Memory Game Plans

Responsibilities:

* Brainstorm ideas and decide on an application
* Establish a schedule to organize time and dates to work on the project
* Create prototype for the interface
* Create a general algorithm
* Create a pseudocode
* Write the code for the application on Netbeans
* Beta-testing

Brainstorming Ideas:

* Login and Sign up page to have sets linked to an account
* Main page that displays existing sets and gives users the option to create a new set of edit an already existing one
  + Upon clicking on an existing set, there is a pop up that gives the user the option to engage in flash card or matching game mode
    - Flashcards mode: there is a scrollbar on the left side that allows users to have a full view of all the cards in the set. To move between cards, there is a left and right arrow. Terms can be added to a “Study Later” folder to be later reviewed on a separate page.
    - Matching game: terms and definitions are scattered on a page, and users must match them to their rightful pair
* Create set page that allows users to add a card with a term and definition
  + The number of cards in the set is updated with every card that is added
  + Upon clicking confirm, it brings users to the main page where their newly created set will appear

Schedule:

* May 3, 2022: Decide on a project plan and begin by brainstorming ideas and jotting notes on what components will be necessary for the project
* May 5, 2022: Create a prototype for the GUI interface on google slides, start by creating a UML class diagram to help organize the relationship of classes
* May 6, 2022: Create a general algorithm and pseudocode for the program
* May 9, 2022: Set up Github and Netbeans before starting to program, create new branches, create a project boards for game plans, and using issues to organize labour division and define priorities for the project
* May 10, 2022: Start programming in Netbeans by creating the designs of the GUI from the prototype and learning how to commit changes to Github Desktop, navigating branches, merge and pull requests, etc.
* May 11, 2022: Continue creating GUIs for each page for the game (login, sign up, home, create sets, flashcards view, memory game)
* May 12, 2022: Program a working login and sign up page for multiple users (login users)
* May 16, 2022: Create classes for sets and cards and program home and create sets page to work
* May 17, 2022: work on flashcards page and tried correcting add cards method to not override cardlist in other sets
* May 18, 2022: got flashcards working, add cards to sets method correct(static was issue)
* May 19, 2022: programmed a study later page and researched dictionaries and multidimensional arrays
* May 20, 2022: Added comments, shortened code, combined edit sets and create sets class, deleted unsed methods
* May 25, 2022: Implement multi-dimensional arrays in edit sets
* May 26, 2022: work on adding a progress bar to flashcards page and to remove a card from a set in edit sets, added 2d arrays throughout all code
* June 6, 2022: work on the general algorithm, tried adding remove study later card
* June 7, 2022: finish general algorithm and begin working on pseudocode, working remove study later card, adding star to card
* June 8, 2022: star icon add to any card in study later section, removed bug of duplicate cards in studylater, work on pseudocode
* June 9, 2022: finish pseudocode
* June 10, 2022: finished matching game

Porblems and Fixes:

* Problem implementing dictionaries so resolved to two dimensional arrays

UML Class diagram:

[Memory Game UML Class Diagram](https://app.diagrams.net/#G1BfXAnIArWlkJjs3JWtzFvgq5o4rSDSfW)

Prototype:

[Memory Game GUI Prototype](https://docs.google.com/presentation/d/1zVYT_ZA0jGGfbbcebf1rcO-PqmqKsd-KN9tGlM8Rfo0/edit#slide=id.g128fabe4470_0_1)

General Algorithm:

[General Algorithm Memory Game](https://docs.google.com/document/d/1etDcx2zZbIf9rZCOJFPQOkbG8wgakbQwVed7ZEQQIzo/edit)

Pseudocode:

[Pseudocode Memory Game](https://docs.google.com/document/d/1h1HCHzDjuhROegli3sINon1l8_a6L8TfnGS9OkynbdQ/edit)

New Feature:

[Dictionary](https://www.educative.io/edpresso/how-to-create-a-dictionary-in-java)

[MultiDimensional Arrays](https://www.educba.com/2d-arrays-in-java/)

[Progress Bars](https://docs.oracle.com/javase/tutorial/uiswing/components/progress.html#bars)

Replaced our card class with two dimensional arrays inside of the sets class. Since 2D arrays are organized as a table with rows and columns, it made since to implement this feature as oppose to an Array List of cards with individual terms. We spend 2 classes researching the best ways to incorporate this into our code without having to greatly change anything else. Initially we used the standard String[][] 2D array but realised when passing data over to the StudyLater 2D array, there would be columns with void data. This caused a void error meaning our data couldn’t be passed along to the study later page. A solution was to use the ArrayList<ArrayList<String>> 2D array. The parameters didn’t set a row/column count which solved the issue of having empty slots of data. Having an ArrayList import also allowed us to remove rows from the list much easier.