CS2006 Haskell 1: Text Adventure - Group Report

Submitted by Group 1 (190020857, 190022658, 200012696) to Edwin Brady.

This project is an implementation of a simple text adventure, as laid out the project specification. Namely:

- The code stubs in `Actions.hs` have been implemented.
  - Movement between rooms is possible.
  - `get`, `drop`, and other actions have been implemented.
  - Helper commands, including a general `?` (beyond the specification), have been implemented.
- The game world has been extended.
  - A living room, bathroom, and porch have been added (see `img/Map.png`).
  - New puzzles were added:
    - Glasses need to be collected and worn before the text prompts can be read properly.
    - You need to brush your teeth in the bathroom before you can have your morning coffee.
    - You need to put your mask on before you can leave the porch.
- Object, direction, and command strings have been replaced by dedicated data types.
- QuickCheck has been used to perform tests on necessary functions
- The input is accepted using Haskeline (including history and tab completion), and processed using `Parsing.hs`.
- The parser accepts two commands at the same time, separated by "and" or "then".
- `load` and `save` functions were implemented.

In summary, all the Basic, Easy, Medium and Hard requirements are met.

In terms of team organisation, we had regular meetings to pick out new tasks and targets, which were then added to Microsoft Planner to keep track of progress. Between meetings, people would assign tasks to themselves and complete them, and intermediate progress reports could be added to the task in Planner to further break down the work if needed, or avoid two people doing the same thing. The provided submission is a Git repository, so precise breakdown of lines of code written can be found by running `git log` or `git blame` (note though that commit identities have not been anonymised -- they contain names and email addresses instead of matriculation numbers).

Instructions for running the game can be found in `README.md`. Instructions to complete the game can be found in `CHEAT.md`.

Design

Haskeline

Haskeline is used for input and output in the program. This means that control keys like backspace work correctly (unlike in standard Haskell `IO` in GHCi), and the up and down arrows can be used to browse the history. There is also support for tab completion of commands, based on code by C. A. McCann.

Puzzles

An extra puzzle was added to the start of the game, which involves the player picking up and wearing their **glasses**. When not wearing glasses (either because they've not been put on yet, or because the player has removed them), all program output (excluding the input prompt) is made fuzzy by replacing a selection of characters with `▒`. Initially, this was going to be done randomly, but since all Haskell functions are pure this randomness would require extra overhead which would make the code structure more complicated. This was deemed unnecessary, so instead the messages are always fuzzed in the same way -- half of alphanumeric characters are replaced with the fuzz symbol, with the rest and any punctuation remaining unchanged. The fuzzing is implemented using `putStrFuzzy` and `putStrLnFuzzy` which are abstractions on top of `outputStr`, taking both the string to output and the current game state. If the game state indicates that the player is blind, the string is fuzzed before being outputted.

Another puzzle was also added to the game, which meant the player had to go to the **bathroom** to **brush their teeth** before the player can drink their coffee. Attempting to drink the coffee without completing this puzzle would result in the player being prompted to brush their teeth. In order to complete this puzzle,the player must go to the bathroom and `get` the **toothbrush** and **toothpaste**. Then the user must `apply` the **toothpaste** and `brush` their teeth. If the player misses any of these steps, or executes them in the wrong order, then they are prompted to follow the correct procedures. We had thought of adding this feature to the game to make it more realistic and enhance the user experience.

Another additional puzzle involves the player needing a **key** to unlock and open the door. The key is in a new room, the **lounge**. The player must `get` the **key** and `unlock` the front door from the **hall**. The player can only unlock the door if they possess the key. The player can then `open` the door, only if they have drunk the coffee.The front door then leads to the **porch**

The final extra puzzle requires the player to wear a **mask** before they can leave another new room, the **porch**. The mask starts hidden in the player's inventory, instead of in a room. To add some realism to the game, the player can't drink coffee while their mask is on.

Save and Load

We have decided to implement `save` and `load` by using Aeson library for JSON. When saving, the GameData is turned into a JSON string and stored in a file of the player's choice, using `writeFile`. When loading, the JSON string is read from the file of the user's choice, and converted to GameData. Initialy, we thought of utilising the `show` and `read` functions to convert the GameData to String object. But we decided against this, as `Show` already had an instance in the program and we did not want to manipulate that any further. Also, the choice of external library to use JSON was further motivated by the fact that the solution will be a lot simpler and easier to read.

Parsing

We switched to using `Parsing.hs` and extended it to `WorldParser.hs` to parse user input. The `process` function was shifted here (it seemed like it belonged here). Here, there are `Parser` functions for `Command', Action', Direction', Object`. With each `Parser` function, there is a function which takes in a String and returns a `Maybe` data. Here, we decided to not take `Object` argument for all `Action'` constructors, because it didn't make sense to make a "teeth" object, and since `Go` had to have a `Direction'` argument while the others had to be `Object`. This made us make different functions to "help" the `Action'` parser. The system also accepts two commands at once, split by "and" or "then". This is implemented by a `Parser` function that returns a `(Action', Action')`. A late idea was to make a function that would return `[Action']` and modify the process function to execute all of them. Using the output state as input state for the next command and storing all the output string in a list to be displayed together.

Testing

QuickCheck

We have implemented several **QuickCheck** tests. We decided against writing any QuickCheck tests for `ActionDir`, `ActionObj` and `Command` types. This choice was made due to the fact that the functions of these types require a `GameData` type as input. Randomly generating GameData variables did not see like the most efficent way to test these functions. Instead, we thought of just using `initState` as the `GameData` and randomly generate the other variables need for testing the function. We also decided against doing this as manual inputting all the possible input, in each game state, was a lot more suitable in testing these functions. Hence, we have tested functions of these types manually covering all the inputs.

Clean run

```
*Main> main
You a▒e i▒ you▒ ▒e▒▒oom.
▒o ▒▒e ▒o▒▒▒ is a ki▒c▒e▒. ▒o ▒▒e wes▒ is a ▒a▒▒▒oom.

You ca▒ see: a ▒ai▒ o▒ g▒asses, a co▒▒ee mug
What now? get glasses then wear glasses

g▒asses is ▒u▒ i▒ i▒▒e▒▒o▒y
Wow!

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.

You can see: a coffee mug
What now? get mug

mug is put in inventory

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.
What now? go west

Moved

You are in the bathroom.
To the east is your bedroom.

You can see: a toothbrush, toothpaste
What now? get toothbrush and get toothpaste

toothbrush is put in inventory
toothpaste is put in inventory
```

```
You are in the bathroom.
To the east is your bedroom.
What now? apply toothpaste

Toothpaste applied to brush

You are in the bathroom.
To the east is your bedroom.
What now? brush teeth

Your teeth are shining

You are in the bathroom.
To the east is your bedroom.
What now? go east

Moved

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.
What now? go north

Moved

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.

You can see: a pot of coffee
What now? get coffee then pour coffee

coffee is put in inventory
Coffee Poured

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.
What now? drink coffee

You feel energized

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.
What now? go west

Moved

You are in the hallway. The front door is closed.
To the east is a kitchen. To the south is a lounge.
What now? go south

Moved

You are in the lounge.
To the north is a hallway.
```

```
You can see: a key
What now? get key

key is put in inventory

You are in the lounge.
To the north is a hallway.
What now? go north

Moved

You are in the hallway. The front door is closed.
To the east is a kitchen. To the south is a lounge.
What now? unlock door then open door

Unlocked the door
Opened the door

You are in the hallway. The front door is open.
To the east is a kitchen. To the south is a lounge. You can go outside.
What now? go outside

Moved

You are standing on the porch. The door is closed.
You can go back inside.
What now? wear mask

Mask worn

You are standing on the porch. The door is closed.
You can go back inside.
What now? open door

Opened the porch door

You are standing on the porch and wearing your mask.
You can go back inside. You can go out into the street.
What now? go outside

Moved

Congratulations, you have made it outside.
```

Not so clean run

```
You a▒e i▒ you▒ ▒e▒▒oom.
▒o ▒▒e ▒o▒▒▒ is a ki▒c▒e▒. ▒o ▒▒e wes▒ is a ▒a▒▒▒oom.

You ca▒ see: a ▒ai▒ o▒ g▒asses, a co▒▒ee mug
What now? go west
```

```
Mo▒e▒

You a▒e i▒ ▒▒e ▒a▒▒▒oom.
▒o ▒▒e eas▒ is you▒ ▒e▒▒oom.

You ca▒ see: a ▒oo▒▒▒▒us▒, ▒oo▒▒▒as▒e
What now? go east

Mo▒e▒

You a▒e i▒ you▒ ▒e▒▒oom.
▒o ▒▒e ▒o▒▒▒ is a ki▒c▒e▒. ▒o ▒▒e wes▒ is a ▒a▒▒▒oom.

You ca▒ see: a ▒ai▒ o▒ g▒asses, a co▒▒ee mug
What now? get glasses and wear glasses

g▒asses is ▒u▒ i▒ i▒▒e▒▒o▒y
Wow!

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.

You can see: a coffee mug
What now? go north

Moved

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.

You can see: a pot of coffee
What now? get coffee

coffee is put in inventory

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.
What now? co south

I don't understand

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.
What now? go south

Moved

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.

You can see: a coffee mug
What now? ?
```

```
 Actions:
         go get drop pour examine drink open unlock wear remove apply brush

 Commands:
         ? inventory load save quit

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.

You can see: a coffee mug
What now? get mug

mug is put in inventory

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.
What now? pour coffee

Coffee Poured

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.
What now? drink coffee

Brush your teeth before you drink coffee

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.
What now? drop coffee

coffee has been dropped

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.

You can see: a pot of coffee
What now? go west

Moved

You are in the bathroom.
To the east is your bedroom.

You can see: a toothbrush, toothpaste
What now? brush teeth

Please attain toothbrush and paste

You are in the bathroom.
To the east is your bedroom.

You can see: a toothbrush, toothpaste
What now? get toothbrush
```

```
toothbrush is put in inventory

You are in the bathroom.
To the east is your bedroom.

You can see: toothpaste
What now? brush teeth

Please attain toothbrush and paste

You are in the bathroom.
To the east is your bedroom.

You can see: toothpaste
What now? get toothpaste

toothpaste is put in inventory

You are in the bathroom.
To the east is your bedroom.
What now? brush teeth

Please apply "toothpaste" to the brush

You are in the bathroom.
To the east is your bedroom.
What now? apply toothpaste

Toothpaste applied to brush

You are in the bathroom.
To the east is your bedroom.
What now? brush teeth

Your teeth are shining

You are in the bathroom.
To the east is your bedroom.
What now? drop toothbrush and drop toothpaste

toothbrush has been dropped
toothpaste has been dropped

You are in the bathroom.
To the east is your bedroom.

You can see: toothpaste, a toothbrush
What now? go north

Unknown location

You are in the bathroom.
To the east is your bedroom.
```

```
You can see: toothpaste, a toothbrush
What now? go east

Moved

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.

You can see: a pot of coffee
What now? go north

Moved

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.
What now? go west

Moved

You are in the hallway. The front door is closed.
To the east is a kitchen. To the south is a lounge.
What now? open door

The door must be unlocked first

You are in the hallway. The front door is closed.
To the east is a kitchen. To the south is a lounge.
What now? unlock door

Have you lost your key again?

You are in the hallway. The front door is closed.
To the east is a kitchen. To the south is a lounge.
What now? go south

Moved

You are in the lounge.
To the north is a hallway.

You can see: a key
What now? get key

key is put in inventory

You are in the lounge.
To the north is a hallway.
What now? go north

Moved

You are in the hallway. The front door is closed.
To the east is a kitchen. To the south is a lounge.
```

```
What now? open door

The door must be unlocked first

You are in the hallway. The front door is closed.
To the east is a kitchen. To the south is a lounge.
What now? unlock door

Unlocked the door

You are in the hallway. The front door is closed.
To the east is a kitchen. To the south is a lounge.
What now? open door

You need energy

You are in the hallway. The front door is closed.
To the east is a kitchen. To the south is a lounge.
What now? drink coffee

You feel energized

You are in the hallway. The front door is closed.
To the east is a kitchen. To the south is a lounge.
What now? drop mug

mug has been dropped

You are in the hallway. The front door is closed.
To the east is a kitchen. To the south is a lounge.

You can see: a coffee mug
What now? open door

Opened the door

You are in the hallway. The front door is open.
To the east is a kitchen. To the south is a lounge. You can go outside.
What now? go outside

Moved

You are standing on the porch. The door is closed.
You can go back inside.
What now? open door

Don't forget your mask

You are standing on the porch. The door is closed.
You can go back inside.
What now? go inside

Moved
```

```
You are in the hallway. The front door is open.
To the east is a kitchen. To the south is a lounge. You can go outside.
What now? inventory

You are carrying:
a key
a mask
You are wearing glasses

You are in the hallway. The front door is open.
To the east is a kitchen. To the south is a lounge. You can go outside.
What now? wear mask

Mask worn

You are in the hallway. The front door is open.
To the east is a kitchen. To the south is a lounge. You can go outside.
What now? drop glasses

No such item

You are in the hallway. The front door is open.
To the east is a kitchen. To the south is a lounge. You can go outside.
What now? remove glasses

Glasses removed and placed in your inventory

You a▒e i▒ ▒▒e ▒a▒▒way. ▒▒e ▒▒o▒▒ ▒oo▒ is o▒e▒.
▒o ▒▒e eas▒ is a ki▒c▒e▒. ▒o ▒▒e sou▒▒ is a ▒ou▒ge. You ca▒ go ou▒si▒e.
What now? wear glasses

Wow!

You are in the hallway. The front door is open.
To the east is a kitchen. To the south is a lounge. You can go outside.
What now? go outside

Moved

You are standing on the porch. The door is closed.
You can go back inside.
What now? open door

Opened the porch door

You are standing on the porch and wearing your mask.
You can go back inside. You can go out into the street.
What now? go outside

Moved

Congratulations, you have made it outside.
```

Save

```
You a▒e i▒ you▒ ▒e▒▒oom.
▒o ▒▒e ▒o▒▒▒ is a ki▒c▒e▒. ▒o ▒▒e wes▒ is a ▒a▒▒▒oom.

You ca▒ see: a ▒ai▒ o▒ g▒asses, a co▒▒ee mug
What now? get glasses then wear glasses

g▒asses is ▒u▒ i▒ i▒▒e▒▒o▒y
Wow!

You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.

You can see: a coffee mug
What now? save wore-glasses.s
Saved!
```

Load

```
You a▒e i▒ you▒ ▒e▒▒oom.
▒o ▒▒e ▒o▒▒▒ is a ki▒c▒e▒. ▒o ▒▒e wes▒ is a ▒a▒▒▒oom.

You ca▒ see: a ▒ai▒ o▒ g▒asses, a co▒▒ee mug
What now? load wore-glasses.s
You are in your bedroom.
To the north is a kitchen. To the west is a bathroom.

You can see: a coffee mug
What now?
```

Drinking coffee with mask on

```
What now? get coffee

coffee is put in inventory

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.
What now? pour coffee

Coffee Poured

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.
What now? wear mask
```

```
Mask worn

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.
What now? drink coffee

Remove your mask to drink

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.
What now? remove mask

Mask removed and placed in your inventory

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.
What now? drink coffee

You feel energized

You are in the kitchen.
To the south is your bedroom. To the west is a hallway.
What now?
```

Evaluation & Known Bugs

- The `save` and `load` commands bypass the parser and are handled directly in the main game loop. This is a good thing as it means that the `process` function doesn't need to worry about IO, but adds a bit of complication since commands are handled in more than one place and the game loop has some more layers. With more time, this could be restructured.
- With more time, the `save` and `load` functions would benefit from improved error handling. At the moment, issues with file IO generally result in the program crashing, which is a poor user experience.
- The program follows the specified structure, with extra files to support input parsing.
- Haskell language features are used throughout, including list comprehensions and `filter` instead of recursive definitions or a more procedural code style.
- The game runs correctly and the commands are sensible.
- The program comes with an accurate `cabal` file to allow for dependencies to be installed and loaded easily. The program is well-documented, with Haddock, a README, and a CHEAT file.
- The program fully meets the specification, including all easy, medium, and hard additional requirements.

Conclusion

The game meets the project specification fully, including implementation of all additional requirements up to Hard. It is well documented, so a player can get started quickly, and is written using idiomatic Haskell. With more time, the program could be improved through better error handling.