

Cipher Madness: Can you guess the output?

Cipher Madness: Can you guess the output? is a text based adventure game that tests your knowledge of encryption functions.

In diesem Spiel geht es darum, durch Lösen von Verschlüsselungen das Geheimnis der Weltformel zu lüften.

Dabei bewegt sich ein Charakter durch verschiedene Räume und wird von einem Tagebuch voller Hinweise begleitet, welche ein Wissenschaftler hinterlassen hat.

Das Projekt beinhaltet Verschlüsselungsalgorithmen, Optionsmenus, die Möglichkeit, sich durch Räume zu bewegen, Kollision mit Wänden und Spezialfeldern, Stories, die in der Konsole ausgegeben werden, Konsoleneffekte, Geheimnisse, einen Cheat-Modus, Einlesen von Dateien

Installationsschritte zum Ausführen des Docker Image aus der GitHub Container Registry

- Installation von Docker Desktop oder einer ähnlichen Docker Umgebung
- Starten des Docker Daemon (indem die installierte Docker Desktop Anwendung gestartet wird)
- Öffnen einer Kommandozeile / eines Terminals (cmd, powershell, bash, etc.)
- Ausführen von `docker pull ghcr.io/dennis960/haskell-project:latest`
- Ausführen von `docker run -it ghcr.io/dennis960/haskell-project:latest`
- (Für mehr Informationen <https://docs.docker.com/docker-hub/quickstart/>)
- Weitere Installationsvarianten stehen in der README.md

Bewertung des Codes

☒ Vollständig ☐ Teilweise ☒ Nicht vorhanden

- ☒ Listen (z.B. in gameLoopElements in Story.hs)
- ☒ list comprehension (z.B. in randomScreen in ConsoleFX.hs)
- ☒ Funktionen mit pattern matching (z.B. in printStory in GameLoopElement.hs)
- ☒ Funktionen mit guards (z.B. in caesarCipher in Cipher.hs)
- ☒ Rekursive Funktionen (z.B. run in Main.hs)
- ☒ Funktionen höherer Ordnung wie map, filter, fold (z.B. in printRoom in Room.hs)
- ☐ Fehlerbehandlung ggf. mit Either oder Maybe (z.B. in loadRoom - findPlayerPosition in Room.hs, allerdings ohne Either oder Maybe)
- ☒ Eigene Datentypen (z.B. GameLoopElement in GameLoopElement.hs)
- ☒ Ein/Ausgabe mit IO-Monaden (z.B. in putTextNL in Typer.hs)
- ☒ Modularisierung (Siehe Ordnerstruktur)
- ☒ Übersichtlicher Code (ggf. let / where verwendet) (Überall)
- ☒ wichtigste Teile des Codes dokumentiert (Die wichtigsten Funktionen, die von außen zugreifbar sind, sind dokumentiert, außerdem gibt es eine Dokumentation der Struktur in der README.md)