Team: I Function

Members:
- Dylan Woodbury, Jose Lopez, Hannah Browne

Project:
We are designing and implementing a text adventure game in PureScript, a functional language that compiles to JavaScript.
La Fin Du Monde

The world is going to end and you’re the only one who knows. You feel a strange urge to tell other people that the end is nigh. At the end of the day, you discover that a race of aliens have determined that you were the supreme ruler of Earth and have planned to destroy the planet except for you and everyone you’ve told. They assume you will be pleased with this and you are whisked away into the galaxy.

You wake one foggy white morning to the alluring aroma of fresh coffee and the surprising knowledge that today, the world is going to end. You’re not quite sure how you know this, but it’s a fact, written in your bones and coursing through your blood. Perhaps your DNA has been wound in such a way as to allow you this bold insight into the workings of the universe. Or perhaps you’ve gone a bit off your rocker. You yawn, and sit up.

--> look around

You look around the room that you share with your closest friend, Emily. Your eyes linger on every small detail, aware of the fact that this may be the last time you see it. It’s neat except for a small pile of clothes heaped onto the dresser and your dog, Max’s, leash thrown over a chair. There’s a note on your desk that you don’t remember seeing before.

--> read note

You grab the crumpled sheet of paper off the desk. There’s a message written in crayon that you can barely make out—Rbbvn gt it? Gt chict? Oh, Emily must have accidentally dropped this here, it just says, “remember: get chocolate.”

--> go to kitchen

You follow your nose to the kitchen, where a pot of coffee sits warmly on the counter, sparkling in the sunlight. At least you’ve found some beauty on this day of impending doom. Your housemate Cody sits at the counter with his face buried in a mountain of textbooks, humming a tune. Despite the early hour, he already has the scent of weed on him; it mingle s in the air with the smell of earthy Java. He belches, loudly. You feel uncomfortable. Your pug Max runs over to you, curled tail wagging in eager expectation of breakfast.

--> feed max
Changes from original proposal

- Haskell -> PureScript
  - PureScript is a Haskell-like language which compiles into JavaScript.
  - Easier to put on the web!
  - Differences:
    - Strictly evaluated
    - Eff Monad instead of IO
    - No special syntax for tuples
    - Composition operators (instead of . , <<<)
Current Progress

- Story close to completion
- HTML/CSS completed
- JavaScript/PureScript portions in progress
Story

- You wake one morning knowing that this is your last day on Earth
- You go about your day with the option of telling others that the world is going to end
Story

- It turns out:
  - The world is destroyed at the end of the day by a gigantic meteor
  - But you’ve been saved! By a legion of friendly aliens who believe you are the supreme being of Earth
Story

- Different endings can be achieved by telling different people about the end of the world
  - ex) if you tell your pug, Max, he will be abducted and saved with you
Program Features

- Users can build/modify worlds
- Easy to switch between playing/modifying
- JSON/txt saving/loading
- Response system
- Out of Reach: Using PureScript’s ability to interact with DOM to offer visualization, other interactions, animation, etc.
newtype Person = Person
  { name :: String
  , age :: Number
  }
newtype Thing = Thing
  { name :: String
  , edible :: Boolean
  }
newtype Room = Room
  { name :: String
  , description :: String
  , people :: [Person]
  , things :: [Thing]
  , rooms :: [Room]
  }
newtype GameState = GameState
  { room :: Room
  , player :: Person
  , rooms :: RoomMap
  , response :: Response
  }

personCody :: Person
personCody = {name: "Cody", age: 23,
relationship: "housemate"}

personEmily :: Person
personEmily = {name: "Emily", age: 22,
relationship: "roommate"}

personEllen :: Person
personEllen = {name: "Ellen", age: 28,
relationship: "boss"}
oNote :: Thing
oNote = {name: "string", edible: false}

New version of state is returned with each change
Game Loop Pseudo

GameLoop:
Get command
Get function(s) associated with command
   Execute if allowed, return new state
Perform checks/functions required after functions are executed
   Ex: after hitting someone, check to see if they are
       the type to get angry, sad, and execute their reaction
Perform “state checks” that look for important events
   If found, perform scripted changes to state and return
       These allow developer greater narrative agency, the
           ability to make events
Compile a response detailing significant plot advancements and
    changes to the state
Project Challenges

- Trying to find a way to use Haskell for web programming; eventually scrapped in favor of PureScript
- Downloading and working in PureScript
Work in Progress

- End game scenarios
- Finishing up the PureScript implementation
- Handling I/O in JavaScript