**Python - Programmentwurf**

**Team Mitglieder:**

|  |  |
| --- | --- |
| **Name** | **e-Mail (DHBW)** |
| Thomas Gingele |  |
| Lukas Eisele | eisele.lukas-it21@it.dhbw-ravensburg.de |
| Jannik Künstler |  |
| Felix Schladt |  |
|  |  |

**Verwendete Python Version: 3.9.7, 3.10.4**

**Verwendete Pylint Version: 2.13.7**

**Verwendete nachinstallierte Bibliotheken: keine**

**Beschreibung des Dateiformats zum Speichern des Spiels:**

* {"turns": 2, "player\_0": {"name": "Lukas", "dices": [1, 1, 2, 5, 1], "flags": [false, true, true, true, true, false, false, true, true, false, false, false, true], "scores": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], "active": true}, "player\_1": {"name": "Thomas", "dices": [6, 2, 4, 6, 5], "flags": [false, false, true, true, true, true, false, false, true, false, true, false, true], "scores": [0, 0, 0, 6, 0, 10, 12, 28, 0, 0, 0, 25, 0, 40, 0, 13, 106], "active": false}}
* Die Datei wird im json-Format gespeichert. Jeder Spieler besteht aus einem dictionary. In diesem dictionary steht jeweils der Name, die geworfenen Augenzahlen der Würfel und der Spielstand.

**Beschreibung des Benutzerinterfaces:**

* Wie sieht die Ein- und Ausgabe aus?

**Beschreibung der Architektur:**

* Grobe Architektur beschreiben (Textuell, oder Diagramm)

→ ./doc/depency\_diagramm.png

→ ./doc/structure\_diagramm.png

* Ca. halbe Seite bis eine Seite

**Dokumentation von einem kompletten Spielablauf:**

* Kopie der kompletten Ausgabe auf der Konsole. Evtl. eigene Datei

→ ./doc/played\_game\_record.log

* + Datei ist mittels tee erstellt → kopie von stdout
  + Ausgabe ohne color codes auf stdout:
    - cat ./doc/played\_game\_record.log > /dev/tty

Kann jedoch falsch aussehen, falls sich die terminal größe unterscheidet

**Log von den Tests:**

* Alle Tests über Konsole ausführen und Ausgabe hier einfügen (oder extra Datei)

**Bewertung der Testergebnisse:**

* Grund für fehlschlagende Tests

**Code-Coverage Ausgabe:**

* Tests mit Coverage ausführen und Ausgabe hier einfügen

**Bewertung der Coverage und Sinnvollheit der Tests:**

* Gewünscht ist eine Coverage von min 75% (pro Datei)
  + Wurde dies erreicht? Wenn nein, warum nicht?
* Begründet warum ihr euch sicher seid alles mit Tests abgedeckt zu haben
* Fehlerfälle in den Tests überprüft?

**Bewertung der Fehlersicherheit:**

* Werden Fehleingaben korrekt abgefangen?
  + Nenne 2 Beispiele wo ein Fehler abgefangen wird

**Pylint Ausgabe:**

* Pylint auf der Konsole Ausführen und Ergebnis einfügen

**Bewertung der Code-Qualität/Lesbarkeit:**

* Begründet noch vorhandene Pylint Warnungen
* Begründet warum euer Code gut lesbar ist

**Bewertungstabelle:**

* Ausgefüllte Bewertungstabelle