

Space Survival

Spiel Beschreibung:

In dem Spiel "Space Survival" sind sie auf einer verlassenen Raumstation gestrandet. Sie müssen die einzelnen Räume der Raumstation aufdecken ohne einen Raum mit fallen oder Aliens zu öffnen. Das Spielfeld ist ein Random generierter Raum mit Random platzierten Fallen. Sie haben Gewonnen wenn es keinen Raum ohne fälle mehr zu aufdecken gibt.

Nutzerinterface Beschreibung:

Der Homescreen:

Im Startmenü können sie auswählen ob sie das spiel Benden , ein Neues Spielfeld Generieren, das Spiel Starten, oder die Credits Anzeigen wollen.

Das Spiel:

Sie werden ein Spielfeld mit variabler Größe gezeigt bekommen, dann können sie über die am Rand angezeigten Koordinaten das Feld auswählen das sie aufdecken wollen. Wenn in dem Feld eine Falle War wird ihnen der Restart screen angezeigt und sie können auswählen ob sie das spiel neu starten wollen oder zum Homescreen zurückkehren wollen. Wenn keine fälle in diesem Feld ist wird eine zahl angezeigt die aussagt wie viel Felder der direkt anliegenden Felder eine fälle haben

Restart Screen:

auf dem Restart Screen kann man durch die Eingabe von "Y" das speil mit einem neuen Spielfeld neu Starten und wenn man "N" eingibt, kommt man wieder zum Homescreen.

Winning screen:

auf dem Winning screen können sie auswählen, ob sie noch eine Runde spielen wollen oder auf das Main Menü zurück gehen wollen

Genutzte Bibliotheken:

- pylint 3.3.6
- coverage 7.8.0
- mypy 1.15.0

Programmablauf:

Wenn sie das Programm starten wird als erstes die main Funktion in der game_start Datei in der Klasse Hub aufgerufen diese ruft dann die Funktion start auf in der ja nach Input ausgewählt wird was gemacht wird. Wenn die 1 ausgewählt wird wird die Funktion für den clear_screen aufgerufen und danach die Funktion game in der Klasse Round in der Datei play. In der Funktion game wird die Funktion play in einer While schleife aufgerufen. In der Funktion play wird dann die spiel Mechanik ausgeführt. Wenn man gewinnt, wird die Funktion win in der Klasse hub in der Datei game_start aufgerufen in der ein Winning screen angezeigt wird und in der man sich entscheiden kann, ob man nochmal spielen will oder zurück zum Hauptmenü will. Wenn man eine Falle ausgelöst hat wird die Funktion restart in der Klasse hub aufgerufen in der man sich entscheiden kann, ob man das Spiel neu starten will oder zurück zum Hauptmenü will. wenn man sich im Hauptmenü die 2 auswählt, wird die Funktion clear_screen aufgerufen und danach wird ein Input abgefragt welche schwierichkeit man haben will und der Input wird dann mit dem aufruf der Funktion map in der Klasse Map in der Datei world_gen aufgerufen und das Spielfeld aktualisiert. Wenn man sich im Hauptmenü die 3 auswählt, wird das Spiel mit dem Aufruf sys.exit beendet. wenn man sich im Hauptmenü die 4 auswählt, wird die Funktion display_credits aufgerufen die dann die Credits des Spiels anzeigt und mit einem Tastendruck einen ins Hauptmenü zurückschickt. wenn man sich im Hauptmenü die 6 auswählt wird die Funktion cheat_mode in der Klasse Round in der Datei play aufgerufen in mit der dann das Spielfeld aufgedeckt angezeigt wird.

Architekturbeschreibung:

1. Module und ihre Verantwortlichkeiten

game_start

- **Verantwortung:**
 - Das Hauptmodul, das den Einstiegspunkt des Spiels bereitstellt.
 - Es initialisiert die Hauptmenü-Logik über die Klasse Hub.
- **Hauptbestandteile:**
 - Hub-Klasse: Enthält Methoden für das Hauptmenü, Neustart, Sieg-/Verlustbildschirme sowie die Anzeige von Credits.

play

- **Verantwortung:**
 - Implementiert die Spielmechanik, einschließlich der Logik für das Aufdecken von Feldern, Überprüfen von Fallen und Verwalten des Spielfortschritts.
- **Hauptbestandteile:**
 - Round-Klasse:
 - play(): Kernspielmechanik (z. B. Eingabevalidierung, Fallenüberprüfung, Spielfeldaktualisierung).
 - cheat_mode(): Zeigt die Position der Fallen an.

- `game()`: Startet das Spiel in einer Endlosschleife.

world_gen

- **Verantwortung:**
 - Generiert das Spielfeld basierend auf dem Schwierigkeitsgrad.
 - Platziert Fallen und berechnet sichere Felder.
- **Hauptbestandteile:**
 - Map-Klasse:
 - `map(difficulty)`: Erstellt eine Karte mit einer bestimmten Größe und Fallenverteilung basierend auf dem Schwierigkeitsgrad.

2. Interaktion zwischen den Modulen

Modul	Abhängigkeiten	Beschreibung der Interaktion
game_start	play, world_gen	Ruft Methoden aus <code>play.Round</code> und <code>world_gen.Map</code> auf, um das Spiel zu starten oder neu zu initialisieren.
play	game_start, world_gen	Nutzt die Spielfeldlogik aus <code>world_gen.Map</code> und interagiert mit dem Hauptmenü (<code>game_start.Hub</code>).
world_gen	Keine	Ist unabhängig und stellt Spielfeldgenerierungslogik bereit.

Tests:

Pylint:

main.py:

```
(.venv1) C:\Users\adria\PycharmProjects\Python-curs-dhbw>Last_Project\source>pylint main.py
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

play.py:

```
(.venv1) C:\Users\adria\PycharmProjects\Python-curs-dhbw>Last_Project\source>pylint play.py
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

world_gen.py:

pylint: disable=too-few-public-methods

Wurde deaktiviert da der Programmteil mit mehr Funktionen geplant war aber dann doch nur eine gebraucht wurde

```
(.venv1) C:\Users\adria\PycharmProjects\Python-curs-dhbw>Last_Project\source>pylint world_gen.py

-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

game_start.py:

```
(.venv1) C:\Users\adria\PycharmProjects\Python-curs-dhbw>Last_Project\source>pylint game_start.py

-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

MyPy:

main.py:

```
(.venv1) C:\Users\adria\PycharmProjects\Python-curs-dhbw>Last_Project\source>mypy main.py
Success: no issues found in 1 source file
```

play.py:

```
(.venv1) C:\Users\adria\PycharmProjects\Python-curs-dhbw>Last_Project\source>mypy play.py
Success: no issues found in 1 source file
```

world_gen.py:

```
(.venv1) C:\Users\adria\PycharmProjects\Python-curs-dhbw>Last_Project\source>mypy world_gen.py
Success: no issues found in 1 source file
```

game_start.py:

```
(.venv1) C:\Users\adria\PycharmProjects\Python-curs-dhbw>Last_Project\source>mypy game_start.py
Success: no issues found in 1 source file
```

Unittest:

File ▲	statements	missing	excluded	coverage
C:\Users\adria\PycharmProjects\PythonProject\source\game_start.py	78	29	0	63%
C:\Users\adria\PycharmProjects\PythonProject\source\play.py	66	26	0	61%
C:\Users\adria\PycharmProjects\PythonProject\source\world_gen.py	23	0	0	100%
test_game_start.py	65	15	0	77%
test_main.py	19	2	0	89%
test_play.py	35	7	0	80%
test_world_gen.py	58	1	0	98%
Total	344	80	0	77%