Section 1: About You

What project are you applying for?

Accessibility Improvements

Why are you interested in working with Oppia, and on your chosen project?

Oppia is really doing a great noble work by providing free quality education. Being a student and coming from a humble background I know the importance of education. I really liked the idea of an offline learning platform (Android version) as still there are so many places where people don't get proper internet connectivity. I really appreciate what oppia is doing to make quality education accessible to all and that makes me motivated to contribute towards Oppia. Apart from that, the community of Oppia is also great. I have been contributing to Oppia-Android for the last 7 months and I can say that I have learned a lot. Codebase of Oppia-android is also very well written. I want to learn how to write clean and concise code. Since I have joined a CLaM team, I have mostly solved issues which are related to accessibility only. As the goal of Oppia suggests making education accessible to all, I really felt that the android version of Oppia should be accessible to people with disabilities also. So, I have chosen this project as I am already familiar with adding accessibility support with testing and also .

Prior experience

Give some details of your prior experience that highlight your experience with regard to the technical skills needed for your chosen project. Describe specific achievements/things you did/problems you've solved in the past. Prioritize open source contributions, where possible.

I have been doing android development for almost more than one year. I have built a couple of android apps using java and firebase as backend. From this one of the major android app which I have built is Tokenizer. Through this app, users can book appointments at any place like hospital, saloons, etc. There was a separate application for business owners who can manage their appointments. I have also worked as an Android Trainer for 1 month where I have been creating video tutorials for Android Development using Kotlin.

Top 5 PRs Merged in Oppia-Android:

1. Use OppiaTestRule in all Espresso Test #4133 and #4179
2. Merge Accessibility Test rule in OppiaTestRule #4025
3. Add label for HomeActivity #3850
4. Merge Completed activity into single File #3689

5. Merge Concept Card Fragment into single xml file. #3443

Project size

Medium

Project timeframe

I will complete the project during the default GSoC coding period only i.e June 13 - September 12.

Contact info and timezone(s)

Email: vrajdesai78@gmail.com

Contact No. 9099735209/8849662799

Github: https://github.com/in/vrajdesai78

Linkedin: https://linkedin.com/in/vrajdesai78

Timezone: Indian Standard Time (IST +5:30 GMT)

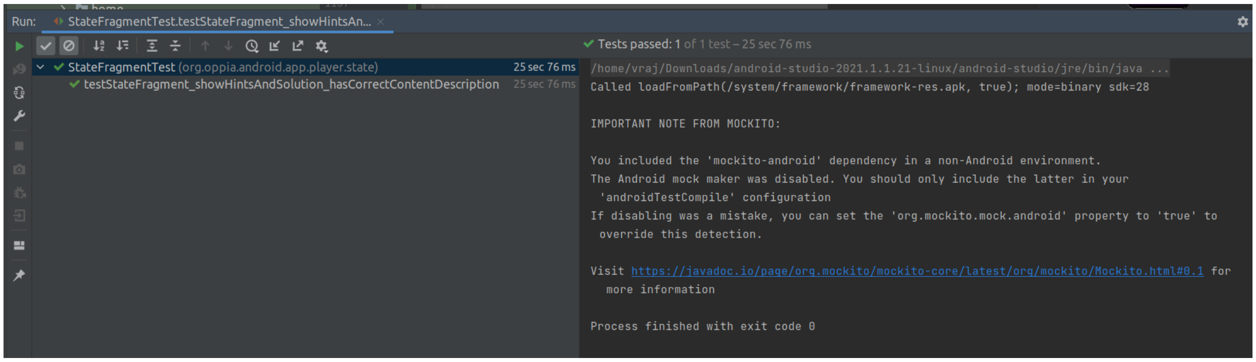
Time commitment

I will work for at least 4-5 hours per day throughout the coding period and also commit more time if it's required.

Essential Prerequisites

Answer the following questions (for Oppia Android GSoC contributors):

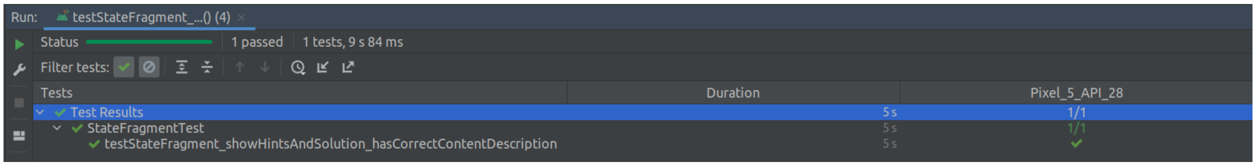
- I am able to run a single Robolectric test target on my machine via Android Studio.



```
✓ StateFragmentTest (org.oppia.android.app.player.state)                      25 sec 76 ms
    ✓ testStateFragment_showHintsAndSolution_hasCorrectContentDescription     25 sec 76 ms
```

```
✓ Tests passed: 1 of 1 test – 25 sec 76 ms

/home/vraj/Downloads/android-studio-2021.1.1.21-linux/android-studio/jre/bin/java ...
Called loadFromPath(/system/framework/framework-res.apk, true); mode=binary sdk=28

IMPORTANT NOTE FROM MOCKITO:

You included the 'mockito-android' dependency in a non-Android environment.
The Android mock maker was disabled. You should only include the latter in your
 'androidTestCompile' configuration
If disabling was a mistake, you can set the 'org.mockito.mock.android' property to 'true' to
 override this detection.

Visit https://javadoc.io/page/org.mockito/mockito-core/latest/org/mockito/Mockito.html#0.1 for
  more information

Process finished with exit code 0
```

- I am able to run a single Espresso emulator test target on my machine via Android Studio. (Show a screenshot of a successful test.)



Status 1 passed 1 tests, 9 s 84 ms

Tests	Duration	Pixel_5_API_28
✓ Test Results	5 s	1/1
✓ StateFragmentTest	5 s	1/1
✓ testStateFragment_showHintsAndSolution_hasCorrectContentDescription	5 s	✓

Other summer obligations

Mostly my end term examination should be completed before the start of the coding period in June. So, I have no other commitments and I am fully available to commit my time for GSoC.

Communication channels

I would be comfortable with all communication channels like gmail, hangouts, Google meet, zoom, etc. whichever suits my mentor. I would generally communicate with my mentor for 2-3 days per week according to the requirement and availability of a mentor.

Section 2: Proposal Details

Problem Statement

Link to PRD (or N/A if there isn't one)	N/A
Target Audience	Learners with vision-related impairment, learners
Core User Need	As a learner with a vision-related impairment, I want the same user experience that an uninhibited learner has. As a learner, I want the App to work properly with all types of settings like large text, bed-time mode, night-light mode, magnification mode, etc. As a learner I want my data to remain persistent when I rotate my device. So, for example, if I have entered some text in a textbox then it should remain there as it is even if I rotate my device. This helps to ensure that I don't lose context when switching device orientations.
What goals do we want the solution to achieve?	Improved accessibility support such that both users with vision related impairment and uninhibited users have the same user experience. Application is fully compatible with all types of accessibility settings. Improved user experience such that the data remains persistent if the user rotates their screen.

Section 2.1: WHAT

This section enumerates the requirements that the technical solution outlined in "Section 2: HOW" must satisfy.

Key User Stories and Tasks

#	Title	User Story Description (role, goal, motivation) *"As a ..., I need ..., so that ...."*	Priority[1]	List of tasks needed to achieve the goal (this is the "User Journey")	Links to mocks / prototypes, and/or PRD sections that spec out additional requirements.
1	Add meaningful labels to all components	As a learner with vision related impairment, I want to know which	Must Have	Add proper content description for all components.	N/A
				Make sure that all screens	N/A

[1] Use the **M**o**SC**ow system ("Must have", "Should have", "Could have"). You can read more here.

		component I am interacting with.		have the same labels and content description.	
2	Improve the flow for screen - readers	As a learner with vision related impairment, I want to access the components in the same flow as normal users are accessing.	Must Have	Make the flow of the screen readers consistent for RTL users . For example, uninhibited users for right to left languages (Arabic) read a screen from top-right so for RTL users with vision related impairment also screen readers should start from top-right only.	N/A
3	Make app fully accessible with all settings	As a learner, I want that App is compatible with all settings.	Must Have	Make the app compatible with different accessibility services and settings like Large text, bed-time mode, night-light mode, talkback, etc. and should pass Accessibility Scanner checks.	N/A
4	Persist Data on screen rotation	As a learner, I want that when I rotate my device my entered data or selected options should remain the same.	Must Have	Keep the data persistent on screen rotation. For example if a user has typed some answer then it should remain the same or as it is after rotating the screen.	N/A

Technical Requirements

Data Handling and Privacy

There is no requirement to store data which leads to privacy issues.

Section 2.2: HOW

Existing Status Quo

Following is an accessibility audit of all different flows of the app which shows the difference between the experience for uninhibited users and users with vision related impairment. Following audit is based on our current develop branch (not MR4).

Note: Docs doesn't support long GIFs so I have uploaded a GDrive link of screen recording.

Description of Flow	Experience for uninhibited users	Experience for users with vision related impairment
Startup Screen to Home screen and navigating different topics.	Video Link Here, once we click admin we will see the home screen where we can navigate through different topics.	Video Link Here, once we click on Admin, we can see the home screen. The user's experience is the same as uninhibited users.
Navigating First Test Topic	Video Link Here, we navigate to various tabs inside the First Test topic.	Video Link Here, when we navigate to various tabs we haven't found any problem with the flow of talkback reader and user experience is the same as uninhibited users.
Navigating to recommended stories.	Video Link We are navigating to recommended stories and lessons flow tab.	Video Link When navigating to the lessons tab the flow is not good, and the content description is very repetitive which needs to be fixed.
Navigation to navigation menu	Video Link Navigating to different activities like profile, options, help and administrator controls.	Video Link While navigating through different activities using Talkback reader, the user experience is almost same with the uninhibited user's and there is no issue with the talkback reader.
Navigating prototype exploration of First Test topic	Video Link - We are navigating to an exploration player where we have to enter an answer in the textbox.	Video Link - When we go back while writing answer the hint of editbox is disappeared which does not create a good user experience as the content description for that editbox is changed to just 'editbox'.
Navigating to further questions in exploration of First test topic	Video Link - We are answering questions where we have to select a radio button.	Video Link - When we submit our answer by selecting a radio button, the flow of talkback is shifted to the top i.e. to navigate up button. This won't create a good user experience for users with vision related impairment.

Navigating to remaining questions in prototype exploration of First test topic.	Video Link - We are answering questions where we have to select answers by drag and drop.	Video Link - When we successfully answer all questions there is no feedback for users with vision related impairment that all questions are completed, it just displays a return to the topic button.
Navigating to Image Region Selection Exploration.	Video Link - Exploring Image Region Selection where uninhibited users can select regions from the image.	Video Link - When we select any planet from the image it won't show any feedback to users with vision related impairment.
Navigating to Fractions chapter 1.	Video Link - Answering questions based on fractions. Here, uninhibited users can understand the question by seeing Images.	Video Link - For certain cases there are various images which don't have a content description. So, when a user with vision related impairment uses talkback to read the question he/she doesn't know what that image is due to not having a description.
Navigating to the remaining part of Fractions chapter 1.	Video Link - Answering remaining questions of Fractions chapter 1. Uninhibited users can see options and select an appropriate image as the answer.	Video Link - For some questions, the user has to select an image as an answer since but that image doesn't have a content description, it is impossible to effectively answer such questions.
Navigating to chapter 2 of Fractions.	Video Link - Answering questions of Fractions chapter 2. Uninhibited users can understand questions by seeing images.	Video Link - Here also, in almost every question there are images which don't have content description which doesn't create a good experience for users with vision related impairment.

Results of accessibility scanner with all system settings.

Accessibility Setting	Test Result
Large Font Size: Increased the font size from system settings to large.	Tested all activities of the app with Accessibility Scanner but didn't find any specific issue with large font size.
Small Font Size: Decreased the font size from system settings to small.	Tested all activities of the app with accessibility scanner but here also I haven't

	found any specific issue with small font size.
Bedtime Mode: Enabled bed time mode from system settings which changed the phone’s colourful screen with black and white.	I have tested all app screens in bedtime mode and I haven’t found any issues with color contrast.
Night Mode: Enabled night light from system settings which dimmed the screen brightness.	Here also I have tested all screens of the app in night mode and haven’t found any issue with color contrast.
Talkback Reader: Tested using accessibility scanner. Also, tested it manually by enabling talkback reader and wrote all results in answer number 2.	I have tested all screens of the app with Google's accessibility scanner. There are few issues with it as follow: ● No content description for Images which are included in fractions topic. ● Blank space in Resume lesson activity when there is no chapter description. ● Proper content description should be there for the ImageRegionSelectionInteraction view.
Dark Mode: Enabled dark mode from system settings	Actually, I didn’t know exactly how many activities support dark mode currently. But whatever activities I have observed which have implemented dark mode fully don't have any issue with the accessibility scanner. Checked activities include: administrator control activity, select profile activity and Add profile activity. Issues Found (Where dark mode not implemented completely): ● When we click on Topic then topic toolbar title, back button, topic_description_text_view doesn’t have proper color contrast. ● Also, topic_play_text_view doesn’t have proper color contrast.

Note: I have added a same video link for some sections to show accessibility scanner results in which I have tested whole application (MR4 v0.7)

Full accessibility audit based on accessibility testing tracker
Green - Tests requires no work
Yellow - Tests requires work that will be addressed by this project

Gray - Tests requires work but will not be addressed in this project

Test Description	Result
Basic Tests	
1.1. Do all images, edittexts, have meaningful labels, and pass on accessibility scanner?	Images that are present in questions and options don’t have a content description and also edittexts don’t have a content description once the hint goes away. Video Link editbox.gif fractionsOp.gif
1.2. Are all ephemeral/temporary things announced correctly?	Yes, correct answers and congratulations are announced correctly via screen reader but for that user needs to click on the screen after submitting the answer.
1.3. All flows can be completed through touch discovery and swipe navigation and all permanent items within those flows are completely reachable?	Yes, all items are reachable but in the lessons tab, disabled chapters are not clickable when talkback is on but still the user will know that there are disabled chapters in the list as the talkback will say the number of chapters in the list. The reason we have kept disabled chapters as non clickable is because by default talkback doesn’t work when elements are disabled (to ensure consistency between screenreader & non screenreader users).
1.4. Are all disruptive announcements (announceForAccessibility) limited to important cases only?	Yes, all disruptive announcements like correct answers and congratulations are limited to important cases and when no other text with it is announced.
1.5. Do all interactions inside the app have proper accessibility action to control the UI?	Yes, all interactions have proper accessibility action to control the UI.
1.6. Are the purpose of each link, control, button mentioned via correct labels?	All buttons have correct labels and accept links. Accessibility scanner suggests that link text should have more description (Video Link screenshot).
1.7. Is the user's context unchanged after any step?	Yes, the user's context is unchanged after any step like submitting an answer in the exploration player.
1.8. Are there proper alternatives to all actionable or informative things which time out? (Animations generally don't fit into this category)	Yes, we have alternatives for things which get time out.

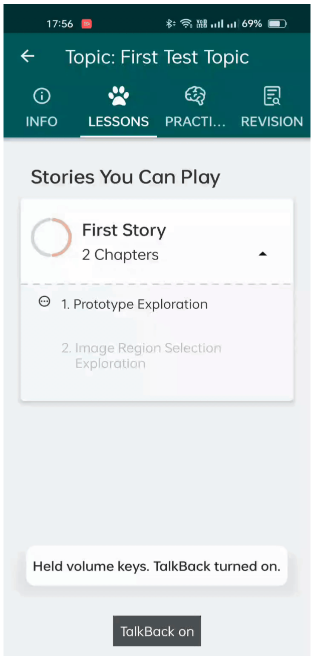
1.9. Are all errors displayed in text with correct mitigations & context to fix the error?	Yes, all errors are showing correctly.
1.10. Are views/content which automatically flashes, updates and moves are stoppable and safe for people with attention deficit disorder?	Yes, colors are smooth enough that it won't harm people with attention deficit disorder. Regarding confetti announcements, the accessibility scanner suggests increasing the text's foreground to background contrast ratio (screenshot).
1.11. Does the app pass all contrast ratio, minimum size, touch target size tests in App Scanner for all items and embedded images?	For some text in the exploration player, the accessibility scanner suggests increasing the text foreground to background ratio (screenshot).
1.12. Does the app pass all contrast ratio tests for content images and thumbnails on Accessibility Scanner?	The accessibility scanner suggests improvements in for contrast ratio in confetti text and some specific screens which have content images as shown in this video
1.13. Does every audio/video have transcripts?	Yes, app supports audio in all languages i.e. English, Hindi, Arabic, Brazilian, Portuguese, Swahili, and Hinglish.
1.14. Does the app pass accessibility scanner tests in dark mode?	No, in dark mode still we need to work as currently only a few screens are supported in dark mode.
1.15. Are all custom and non-standard accessibility features documented in the Help menu of the app?	Yes. We don't have any features which needs special documentation in Help menu.
1.16. Does the application work correctly in portrait and landscape modes?	Currently, our application supports both portrait and landscape mode but when the user rotates the screen, the user's entered or selected answer is not retained.
1.17. Are all touch targets large enough to work without accurate precision?	Yes, all touch targets are accurate and easily accessible according to the accessibility scanner.
1.18. Is all the information communicated with more than one sensory characteristic?	Yes, all information is communicated with more than one sensory characteristic like audio and text.
Advanced Tests	
2.1. Is UX consistent across all screens when the screen has similar UI components?	Yes, UX is consistent across all screens except ImageSelectionRegion interaction.

2.2. Do all important actions have undo functionality?	Yes, all important actions have undo functionality.
2.3. Can the audio player be paused, stopped along with hardware volume controls?	Yes, the audio player will be paused, stopped along with hardware volume controls.
2.4. Are all labels clear and describe the purpose of the page correctly?	Yes, all labels are clear except the lessons tab in which we have repetitive content description (Sr.gif).
2.5. Is the app accessible for magnified screen, large text, bed-time mode, night-light mode, without visible screen(with Talkback) and with different accessibility services?	Yes, the app is fully accessible for magnified screen, large fonts, small fonts, bed-time mode, night-light mode, talkback reader, and small fonts.
2.6. Data persists when transitioning from portrait to landscape and vice-versa.	No, data doesn't persist when we rotate for all interactions like SelectionInput, TextInput, NumericInput, FractionInput, DragAndDrop, and ImageRegionSelection (rotate.gif).
Content Tests	
3.1. Do all images, edittexts, have meaningful labels and pass on accessibility scanner?	Images which are present in questions and options don’t have content descriptions and also edittexts don’t have content descriptions once the hint goes away (video Link editbox.gif).
3.2. Are the purposes of each link, control, and button mentioned via correct labels?	All buttons have correct labels except links. The accessibility scanner suggests that link text should have more description (see video and screenshot).
3.3. Does the app pass all contrast ratio, minimum size, touch target size tests in App Scanner for all items and embedded images?	For some text in the exploration player, the accessibility scanner suggests increasing text foreground to background ratio.
3.4. Does the app pass all contrast ratio tests for content images and thumbnails?	The accessibility scanner suggests improvements for contrast ratios in confetti text and some specific screens which have content images as shown in this video.

Solution Overview

We need to improve the flow of talkback screen readers as well as need to add meaningful non-repetitive content description for cases such as below screen (Topic Lessons Tab) . We can see that there is no proper content description and also it’s repetitive. The flow of talkback is also not efficient. We need to make the flow of talkback similar to normal users.

Sr.gif

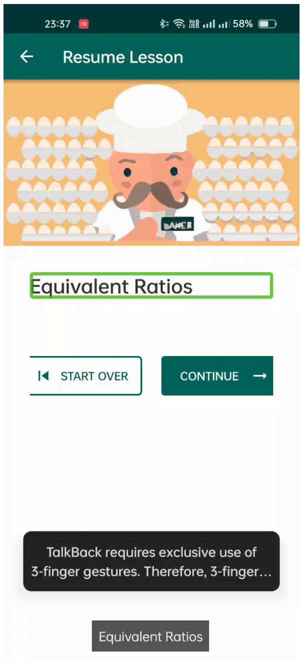


Held volume keys. TalkBack turned on.

TalkBack on

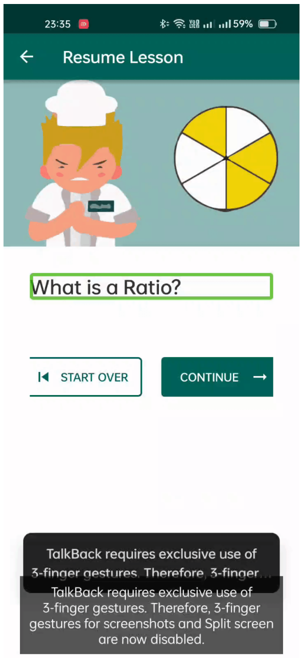
We can remove extra blank space from Resume Lesson Activity of Ratio and Proportions as it won’t create a good experience in talkback reader as shown below. It happened because Ratio and Proportions did not have a description for resume lesson activity.

space1.gif space2.gif



TalkBack requires exclusive use of 3-finger gestures. Therefore, 3-finger…

Equivalent Ratios

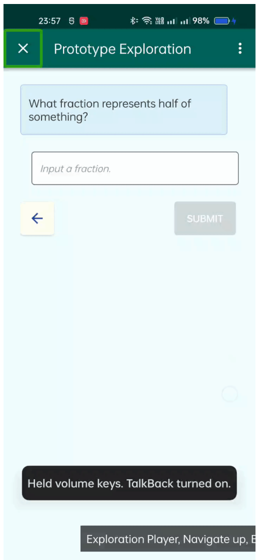


TalkBack requires exclusive use of 3-finger gestures. Therefore, 3-finger

TalkBack requires exclusive use of 3-finger gestures. Therefore, 3-finger gestures for screenshots and Split screen are now disabled.

In every custom InputInteractionView there is no proper content description. By default, when we use talkback reader it will read out a hint of the InputInteractionView but if we go back then that hint will be removed which won't create a good user experience for users with vision related impairment as shown below.

editbox.gif



What fraction represents half of something?

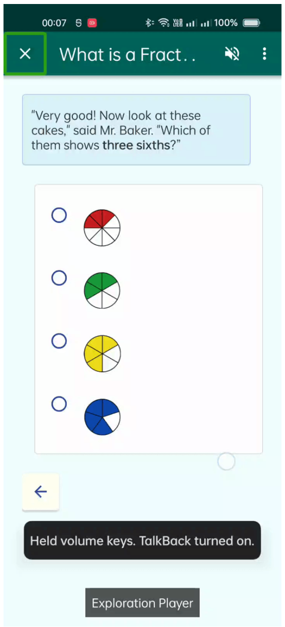
Input a fraction.

Held volume keys. TalkBack turned on.

Exploration Player, Navigate up, E

For the below question in Fractions Exploration player there is no proper content description for image options. As all options are of images, it should have meaningful content descriptions which help users with vision related impairments to understand them properly.

fractionsOp.gif



"Very good! Now look at these cakes," said Mr. Baker. "Which of them shows **three sixths**?"

Held volume keys. TalkBack turned on.

Exploration Player

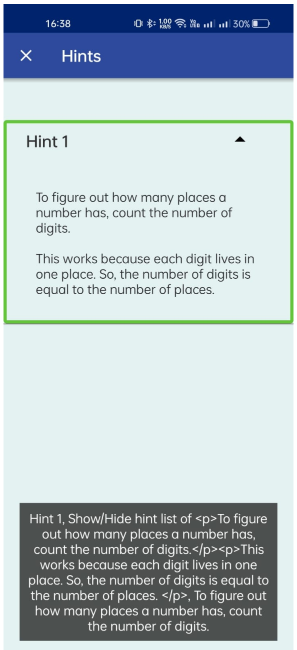
When talkback is turned on and we click on a link to a locked chapter item it won't navigate us to the top of the page (i.e. unlocked chapter). To solve this issue we need to send an accessibility event to the first index of recyclerview which is responsible to show all chapters (#3892--Video Link).

On the pin verification screen when the user enters an incorrect pin it won't speak out that it's incorrect when talkback is turned on (#2627). To solve this issue we will remove the error text textview. We will directly set an error message to our edittext.

Unable to exit overflow menu in exploration player as well as in revision card (#3894--Video Link). When the user is in the overflow menu then he/she should have the option to exit from it when talkback is turned on. To solve this issue we will replace the overflow menu with a bottomsheet which improves the user experience when talkback is turned on since it provides the user with an explicit exit option.

A label needs to be added for RecentlyPlayedActivity so that when talkback is on and we open RecentlyPlayedActivity, it should speak out as *"stories for you"* or *"recently played stories"* (#2824).

Talkback reads html tags for hints. So, when we go to the hint dialog then talkback should read hints as text only, and not read html tags (#3906). To solve this issue we have to give proper text for content descriptions of hints by parsing the html properly.



To figure out how many places a number has, count the number of digits.

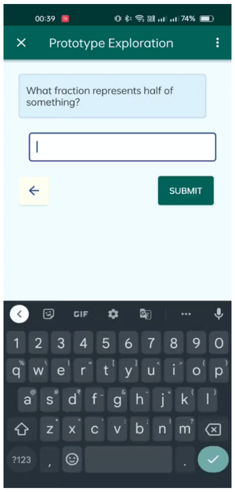
This works because each digit lives in one place. So, the number of digits is equal to the number of places.

Hint 1, Show/Hide hint list of <p>To figure out how many places a number has, count the number of digits.</p><p>This works because each digit lives in one place. So, the number of digits is equal to the number of places. </p>, To figure out how many places a number has, count the number of digits.

ImageRegions of ImageRegionSelectionInteraction don't have accessibility support. Currently, when we open ImageRegionSelectionInteraction and enable talkback then the selected image region doesn't have a content description (#3712).

Currently, we don’t have proper text for hyperlinks as accessibility scanner indicates that links need more descriptive description. We need to provide proper text for hyperlink such that it will pass for Accessibility Scanner checks (#3895).

In the below gif we can see that when we rotate the screen our entered answer is lost and this is the case for almost all screens in the exploration and question players. So, we need to retain this data when the screen is rotated. To handle this we can use the onSavedInstance method and put the RawUserAnswer (proto) in a bundle and retrieve it from the bundle using onCreateView. rotate.gif



What fraction represents half of something?

Third-Party Libraries

No new third-party libraries are required.

“Service” Dependencies

No new service dependencies are required.

Impact on Other Oppia Teams

There will not be any major impact on other Oppia Teams.

Key High-Level and Architectural Decisions

Decision 1: How to handle configuration changes for Exploration and Question Player

We have considered the following alternatives:

1. Use a singleton class in the domain layer which stores userAnswer.
2. Use onSavedInstance to handle configuration changes.

3. Use ViewModel and LiveData to handle configuration changes.

Among these, we believe that using onSavedInstance handle configuration changes is the best approach, because:

- We don’t need to create any additional class like we are creating a singleton class for singleton class approach.
- We don’t need to handle the issue of memory leak as onSavedInstance is life-cycle aware.
- It is easier to implement onSavedInstance as we can directly store UserAnswer (proto) in our bundle.

The above three approaches are contrasted in detail in the following table:

onSavedInstance approach	Singleton class approach	ViewModel approach
Easier to implement and less code complexity	Complexity of code is increased as we have to write additional singleton class	It is difficult to handle interactions when we have more than two interactions on one screen as all interactions are tightly coupled.
Don’t need to handle the issue of memory leak as onSavedInstance is life-cycle aware.	Need to handle the issue of memory leak separately as it is not life-cycle aware.	Don’t need to handle the issue of memory leak as the ViewModel approach is life-cycle aware.

Risks and mitigations

The main potential risk is that some accessibility issues may not be discovered until completing the project. To mitigate this risk of discovering new accessibility issues we will audit the full app before starting the project and also after completing the project such that if any more issues come after completing the project then it will be resolved.

Implementation Approach

Issues related to configuration change

Platform Parameter for configuration change

As we need to implement a onSavedInstance approach to persist data on configuration change we need at least 4-5 PRs to make it work for all interactions. So, while doing this we need to put our changes behind the feature flag such that if that feature flag is enabled then only our changes will be reflected using a platform parameter.

We will create a platform parameter *EnableConfigurationChangeOnSavedInstance* with a default constant value as false (off). Then we will use this platform parameter in StateFragmentPresenter when we handle onViewRestored and onDestroyView. If the default value of EnableConfigurationChange flag is changed to true

1. Create a platform parameter EnableConfigurationChange in utility\src\main\java\org\oppia\util\platformparameters\PlatformParameterConstants.kt with constant value as false.

2. Provide the platform parameter in domain\src\main\java\org\oppia\android\domain\platformparameter\PlatformParameterModule.kt and return constant which we created in above step.

3. Use the platform parameter in app\src\main\java\org\player\state\StateFragmentPresenter.kt by injecting it in the constructor. We can use this boolean value as a flag to enable and disable configuration change inside handleOnViewStateRestored and handleOnSavedInstance.

4. Use the platform parameter in app\src\main\java\org\player\state\QuestionPlayerFragmentPresenter.kt by injecting it in the constructor. We can use this boolean value as a flag to enable and disable configuration change inside handleOnViewStateRestored and handleOnSavedInstance.

5. Add support for testing in testing/src/main/java/org/oppia/android/testing/platformparameter/TestPlatformParameterModule.kt such that we can force to enable our flag for configuration change for testing.

Handle configuration change for Interactions using onSavedInstance approach

- RawAnswerInteraction in exploration.proto

```
message RawUserAnswer {
  oneof answer_input_type {
    bool continue = 1; // Just indicates the presence of the type.
    string text = 2;
    string fraction = 3;
    string numeric = 4;
    string number_with_units = 5;
    string ratio_input = 6;
    string numeric_expression = 7;
    string algebraic_expression = 8;
    string math_equation = 9;
    int32 multiple_choice_selection_index = 10;
    ItemSelectionRawAnswer item_selection = 11;
    ListOfSetsOfTranslatableHtmlContentIds drag_and_drop = 12;
    ClickOnImage image_region_selection = 13;
  }
```

- ItemSelectionRawAnswer in exploration proto

  ```
  message ItemSelectionRawAnswer {
    repeated int32 selected_indexes = 1;
  }
  ```

  InteractionAnswerHandler:
  - setRawUserAnswer -> Takes RawUserAnswer as a parameter and which is further override by every itemviewmodel which is related to user input like DragAndDropInteractionViewModel, FractionInteractionViewModel, ImageRegionSelectionViewModel, NumericInputViewModel, SelectionInteractionViewModel and TextInputViewModel.
  - getRawUserAnswer:
    - Returns RawUserAnswer (proto) and needs to be overridden by each interaction's viewmodel (similar to getPendingAnswer) such that we get a plain answer even if the answer is in error state.
    - If pendingAnswerError is null then we assign the answer to RawUserAnswer (proto) as we are doing it in the getUserAnswer method. If pendingAnswerError is null that means our answer is in error state then we just assign plainAnswer in RawUserAnswer, we won't assign its interaction form.
    - In every interaction view model like TextInputInteractionViewModel and others we can create a new variable i.e. pendingAnswerText such that this will store raw answer without removing any trailing and leading spaces. We can use this pendingAnswerText to assign it to a plainAnswer.
- StateFragment and QuestionPlayerFragment:
  - Override onSaveInstanceState() -> Get RawUserAnswer (proto) by calling handleOnSavedInstance() and put that proto in a bundle by calling bundle.putProto() method.

    - Create a new variable for RawUserAnswer (proto) in onCreateView and assign pendingAnswer to it by getting value from bundle using bundle.getProto() method.
- StateFragmentPresenter and QuestionPlayerFragmentPresenter:
    - handleOnSavedInstance() -> Returns RawUserAnswer (proto) by calling viewModel.getPendingAnswer() method.
    - Added a new parameter RawUserAnswer in handleCreateView which we use to pass in viewModel.setPendingAnswer() method.
- StateViewModel and QuestionPlayerViewModel:
    - setRawUserAnswer -> It takes RawUserAnswer as a parameter which is used to set Answer in viewmodels.
    - getRawUserAnswerWithoutError -> Returns RawUserAnswer (proto) from InteractionViewHandler which is further called in setRawUserAnswer.
- We have to assign values which we get from RawUserAnswer (proto) to each interaction such that it will reflect changes in UI. Following are the interactions in which we need to assign answer which we get from RawUserAnswer (proto):
    - FractionInteraction, NumericInputInteraction, TextInputInteraction, RatioExpressionInputInteraction, and MathExpressionInputInteraction:
        - Need to assign the string from proto
    - ImageRegionSelectionInteraction
        - Need to assign selected image region from proto
    - DragAndDropInteraction
        - Need to assign selected order and merged options from proto.
    - SelectionInteraction
        - Need to assign selected options from proto.

Other accessibility issues

Improve flow of Lessons Tab #3896

- Currently, when we navigate to the lesson flow tab, the title of the story is repeated twice, first when the whole recyclerview is selected and second when we select the title separately to open the story summary. Now, to remove this repetition of title in talkback reader, instead of giving content description to the whole recyclerview we can make each component accessible separately such that talkback will navigate to progress and chapter title separately.
- To make each component accessible we disable importantForAccessibility for story_container in topic_lessons_story_summary.xml and set content description for story_progress_container as *storyProgressPercentText + completed* (we can also declare separate string in strings.xml for content description of story_progress_container).
- We can also remove repetition from the recyclerview which shows a list of chapters by disabling importantForAccessibility for chapter_name in lessons_chapter_view.xml.

Remove the black space from Ratio and Proportion resume lesson activity.

- Need to handle edge cases for resume_lesson_chapter_description_text_view for Ratio and Proportion for when there is no summary description for the chapter.
- We can give a conditional statement for the importantForAccessibility attribute based on whether the chapter summary is an empty string or not.

Locked chapter item selection change is incorrect #3892

- To change the focus of the talkback reader to the unlocked chapter, we use sendAccessibilityEvent and change the focus of talkback reader to unlocked chapter i.e. first item in recyclerview.

Add meaningful content descriptions to every custom input interaction view.

- A meaningful content description needs to be added to FractionInputInteractionView, NumericInputInteractionView, RatioInputInteraction, MathExpressionInteraction and TextInputInteractionView.
- We added default hint text of all interactions as its content description like for NumericInteraction we can set its content description as numeric_input_hint (from strings.xml)
    - FractionInputInteraction:
        - *Enter a fraction in the form of x/x, or a mixed number in the form x x/x.* (fractions_default_hint_text)
        - *Enter a fraction in the form x/x.* (fractions_default_hint_text_no_integer).
    - NumericInputInteraction:
        - *Enter a number.* (numeric_input_hint)
        - *Write numbers with units here.* (number_with_units_input_hint_text)
        - *Type an expression here, using only numbers* (numeric_expression_default_hint_text)
    - RatioInputInteraction:
        - *Enter a ratio in the form x:y* (ratio_default_hint_text)
    - MathExpressionInteraction:
        - *Type an equation here* (math_equation_default_hint_text)
        - *Type an expression here* (algebraic_expression_default_hint_text)
    - TextInputInteraction:
        - *Enter text* (text_input_default_content_description)
- For interactions like FractionInputInteraction we have multiple hint texts one is default hint text and another is hint_text_no_integer so while giving content description we need to handle these cases based on conditions which we are using currently while assigning hint text.
- We need to give hint text inside the init block of every InputInteraction's implementation.
- In the implementation of custom interactions like FractionInputInteractionView, NumericInputInteractionView, RatioInputInteraction, MathExpressionInteraction and TextInputInteractionView we initialize hint text as empty string in its init block. So, instead of assigning empty string to hint text we can assign the default hint for that interaction which will solve our problem of disappearing hint text.

Accessibility support for Rich-text based images. #3432

- For rich-text based images we need to make sure that whenever talkback is turned on the user should know what that image is trying to convey.
- So, to achieve this accessibility support for rich-text based images we can replace the images with text whenever talkback is turned on.
- We add an additional boolean parameter i.e. *isTalkbackEnabled* parameter in handleTag method in CustomHtmlContentHandler.kt.
- isTalkbackEnabled will be passed as true if talkback is turned on and to check that we can use AccessibilityService.isScreenReaderEnabled().
- In the implementation of handleTag, first we get the text for image from “alt-with-value” and if it’s not null and isTalkbackEnabled is true then we replace the image with text which we get from “alt-with-value”.

```
  /** Handler interface for a custom tag and its attributes. */
  interface CustomTagHandler {
    /**
     * Called when a custom tag is encountered. This is always called
after the closing tag.
     *
     * @param attributes the tag's attributes
     * @param openIndex the index in the output [Editable] at which
this tag begins
     * @param closeIndex the index in the output [Editable] at which
this tag ends
     * @param output the destination [Editable] to which spans can be
added
     * @param isTalkbackEnabled [Boolean] to check whether
screenReader is turned on or not
     * @param imageGetter a utility to load image drawables if needed
by the handler
     */
    fun handleTag(
      attributes: Attributes,
      openIndex: Int,
      closeIndex: Int,
      output: Editable,
      handlingForScreenReader: Boolean,
      imageRetriever: ImageRetriever
    ) {
    }
```

Improve accessibility in Pin Verification Screen #2627

- We will remove the separate textview used currently to show an error message.
- Instead of that we will set the error message to pin_password_input_edit_text which is using Material TextInputLayout in PinPasswordActivityPresenter.
- In the extension function onTextChanged of pin_password_input_edit_text, we will check that if length of inputted text is equal to length of correct pin then we will set an error message i.e. Incorrect pin .
- By setting an error message whenever a user enters a wrong pin it will show the error inside edittext only which solves the issue of talkback not reading incorrect pin text.

Unable to exit overflow menu (exploration + revision card) #3894

- We will add a bottom sheet to replace the current overflow menu so that users can exit from it when talkback is turned on.
- We create a new file to create a bottom sheet for ExplorationActivity and RevisionActivity.
- To implement the bottom sheet we need to create a BottomSheetOptionsMenu click listener and need to replace the overflow menu with a new bottom sheet implementation inside ExplorationActivity.

Add label for RecentlyPlayedActivity #2824

- To correctly add a relevant label for RecentlyPlayedActivity, we have to set the title in every activity/fragment from which RecentlyPlayedActivity could open.
- We need to set the title in RecentlyPlayedActivity, ProfileProgressActivity and HomeActivity.
- As per PR #3065 if we directly pass the title string via intent instead of proto then the screen will flicker.
- The issue with passing string is that it won’t support localization properly because of which the screen flickers as the title needs to be updated before it can be displayed with the correct content (which requires an asynchronous operation to complete).
- So, we will create a new proto which we send in an intent to another activity to set the label for RecentlyPlayedActivity. The benefit of using proto is that it makes it easier to introduce localization support for the label.

Incorrect output for rich-text in Hints and Solution #3906

- Currently, html is not parsed when we are setting the content description for hints and solutions.
- So, we have to use parseOppiaHtml function to parse html text but we can’t directly call our existing parseOppiaHtml as it required some extra parameters which we can’t give inside HintsViewModel (Reference PR #4218).
- Currently, parseOppiaHtml requires htmlContentTextView and rawString as parameters. So, in HintsViewModel we are only able to pass rawString such that parseOppiaHtml will return the parsed text from the given html text.
- Here, we will overload parseOppiaHtml in HtmlParser such that we can use it inside HintsViewModel.

Current implementation of parseOppiaHtml

```
  /**
   * Parses a raw HTML string with support for custom Oppia tags.
   *
   * @param rawString raw HTML to parse
   * @param htmlContentTextView the [TextView] that will contain the
returned [Spannable]
   * @param supportsLinks whether the provided [TextView] should
support link forwarding (it's recommended not to use this for
[TextView]s that are within other layouts that need to support
clicking (default false)
   * @return a [Spannable] representing the styled text.
   */

fun parseOppiaHtml(
  rawString: String,
  htmlContentTextView: TextView,
  supportsLinks: Boolean = false,
  supportsConceptCards: Boolean = false
): Spannable { ... }
```

New overloaded implementation of parseOppiaHtml

```
/**
 * Parses a raw HTML string without any additional parameters.
 *
 * This should be used in cases where standard HTML tags need to be
removed, but never rendered. For strings requiring rendering, use the
other implementation of [parseOppiaHtml].
 *
 * @param rawString raw HTML to parse
 * @return a [Spannable] representing text without any html tags
*/
fun parseOppiaHtml(rawString: String): Spannable { ... }
```

- In parseOppiaHtml we need to call fromHtml function from CustomHtmlContentHandler.kt but currently the fromHtml function requires additional parameters which we can't give inside our overloaded version of parseOppiaHtml so we need to overload fromHtml function.

  Current implementation of fromHtml function in CustomHtmlContentHandler

```
/**
  * Returns a new [Spannable] with HTML parsed from [html] using the specified [imageRetriever]
  * for handling image retrieval, and map of tags to [CustomTagHandler]s for handling custom tags. All possible custom tags must be registered in the [customTagHandlers] map.
  */
fun <T> fromHtml(
   html: String,
   imageRetriever: T,
   customTagHandlers: Map<String, CustomTagHandler>
): Spannable where T : Html.ImageGetter, T : ImageRetriever { … }
```

New overloaded implementation of fromHtml

```
/**
  * Returns a human-readable  representation of the provided raw HTML.
  *
  * This method is useful for generating text that can be read by screen readers for accessibility use cases.
  *
  * Returns a new [Spannable] with HTML parse from [html]
  */
fromHtml(html: String): Spannable { ... }
```

- Then we can directly call parseOppiaHtml inside the computeHintListDropDownContentDescription method of HintsViewModel.

Add accessibility support for ImageRegionSelectionInteraction #3712

- To add accessibility support we need to set the content description to each image region by adding a content description to each view in addRegionViews() function of ClickableAreasImage.kt.
- Now, to get contentDescription we have to add a new field in the json file as well as the proto file for ImageRegionSelectionInteraction.
- Inside 13.json and 13.textproto (domain/assets/) we will add a new field below the label i.e. content description where we mention the content description for each image region.

13.json

```
...
 "imageAndRegions": {
              "value": {
                "imagePath": "Screen Shot 2015-02-18 at 3.51.49
```

```
AM_height_400_width_816.png",
                "labeledRegions": [{
                  "label": "Mercury",
                  "contentDescription": "This is a content
description for Mercury planet",
                  "region": {
                    "regionType": "Rectangle",
                    "area": [[0.010981758125126362,
0.09856376051902771], [0.04871760681271553, 0.17073357105255127]]
                  }
                }, {
                  "label": "Venus",
                  "region": {
                    "regionType": "Rectangle",
                    "area": [[0.03692515566945076,
0.1755448877811432], [0.10060440003871918, 0.2958278954029083]]
                  }
                },
...
```

13.textproto

```
customization_args {
        key: "imageAndRegions"
        value {
          custom_schema_value {
            image_with_regions {
              image_path: "Screen Shot 2015-02-18 at 3.51.49
AM_height_400_width_816.png"
              label_regions {
                label: "Mercury"
                contentDescription: "This is a content description
for Mercury planet"
                region {
                  region_type: RECTANGLE
                  area {
                    upper_left {
                      x: 0.010981758
                      y: 0.09856376
                    }
                    lower_right {
                      x: 0.048717607
                      y: 0.17073357
                    }
                  }
                }
              }
```

interaction_object.proto

```
message ImageWithRegions {
  // A string that represents a filepath which will be prefixed with
'[exploration_id]/assets.
  string image_path = 1;

  // A region of an image, including its shape and coordinates.
  message LabeledRegion {
    // The name for the region.
    string label = 1;

  message Region {
      // Represents the different Region Types.
      enum RegionType {
        REGION_TYPE_UNSPECIFIED = 0;
        RECTANGLE = 1;
      }

      // A shape value for a particular region.
      RegionType region_type = 1;

      // A rectangle normalized so that the coordinates are within
the range [0,1].
      message NormalizedRectangle2d {
        // Origin is top-left, increasing x is to the right,
increasing y is down.
        Point2d upper_left = 1;
        Point2d lower_right = 2;
      }

      NormalizedRectangle2d area = 2;
    }

    Region region = 2;
    string content_description = 3;
  }
  repeated LabeledRegion label_regions = 2;
}
```

- After that we can set this content description to ImageRegion by getting a json object in StateRetriever (for JSON; for protos, the field will already be populated and ready-to-use).
- In app/utility/ClickableAreasImage we will set the content description to each clickable area.

Rich-text hyperlink text improvement #3895

- We can change the url text to more descriptive text in the textproto and json of particular exploration which have hyperlink text.
- In GJ2rLXRKD5hw_2.json and GJ2rLXRKD5hw_2.textproto, Learn more link text should be change to more descriptive text i.e. *“This concept card demonstrates overall concept card functionality”*

```
{
  "topic_id": "GJ2rLXRKD5hw",
  "topic_name": "Fractions",
  "page_contents": {
    "subtitled_html": {
      "content_id": "content",
      "html": "<p>Description of subtopic is here.
<oppia-noninteractive-skillreview
skill_id-with-value=\"&amp;quot;5RM9KPfQxobH&amp;quot;\"
text-with-value=\"&amp;quot;This concept card demonstrates overall
concept card
functionality&amp;quot;\"></oppia-noninteractive-skillreview></p>"
    },
…
```

[Android only] Test data changes

We need to make changes in test data for ImageRegionSelectionInteraction. We have to add a new field for content description in 13.json and 13.textproto.

Also, In GJ2rLXRKD5hw_2.json and GJ2rLXRKD5hw_2.textproto, Learn more link text should be change to more descriptive text i.e. *“This concept card demonstrates overall concept card* *functionality"*

[Android only] Testing library changes

Utility to test activity rotation should be available for this project. PR #2527

[Android only] Script & CI changes

No changes are required in CI.

Documentation changes

A wiki page on how to audit the app for accessibility will be added. It also includes my approach on how I decided which issues to fix and how. These should be documented such that anyone can perform the same steps and yield the same findings.

Testing Plan

Note: Currently there is no option to check flow of a screen reader on Android by automated testing. In the following end-to-end testing plan I have added test cases to check content description and also to check whether data persists on screen rotation or not.

Feature testing

Does this feature include non-trivial user-facing changes? **Yes**

Robolectric/Espresso and Manual Tests

#	Test name	Initial setup step	Step	Expectation
1.	textInputInteraction_checkContentDescription	Open exploration player	Launch any activity from exploration player with TextInputInteraction	Any activity from exploration player will be opened with Text as input
			Click on TextInputInteraction	Keyboard should be shown to enter text
			Click back	Get back to normal screen without keyboard
			Check content description of TextInputInteraction	Returns error if content description not match with given content description
2.	fractionInputInteraction_check	Open exploration	Launch any activity from exploration player with	Any activity from exploration player will be opened with Fraction as input

	ContentDescription	player	FractionInputInteraction	
			Click on FractionInputInteraction	Keyboard should be shown to enter text
			Click back	Get back to normal screen without keyboard
			Check content description of FractionInputInteraction	Returns error if content description not match with given content description
3.	numericInputInteraction_checkContentDescription	Open exploration player	Launch any activity from exploration player with NumericInputInteraction	Any activity from exploration player will be opened with numeric as input
			Click on NumericInputInteraction	Keyboard should be shown to enter text
			Click back	Get back to normal screen without keyboard
			Check content description of NumericInputInteraction	Returns error if content description not match with given content description
4.	ratioInputInteraction_checkContentDescription	Open exploration player	Launch any activity from exploration player with RatioInputInteraction	Any activity from exploration player will be opened with ratio as input
			Click on RatioInputInteraction	Keyboard should be shown to enter text
			Click back	Get back to normal screen without keyboard
			Check content description of RatioInputInteraction	Returns error if content description not match with given content description
4.	MathExpressionInteraction_checkContentDescription	Open exploration player	Launch any activity from exploration player with MathExpressionInteraction	Any activity from exploration player will be opened with MathExpression as input
			Click on MathExpressionInteraction	Keyboard should be shown to enter text
			Click back	Get back to normal screen without keyboard
			Check content description of MathExpressionInteraction	Returns error if content description not match with given content description
5.	checkContentDescription_Res	Open Ratio and	Check content description for	Returns error if it has empty string as content description

	umeLessonChapterDescription	Proportions story summary	resume_lesson_chapter_description_textview	
6.	isDataPersistAfterScreenRotationForTextInput_ExplorationPlayer	Open exploration player.	Launch any activity from exploration player with TextInputInteraction	Any activity from exploration player will be opened with Text as Input
			Enter any answer and store it in some variable	Answer will be entered
			Perform screen rotation	Screen will be rotated
			Get the answer after screen rotation and assert that it equals the previously entered answer or not.	Returns error if previously entered answer is not retained after screen orientation.
7.	isDataPersistAfterScreenRotationForMathExpression_ExplorationPlayer	Open exploration player.	Launch any activity from exploration player with MathExpression	Any activity from exploration player will be opened with MathExpression as Input
			Enter any answer and store it in some variable	Answer will be entered
			Perform screen rotation	Screen will be rotated
			Get the answer after screen rotation and assert that it equals the previously entered answer or not.	Returns error if previously entered answer is not retained after screen orientation.
8.	isDataPersistAfterScreenRotationForImageRegionSelection_ExplorationPlayer	Open exploration player.	Launch ImageRegionSelectionactivity from exploration player	ImageRegionSelectionActivity from exploration player will be opened
			Select any image region and get the label from that region and store it in some variable.	Image region will be selected
			Perform screen rotation	Screen will be rotated
			Get the label after screen rotation and assert that it equals the label stored in a variable.	Returns error if previously selected image region is not retained after screen orientation.
9.	isDataPersistAfterScreenRotationForDragAndDropAnswer_ExplorationPlayer	Open exploration player.	Launch any activity from exploration player which has DragAndDropAnswer	Any activity from exploration player will be opened which has DragAndDropAnswer
			Randomly Drag and Drop and store its sequence in some variable.	Randomly options will be Dragged and Dropped.

			Perform screen rotation	Screen will be rotated
			Get the sequence after screen rotation and assert that it equals the previously selected sequence.	Returns error if previously selected sequence is not retained after screen orientation.
10.	isDataPersistAfterScreenRotationForSelectionInteraction_ExplorationPlayer	Open exploration player.	Launch any activity from exploration player which has SelectionInteraction	Any activity from exploration player will be opened which has SelectionInteraction
			Select any answer and store it in some variable.	Any answer will be selected
			Perform screen rotation	Screen will be rotated
			Get the answer after screen rotation and assert that it equals the previously selected answer or not.	Returns error if previously selected answer is not retained after screen orientation.
11.	isDataPersistAfterScreenRotationForRatioInputInteraction_ExplorationPlayer	Open exploration player.	Launch any activity from exploration player with RatioInputInteraction	Any activity from exploration player will be opened with Ratio as Input
			Enter any answer and store it in some variable	Answer will be entered
			Perform screen rotation	Screen will be rotated
			Get the answer after screen rotation and assert that it equals the previously entered answer or not.	Returns error if previously entered answer is not retained after screen orientation.
12.	isDataPersistAfterScreenRotationForTextInput_QuestionPlayer	Open question player.	Launch any activity from question player with TextInputInteraction	Any activity from question player will be opened with Text as Input
			Enter any answer and store it in some variable	Answer will be entered
			Perform screen rotation	Screen will be rotated
			Get the answer after screen rotation and assert that it equals the previously entered answer or not.	Returns error if previously entered answer is not retained after screen orientation.
13.	isDataPersistAfterScreenRotationForSelectionInteraction_QuestionPlayer	Open question player.	Launch any activity from question player which has SelectionInteraction	Any activity from question player will be opened which has SelectionInteraction
			Select any answer and store it in some variable.	Any answer will be selected

			Perform screen rotation	Screen will be rotated
			Get the answer after screen rotation and assert that it equals the previously selected answer or not.	Returns error if previously selected answer is not retained after screen orientation.
14.	isPlatformParameterWorking_for_ConfigurationChange_Enable	Open Exploration player.	Set feature flag for Configuration change to true	Platform parameter for configuration change will be set to true
			Launch any activity from exploration player with any input interaction.	Any activity from question player will be opened which has SelectionInteraction
			Select any answer and store it in some variable.	Any answer will be selected
			Perform screen rotation	Screen will be rotated
			Get the answer after screen rotation and assert that it equals the previously selected answer or not.	Returns error if previously selected answer is not retained after screen orientation.
15.	isPlatformParameterWorking_for_ConfigurationChange_Disable	Open Exploration player.	Set feature flag for Configuration change to false	Platform parameter for configuration change will be set to false
			Launch any activity from exploration player with any input interaction.	Any activity from question player will be opened which has SelectionInteraction
			Select any answer and store it in some variable.	Any answer will be selected
			Perform screen rotation	Screen will be rotated
			Get the answer after screen rotation and assert that it is not equal to the previously selected answer or not.	Returns error if previously selected answer is matched after screen orientation.
16.	bottomSheetOptionsMenu_working	Open Exploration player	Click on menu icon from toolbar	Bottom sheet options menu will open
			Click on options	Click on options from bottom sheet
			Check if options menu is opened or not	Returns error if options menu is not opened
			Click on help	Click on help option from bottom sheet
			Check if help menu is opened or not	Returns error if help menu is not opened

			Click close	Click close option from bottom sheet
			Check if bottom sheet options menu is closed or not	Returns error if bottom sheet options menu is not closed
17.	testStateFragment_showHintAndSolution_hasCorrectContentDescription	Open Exploration Player	Enter incorrect answer and click submit	Enters incorrect answer and click on submit button
			Enter incorrect answer and click submit	Enters incorrect answer and click on submit button
			Click on hints and solution	Click on hint and solution which will open hint and solution activity.
			Click on reveal hint button	Click on the reveal hint button which will show a hint.
			Check content description of hint_list_drop_down_icon	Returns error if the content description won't match with our given content description.
18.	imageRegionSelectionInteracton_hasCorrectContentDescription	Open Exploration Player	Launch ImageRegionSelectionInteraction from exploration player	ImageRegionSelectionInteraction activity will be launched
			Click on specific image region (here planet)	Click on given image region i.e. specific planet
			Check content description of image region	Returns error if that image region has incorrect content description.
19.	checkLabel_recentlyPlayedActivity	Open Home Page	Click on view all on Home page activity	This action will open a recently played activity page.
			Check the title of RecentlyPlayedActivity	Returns error if title doesn't match with desired string
			Click on View all in ProfileProgressActivity	Opens up recently played activity page
			Check the title of RecentlyPlayedActivity	Returns error if title doesn't match with desired string
20.	checkContentDescription_richTextImages	Open exploration player	Open any activity from exploration player which has rich text images	Any activity with rich text images will be opened
			Check content description of rich text image	Returns error if content description of rich text image doesn't match with desired string.

Implementation Plan

Milestone Table (include both PRs and other actions that need to be taken prior to launch)

Milestone 1: Ensure that screens with similar UI components have a consistent UX throughout, for both uninhibited users and users with vision related impairment. Ensure that the app is fully accessible for any combination of settings in the Android OS.

No.	Description of PR / action	Prereq PR numbers	Target date for work start	Target date for PR creation	Target date for PR to be merged
1	Improve flow of lessons tab #3896	N/A	20th July	21st July	22nd July
2	Remove blank space from Ratio and Proportion resume lesson activity.	N/A	21st July	22nd July	25th July
3	#3892 Locked chapter item selection change is incorrect	N/A	22nd July	24th July	28th July
4	Pin verification screen #2627	N/A	24th July	27th July	30th July
5	Unable to exit overflow menu (exploration + revision card) #3894	N/A	27th July	31st July	3rd August
6	Rich-text hyperlink text improvement #3895	N/A	31st July	2nd August	5th August
7	Add label for RecentlyPlayedActivity #2824	#3095	2nd August	5th August	8th August
I will be traveling from 9th to 16th August					
8	Add accessibility support for ImageRegionSelectionInteraction #3712	N/A	16th August	18th August	22nd August

Tests needed to run (manual tests are also included) to verify that all issues are solved for milestone 1.

Test name	Explanation	Result
Check Lessons flow tab [Manual test]	Enable talkback reader and go to lesson tab to check the flow of talkback reader. Pass the test if repetitive content description is removed.	
Check pin verification screen Error [Manual test]	Add a new profile or select an already created profile (except admin) and enter an incorrect pin (make sure that talkback is turned on). Pass the test if talkback reads the error message for incorrect pin.	
Check locked chapter item selection [Manual test]	Go to any story and click on the story name to get a summary of the story. Turn on talkback reader and click on the go to unlocked chapter. Pass the test if the focus of the talkback reader shift to the unlocked chapter.	
checkContentDescription_ResumeLessonChapterDescription [Unit test]	Check the content description of Resume Lesson chapter description for Ratios and proportions where description of resume lesson chapter is empty.	
bottomSheetOptionsMenu_working [Unit test]	Check all options in bottom sheet menu by clicking it one by one and at last click on exit to check whether the user is able to exit from menu or not.	
Check hyperlink text [Manual test]	Check the hyperlink text in the Fractions of a group (revision section of Fractions topic) using accessibility scanner.	
checkLabel_recentlyPlayedActivity [Unit test]	Checks whether the title of Recently played activity matches with desired label.	
imageRegionSelection_hasCorrectContentDescription [Unit Test]	Check the content description for image regions.	

Milestone 2: Ensure that data always persists on screen rotations.

No.	Description of PR / action	Prereq PR numbers	Target date for work start	Target date for PR creation	Target date for PR to be merged
1	Create a parameter gatting for configuration change	N/A	26th August	29th August	3rd September
2	Handle configuration change using onSavedInstance	N/A	29th August	5th September	8th September
3	Use onSavedInstance to handle configuration changes on FractionInputInteraction, NumericInputInteraction, TextInputInteraction, MathExpressionInteraction, RatioExpressionInteraction	N/A	5th September	10th September	15th September
4	Use onSavedInstance to handle configuration changes on ImageRegionSelectionInteraction	N/A	10th September	14th September	18th September
5	Use onSavedInstance to handle configuration changes on DragAndDropInteraction	N/A	14th September	18th September	22nd September
6	Use onSavedInstance to handle configuration changes on SelectionInputInteraction	N/A	18th September	22nd September	27th September
7	Incorrect output for rich-text in Hints and Solution #3906	#3852	22nd September	27th September	2nd October
8	Add meaningful content description to every edit box in exploration and question player.	N/A	27th September	1st October	6th October
9	Accessibility support for Rich-Text images. #3432	N/A	1st October	7th October	13th October
10	Create documentation on accessibility Audit.	N/A	7th October	13th October	20th October

Tests needed to run (manual tests are also included) to verify that all issues are solved for milestone 2.

Test name	Explanation	Result
isPlatformParameterWorking_for_ConfigurationChange_Enable	Checks whether configuration change works for interactions when the platform parameter is turned on.	
isPlatformParameterWorking_for_ConfigurationChange_Disable	Checks whether configuration change works for interactions when the platform parameter is turned off.	
isDataPersistAfterScreenRotationForTextInput_ExplorationPlayer	Checks whether data persists when activity with TextInputInteraction is rotated in the exploration player.	
isDataPersistAfterScreenRotationForMathExpression_ExplorationPlayer	Checks whether data persists when activity with MathExpressionInteraction is rotated in the exploration player.	
isDataPersistAfterScreenRotationForImageRegionSelection_ExplorationPlayer	Checks whether data persists when activity with ImageRegionSelection is rotated in the exploration player.	
isDataPersistAfterScreenRotationForDragAndDropAnswer_ExplorationPlayer	Checks whether data persists when activity with DragAndDropAnswer interaction is rotated in the exploration player.	
isDataPersistAfterScreenRotationForSelectionInteraction_ExplorationPlayer	Checks whether data persists when activity with SelectionInteraction is rotated in the exploration player.	
isDataPersistAfterScreenRotationForRatioInputInteraction_ExplorationPlayer	Checks whether data persists when activity with RatioInputInteraction is	

	rotated in the exploration player.	
isDataPersistAfterScreenRotationForTextInput_QuestionPlayer	Checks whether data persists when activity with TextInputInteraction is rotated in the question player.	
isDataPersistAfterScreenRotationForSelectionInteraction_QuestionPlayer	Checks whether data persists when activity with SelectionInteraction is rotated in the question player.	
testStateFragment_showHintAndSolution_hasCorrectContentDescription	Checks whether the Hints and Solution has the correct content description or not.	
textInputInteraction_checkContentDescription	Checks content description for TextInputInteraction.	
fractionInputInteraction_checkContentDescription	Checks content description for FractionInputInteraction.	
numericInputInteraction_checkContentDescription	Checks content description for NumericInputInteraction.	
ratioInputInteraction_checkContentDescription	Checks content description for RatioInputInteraction.	
MathExpressionInteraction_checkContentDescription	Checks content description for MathExpressionInteraction.	
checkContentDescription_richTextImages	Check content description for rich text images.	

Future Work

Note*: This section is mainly for reference (since it is understood that items in this section will not be part of the GSoC project). Proposals will primarily be evaluated based on the implementation plan above.*

A11Y issues which are not covered in this project from CLAM Team

#4318: Unable to scroll down the home page when magnification is turned on
#3122: Dark theme make colors of the images distorted

#3893: Need DiffUtils with RecyclerView for better experience
#4235: Fix accessibility issues for ClickableSpan