Team PythonScript
Tianrun Liu, Bo Hui Lu, Brian Lee, Tim Marder
SoftDev1 Pd6
2019-01-08

Final Project - “Mykolyk’s Ultimate Dictionary” derived from the acronym MUD (Multiplayer Dungeon), the inspiration for the game

Plot: A crazy kid found a magical book(the Necronomicon) in the QAF after failing horribly in a spelling bee competition. Out of anger, he used the power of the Necronomicon to convert all words in the dictionary into unspeakable horrible monsters so that no one can make fun of him for spelling something “wrong.” The chosen one(user), was the winner of the competition that the evil kid lost in, and wanted to restore peace to the world by defeating the evil wizard. Along the way, the hero restores words into an ancient artifact(“The Mykolyk Dictionary”) by beating a puzzle every time he encounters a monster. Our hero’s IQ grows every x Number of words that he restores, which makes it easier for him to defeat the nameless monsters.

COMPONENTS:

1. A dungeon crawling game displayed on a website
    a. Similar to Discworld MUD http://discworld.starturtle.net/lpc/ but with plot mentioned above
        i. Try [telnet discworld.starturtle.net 23] in the terminal to get an idea
    b. TUI - text-based user interface
        i. Use javascript
        ii. Compass
        iii. Dialog box: shows text and combat prompt
        iv. Command prompt: user types command as he goes (type “w” for going to the west, wasd, etc)
        v. (nice to have) Status bar: shows character status (hp, pw, ac, etc..)
    c. Map Generation
        i. Random Dungeon map is generated every time the user starts a new game or enters a new area https://medium.freecodecamp.org/how-to-make-your-own-procedural-dungeon-map-generator-using-the-random-walk-algorithm-e0085c8aa9a
    d. Level layout
        i. Tutorial level (pre-generated)
        ii. Dungeon level (random-generated)
        iii. (nice to have) Boss level (pre-generated) - might end up being a randomly generated monster with higher difficulty compared to others

e. In-depth description
    i. The user must move around a map and fight monsters with unpronounceable name. When the user encounters a monster, a combat prompt will pop out and ask the user what letters are in the monster's original name. If the user fails to complete the word after x tries, he will become mortally wounded and have to start the game again. (add-on) the user will be given a certain number of hints based off of how many words they collected. (nice to have)The user then can use items to help themselves (i.e. showing the definition of the word)
    ii. (other thought) they get a multiple choice question about the definition of the word and must solve it correctly to collect the word. The word's spelling might be scrambled for extra difficulty.

2. Multiplayer features
    a. (nice to have) Group Chat (https://scaledrone.github.io/javascript-chat-room-tutorial/)
    b. Profile page
        i. Keeps track of which words the user has collected so far
        ii. Shows the level of the user(tentative addition)
    c. Forum for users to talk about the game(tentative)
    d. ? Q/A page (good for notifying players about confusing parts or bugs) **May fit in our forum idea
3. Database
    a. Store user login info
    b. Stores the game state for each user
4. Web API
    a. Word API:
        i. Used to generate English words for the game
    b. (nice to have, but we probably won't go through all that hassle) d&d 5th edition API
5. References to other works belong to the nerd subculture
    a. Because it is text-based roguelike RPG, so prepare to get cultured.

DATABASE SCHEMA:

USERS

Name	**Key**	**Unique**	**Type**

user_id	Primary	Yes	Integer
username	none	Yes	String
password	none	No	String

GAME_STATE

Name	Key	Unique	Type
user_id	Primary	Yes	Integer
iq	none	No	Integer
level	none	No	Integer

WORDS

Name	Key	Unique	Type
user_id	Primary	No	Integer
word	none	No	Text

TASKS:

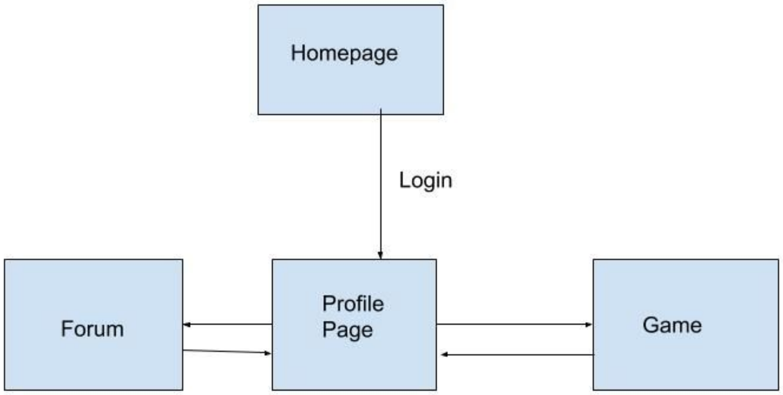
Tim: profile/maybe Q and A

Bo: database, API

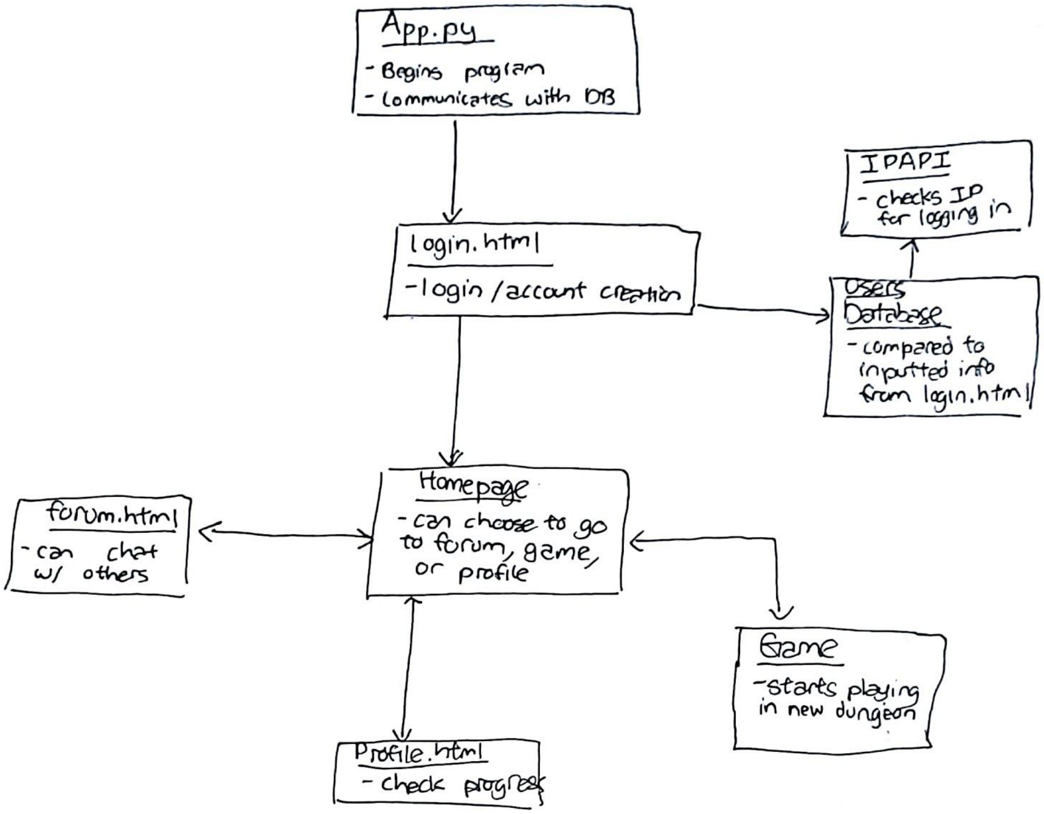
Tianrun: Dungeon/game movement

Brian: CSS, API

SITE MAP:



Component Map:


App.py
- Begins program
- Communicates with DB
IPAPI
- checks IP for logging in
login.html
-login /account creation
Users Database
-compared to inputted info from login.html
forum.html
- can chat w/ others
Homepage
- can choose to go to forum, game, or profile
Game
-starts playing in new dungeon
Profile.html
- check progress