spielfelds
- 2 Scannen eines Feldes
- 3 Markieren eines Feldes

6. Programmablauf

1. Spielstart
    - Generierung des Spielbretts
    - Platzierung der Fallen
2. Spielzug
    - Anzeige des aktuellen Spielstands
    - Eingabeaufforderung
    - Verarbeitung des Zugs
3. Spielende
    - Bei Treffer einer Falle
    - Bei Aufdecken aller sicheren Felder oder Markieren aller Fallen

7. Testergebnisse

Getestet wurde unter macOS 15.3.2 und Ubuntu 24.10

Unittest Coverage

```
Name                    Stmts   Miss  Cover
-----------------------------------------
source/gamemap.py          18      0   100%
source/logic.py            53     13    75%
source/main.py             72     66     8%
tests/test_gamemap.py      32      0   100%
tests/test_logic.py        34      0   100%
tests/test_main.py          7      0   100%
-----------------------------------------
```

Die Coverage der main.py ist niedrig, da diese nur User Input und den haupt Gameplay-Loop handhabt und Tests daher nicht sinnvoll sind.

Pylint Bewertung

```
source/main.py: 10.00/10
source/gamemap.py: 10.00/10
source/logic.py: 10.00/10
```

MyPy Überprüfung

Alle Dateien haben die MyPy-Überprüfung erfolgreich bestanden:

```
Success: no issues found in 3 source files
```