

Team Advent: Tomas Acuna, Tami Takada, Oscar Wang, Renggeng Zheng

Word in the Middle - Team Advent (turn-based local game)

Front end framework: Foundation

- Forms are easier to format using the same grid classes
- Easier to understand/learn
- Similar elements as are in Bootstrap (containers, grids, etc)

Project Explanation

Two players (local on one computer) will be assigned one randomly generated “word in the middle” by the *Random Word API*. They will also individually be given two distinct, also randomly generated, starting words and a selection of related words that appear as links. Related words are any words that appear as synonyms for the current word given by the *Thesaurus API*, and as links in the current word’s Wikipedia entry given by the *Wikipedia API*.

The aim of the game is to get closer to the word in the middle by strategically choosing words that relate to the previous word. The winner of the game is the player that reaches the word in the middle first, determined by their total score at the end of the game, which is calculated by the total number of turns they took.

Tablified Version

Logistics of game progression
ONE PLAYER ONLY

Starting word (Turn 0): OMPHALISM
Target word (Turn 8): ZIRCONIUM

Turn #	Current Word
0	OMPHALISM
1	OMPHALOS HYPOTHESIS
2	STAR
3	HYDROGEN
4	CHEMICAL ELEMENT
5	CARBON
6	DIAMOND
7	CUBIC ZIRCONA
8	ZIRCONIUM

APIs Used

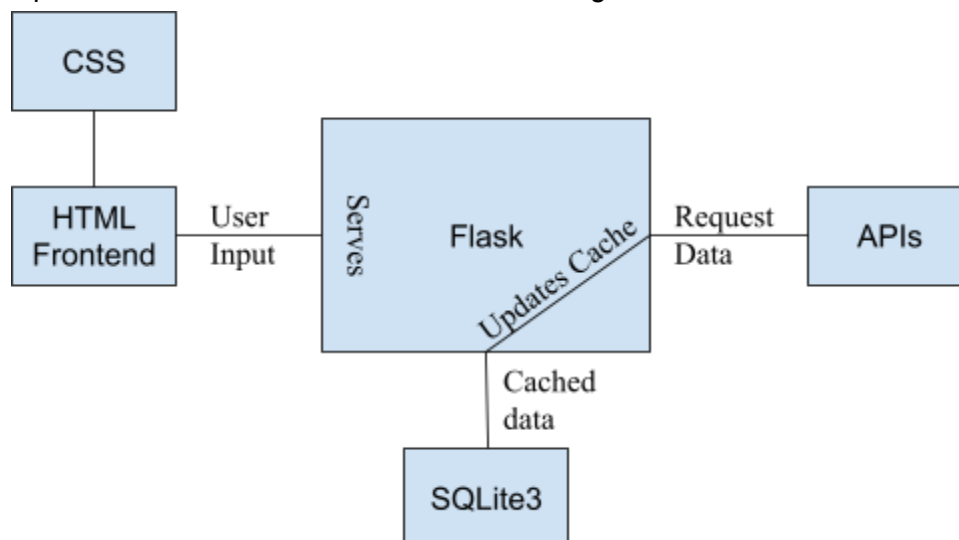
- <https://www.mediawiki.org/wiki/API:Links>
 - Wikipedia relatedness API
- <https://dictionaryapi.com/>
 - Dictionary + thesaurus API
- <https://random-words-api.vercel.app/word>
 - Random word api

Code Components

1. Flask
 - a. Hosts and serves the frontend of the website.
 - b. Renders templates with the randomly generated word, and links for each user's choices of related words for their current word.
 - c. Generate dynamic URLs to any word clicked by the user and repeat (b.) for the word clicked.
 - d. Making sure that users are not able to go back to the previous game page (with their previous word) using cache control or redirect
2. Sqlite3
 - a. Used as a database to cache results from API calls to maximize usage of our API call limits.
 - b. Used to store leaderboard data for any given pair of words in the game.
 - c. Stretch:
 - i. Store past matches against other registered players.
3. CSS/JS (Foundation)
 - a. Used for styling and interactive elements in our frontend.
 - b. Organize word choices in a neat way using the grid system

Component map

Things inside big boxes are what that big box does. Things inside smaller boxes/end nodes represent how that end node interacts with big box.



Program Components

- Word generation
 - There will be a random word in the middle generated by the Random Words API that serves as the goal for the two players
 - Each player is given a random starting word from the Random Words API
 - Choices for the player's next word (words related to their current word) will be provided by Thesaurus API and Wikipedia API
- Definition display
 - Using Merriam Webster API to show the definition of a given word if necessary
- Score tracking
 - The turn number and player (1 or 2) will be iterated and sent through Flask as a dynamic path to that Python can track the number of turns
 - The turn number will be also displayed at the top of the page and iterates each time a player clicks on a link to a related word
 - A player's score is tracked by total number of turns they take to get to the final word, **BUT**
 - If the user only has one option to click on in order to advance the game, their current turn is not counted to their score
- Leaderboard
 - Based on total points for each user
 - Will be tracked on a per game basis, with a game being any set of 2 randomly generated words (e.g. house vs mouse is a game)

Database Organization

Users

- Stores usernames and passwords for authentication

UserID (INTEGER PRIMARY KEY)	Username (TEXT NOT NULL)	Password (TEXT NOT NULL)
0	user	pass
1	user2	pass
...

Cache

- Stores the requested data of a certain word.
- game-by-game basis storage

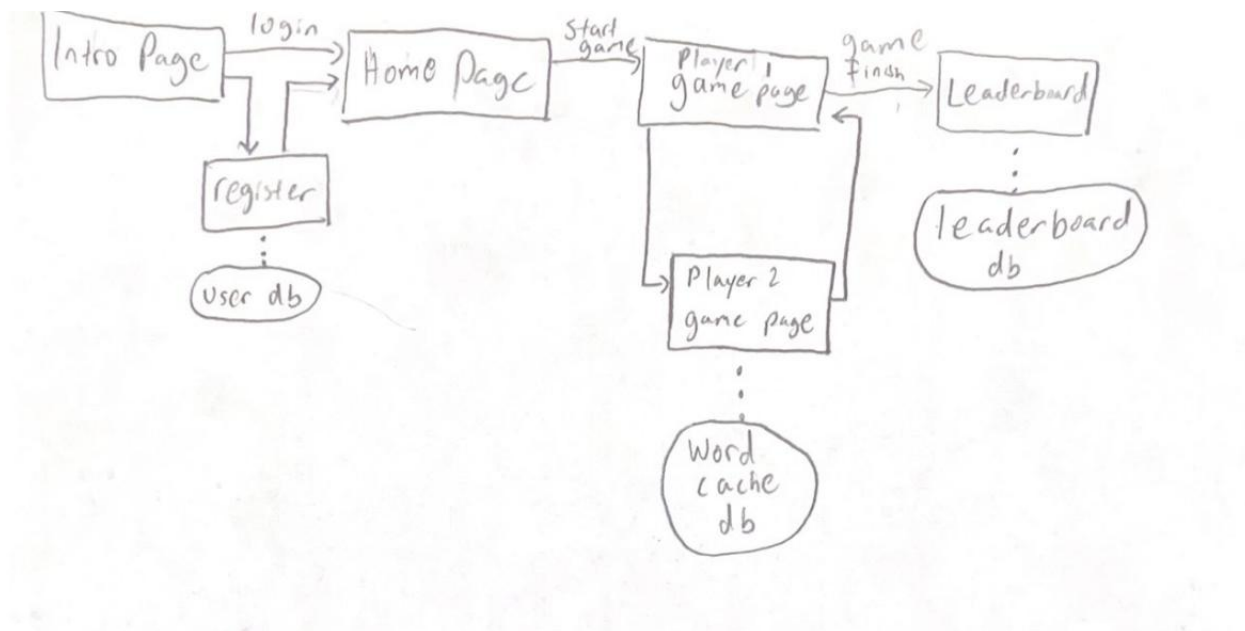
Item (TEXT PRIMARY KEY)	Date (TEXT NOT NULL)	Definition (TEXT NOT NULL)	Synonyms (TEXT NOT NULL)	WikipediaLinks (TEXT NOT NULL)
apple	<date of request>	A fruit with a red color.	JSON dump	JSON dump
banana	<date of request>	A berry with a yellow color and peel.	JSON dump	JSON dump
...

- Leaderboard
 - Leaderboards for a certain word combination
 - Should probably have a SHA-256 hash for them to organize

Hash (TEXT PRIMARY KEY)	Word1 (TEXT NOT NULL)	Word2 (TEXT NOT NULL)	Target_Word (TEXT NOT NULL)	Scores (TEXT NOT NULL)
213af...	apple	zoo		213af { user1: 1; user2: 2; ... }
asdf1...	banana	petroleum		asdf1{ user1: 4; user2: 3; ... }
...

Front end Site Map

- / - Home page
 - Includes links to the game page and leaderboard
 - Displays total points and rank on leaderboard
- /game/<player> - New game page
 - Displays the current word and the word in the middle along with 10 choices for the next word
 - Different words are shown depending on which player (player 0 or player 1) is having their turn
- /leaderboard - Leaderboard page
 - Displays rankings based on total points, depending on game



EXTRA (If time permits):

- Developing more game modes
 - Public figure mode (link two public figures given by NYT API)
 - Turn limit mode (get to target word in set number of turns)

Task Assignment

- Front end - Tami
- Database - Oscar
- API calls - Reng
- Word generation - Tami
- Leaderboard - Tomas

Target Ship Date - 12/20/21