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.

It is written in Haskell.

Installation

Recommended installation using image on GitHub Container Registry

- Install Docker Desktop or a similar Docker environment
- Run the docker daemon (by starting the installed Docker Desktop application)
- Open a command line terminal (cmd, powershell, bash, etc.)
- Run `docker pull ghcr.io/dennis960/haskell-project:latest`
- Run `docker run -it ghcr.io/dennis960/haskell-project:latest`
- (For more information see https://docs.docker.com/docker-hub/quickstart/)

Manual installation using Visual Studio Code and Docker

- Clone the repository
- Install the devcontainer extension for Visual Studio Code
- Open the project in the devcontainer (F1 -Dev Containers: Open Folder in Container) (Might take a while the first time)
- Run the "build and run" task (F1 -Tasks: Run Task -build and run)

Manual installation without installing Docker (Only Linux)

- Clone the repository
- Install the Haskell Tool Stack
- Open the project in Visual Studio Code
- Run the build and run task or
- Run `mkdir -p dist;ghc -o dist/main -isrc src/Main.hs -outputdir build;cp -r src/rooms dist/.;cd dist;./main` in the project root directory
- Run `./dist/main`
- If that does not work: `chmod +x dist/main;./dist/main`

Development

Using Linux

- Install the Haskell Tool Stack
- Open the project in Visual Studio Code
- Run the "build and run" task (F1 -Tasks: Run Task -build and run)

Using Windows

- Install Docker Desktop or a similar Docker environment

- Run the docker daemon by starting the installed Docker Desktop application
- Install the devcontainer extension for Visual Studio Code
- Open the project in the devcontainer (F1 -Dev Containers: Open Folder in Container) (Might take a while the first time)
- Run the "build and run" task (F1 -Tasks: Run Task -build and run)

Special notes for updating the README

- The README can be converted to a pdf using the yzane.markdown-pdf extension for Visual Studio Code

Project structure and how to add new content

Room

- Create new rooms in the src/rooms directory. Rooms are loaded by name, room_passage, room_other_name, etc.
- New Room-Cells can be added inside src/Room.hs. The Room-Cells need to correspond to the symbols used in the room files.

Encryption functions

- Add new encryption functions in src/Cipher.hs

Story

- Add new story elements in src/Story.hs inside the gameLoopElements array. The gameLoopElements array defines the order in which stories, terminals and rooms are shown.
- StorySecretItem needs a secret key. The secret key is the String that the user needs to input to progress.
- The secret key can be used inside the storyText by writing `$SECRET`. It gets encrypted with the specified encryption function.

GameLoopElement

- Specify GameLoop Element types here, as well as useful functions for printing and managing the elements.

KeyEvents

- Add new key events in src/KeyEvents.hs. Currently contains the key events for moving the player and fast forwarding the story-telling.

Typer

- Used for printing out text to the console at the specified typing speed.

StringReplace

- Contains a helper function for replacing substrings. Used for replacing `$SECRET` in the story.

Main

- Contains the main function. Loads the rooms, the story and the key events and starts the game.
- Runs the main game loop.

OptionMenu

- Contains functions for printing out and letting the user select options.

ConsoleFX

- Contains functions for printing special effects.

Other files

- /.devcontainer: Contains the devcontainer configuration for Visual Studio Code
- /.Visual Studio Code: Contains the task for building and running the game
- .git*: Git configuration
- README.md: This file

Temporary files

- /build: Contains build files produced by the build task
- /dist: Contains the compiled game executable and the rooms directory