Meeting 4: Scrum #3

Time: December 12, 5:00 PM to 5:30 PM
Place: Discord call

Description:

Roles:
- Tanush Nadimpalli: Developer **[present]**
- Rohan Shiva: Product Owner **[present]**
- Ramki Pitchala: Scrum Master **[present]**

Ramki

What did you do?
- Finished **[Task 1]**, **[Task 2]**, and **[Task 3]** and committed the changes to Github
- Changed `PermutationGenerator` to a more abstract `SequenceGenerator` which can take policies on how numbers can be selected in a sequence to generate more abstract sequences. Used `SequenceGenerator` to generate unique combinations for `FillInTheBlankProblemGenerator`.

What's not working?
- The `FillInTheBlankProblemGenerator` works well for languages like Python, JavaScript, Java, C++ and so forth but there are some small discrepancies.

What are you planning on working on?
- I plan to help my team record the demo video and fix any small discrepancies or add more tests to problem set generation.

Task Updates
- **[Task 1]** moved to DONE
- **[Task 2]** moved to DONE
- **[Task 3]** moved to DONE

Tanush

What did you do?
- Finished up **[Task 8]** where I modularized the ResultView component to separate the problems by their type and display them accordingly.
- I had to modify the ResultView component unit tests to address the new code changes.
- I also had to re-look at **[Task 5]** and add support for fill in the blank option to the recently added tests to the GenerationForm.

What's not working?

- I didn't have many issues regarding my tasks.

What are you planning on working on?
- I'm planning on working on the demo video, making sure everything is integrated and the electron application builds to windows and mac without any issues.

Task Updates
- **[Task 8]** moved to Done

Rohan

What did you do?
- Fixed issue with some tests failing: there was a problem with how some tests were sending mock props to the components.
- I finished implementing changes and tests to `GenerationForm` such that users can select the number of problems per problem type.
- Completed adding functionality to export generated problem-set to a JSON file.

What's not working?
- No major problems, but having some issues with git locally when fetching updates from remote repo.

What are you planning on working on?
- Finishing the exporting problem-set to a JSON file functionality tests

Task Updates
- **[Task 7]** moved to DONE
- **[Task 9]** moved to DONE
- **[Task 10]** moved to IN PROGRESS

Overall

Looking in great shape, we are on track to complete all the backlog tasks from the sprint backlog by the end of the sprint.

[Tanush]: Great job Ramki on the `FillInTheBlankProblemGenerator`!
[Ramki]: Great job everyone completing a majority of the tasks we had planned at the beginning of the sprint. I think now would be a good time to record a demo to share for our stakeholders.
- [Rohan]: +1
- [Tanush]: +1, I think we can use my computer to record the demo.
[Ramki]: Rohan, would we move **[Task 10]** to the next sprint?
- [Rohan]: I think that makes sense.

Updated Sprint Backlog

[Task 1] Create a `FillInTheBlankProblemGenerator` class
- Assignee: **Ramki**
- Status: DONE
- User Story: **[Epic 1]**

[Task 2] Write unit tests for `FillIntheBlankProblemGenerator`
- Assignee: **Ramki**
- Status: DONE
- User Story: **[Epic 1]**

[Task 3] Modify `ProblemSetGenerator` to use `FillInTheBlankProblemGenerator` to generate "fill in the blank" problems
- Assignee: **Ramki**
- Status: DONE
- User Story: **[Epic 1]**

[Task 4] Modify `GenerationForm` to allow instructors to select "fill in the blank" problem type
- Assignee: **Tanush**
- Status: DONE
- User Story:**[Epic 1]**

[Task 5] Modify `GenerationForm` tests to ensure "fill in the blank" problem type can be selected in GenerationForm
- Assignee: **Tanush**
- Status: DONE
- User Story:**[Epic 1]**

[Task 6] : Modify GenerationForm to allow the # of problems per problem type to be set by the user
- Assignee: **Rohan**
- Status: DONE
- User Story: **[Epic 3]**

[Task 7] Modify `GenerationForm` tests to ensure # of problems per problem type implementation is respected
- Assignee: **Rohan**
- Status: DONE
- User Story: **[Epic 3]**

[Task 8] Modularize `GenerationResult` component and display the "fill in the blank" problem type. Update tests to make sure they reflect the new changes.
- Assignee: **Tanush**
- Status: DONE
- User Story: **[Epic 3]**

[Task 9] Add functionality in `GenerationResult` to export a generated problem set as a JSON file.
- Assignee: **Rohan**
- Status: DONE

- User Story: **[Epic 4]**

[Task 10] Modify `GenerationResult` tests to ensure export button to export generated problem set as a JSON file functions properly.
- Assignee: Rohan
- Status: IN PROGRESS
- User Story: **[Epic 4]**