

Climber App Development Report

Student Name: Dang Khoa Le

Student ID: 103844421

Tutorial: Wednesday 4.30-6.30 PM

1. Introduction

This report documents the development of an Android climbing-scoring app named Climber, specifically designed for a local climbing club. The app facilitates the scoring of a climber's progress as they ascend / simulate a modified climbing wall, following specific rules for scoring and penalties. Key features of the app include multi-language support, orientations handling, user feedback through visual effects, and responsive UI interactions.

2. Development Plan and Time Logs

Planning and Research:

- **Research:** Investigated climbing scoring systems and Android development best practices. Explored methods for implementing scoring logic, run-time language switching, and interactive UI/UX elements.
- **Planning:** Structured the app, outlining the required layouts, core logic, and components. Focused on creating a user-friendly and efficient interface.

Development:

- **Layouts:** Created main layout for both portrait and landscape orientations (layout/activity_main.xml for default linear view and layout-land/activity_main.xml for constraint view).
- **Logics:** Implemented the core logic in MainActivity.kt handling score tracking, button functionality, and language switching.
- **Multi-language Support:** Added dynamic language support with separate strings.xml files for English, Spanish, Vietnamese, and Chinese with function implemented in MainActivity.kt to allow language switching .
- **UI Effects:** Integrated ripple effects for button interactions through click_effect.xml.

Testing:

- Ensured consistency of scoring logic across different scenarios.
- Tested on different screen orientations to ensure interactive UI/UX.
- Verified multi-language functionality, allowing effective language switching.
- Ensured that all UI components functioning seamlessly, correctly and providing appropriate feedback.

Time Logs:

- **Day 1+2:** Research and initial setup - 3 hours
 - **Day 2+3:** Layout design and button functionality - 4 hours
 - **Day 4:** Multi-language support - 5 hours
 - **Day 5:** Bug fixes, code refinement - 4 hours
 - **Day 6+7+8:** UI improvement (background, logo image + ripple effects), and documentation - 5 hours
-

3. Key Design Decisions

1. Button Actions and Scoring Logic:

The app's scoring system is divided into three zones (blue, green, red) representing different levels of difficulty. Points awarded per climb increase as the climber

progresses through these zones. Buttons (Climb, Fall, Reset) are linked to listeners that manage these state transitions, ensuring the logic is consistent with the climbing rules.

2. **Orientation Handling:**

Separate layouts for portrait and landscape orientations ensure that the app remains functional and aesthetically pleasing regardless of screen orientation. Orientation changes preserve the score and climber's position through the use of 'onSaveInstanceState' and 'onRestoreInstanceState'.

3. **Multi-language Support:**

The app supports four languages: English, Spanish, Vietnamese, and Chinese. A language spinner allows users to switch languages dynamically from defined strings.xml files. The app's configuration is updated accordingly, ensuring that the user interface reflects the chosen language.

4. **UI Effects:**

Implemented ripple effects to enhance user interaction. This visual feedback helps users understand that their input is registered.

4. Issues Encountered and Solutions

1. **Language Switching Errors:**

Issue: The app experienced crashes when switching languages, particularly with the spinner (drop-down menu).

Solution: Refactored the language switching logic by encapsulating it in a separate private method. Ensured the app's configuration is updated before recreating the activity.

2. **Orientation Resetting Score:**

Issue: Score reset on orientation change.

Solution: Implemented 'onSaveInstanceState' and 'onRestoreInstanceState' to preserve the score and current hold values.

3. **Deprecated Libraries:**

Issue: Attempted to use an outdated library (EasingInterpolator) for a button effect / animation, causing compatibility issues.

Solution: Removed the dependency and opted for Android's built-in ripple effect for simplicity and avoiding dependency on deprecated code.

5. Reflection

What Worked Well:

- The app successfully handles the core functionality of scoring climbers, with proper responsive mechanisms like color changing and ripple effects.
- The multi-language feature enhances usability for non-English speakers, making the app accessible to a wider audience.
- The logic for preserving the app state on orientation changes ensures a smooth user experience.

What Could Be Improved:

- The UI design could be enhanced further with more visual elements and animations to make the app more engaging.

6. Technical Usage

MainActivity.kt:

The MainActivity.kt file implements the primary logic for handling user interactions, managing the score, and preserving the app's state across orientation changes. Key methods and their roles:

a) onCreate:

- Initializes the UI components and sets up key variables (such as integer variable score, currentHold, and boolean variable hasFallen).
- Restores the state if the app is being reloaded after a configuration change.
- Sets up listeners for the buttons (Climb, Fall, Reset).
- Initializes the language spinner.
- Handles the climbButton listener to increment the score and move to the next hold, considering the restrictions on when the climber can climb. If the climber is between specific zones, the score is updated according to the rules (e.g., score can't be incremented when climber has fallen or current hold not in between 0 and 8; for hold between 1 and 3, add score by 1; when hold between 4 and 6, add score by 2; and hold between 7 and 9, add score by 3), and the score display is updated accordingly.
- Handles the fallButton listener to decrease the score (by 3), reset the climber's hold position when the climber falls, and set the hasFallen boolean value to true.
- Handles the resetButton listener to reset the score, climber's hold position, and hasFallen value.

```
// Listen to activity on climb button
climbButton.setOnClickListener { it:View?
    if (!hasFallen && currentHold in 0..8) { // Prevent climbing after falling and current hold
        currentHold++ // Increment current hold
        score += when (currentHold) {
            in 1..3 -> 1
            in 4..6 -> 2
            in 7..9 -> 3
            else -> 0
        }
        score = minOf(score, 18) // Ensure the score does not exceed 18
        updateScoreDisplay(scoreText)
    }
    Log.d(tag: "ClimbingApp", msg: "Climb button clicked, current hold: $currentHold, score: $score")
}
```

```
// Listen to activity on fall button
fallButton.setOnClickListener { it:View?
    if (currentHold in 1..8) {
        score = maxOf(0, score - 3)
        currentHold = 0 // Reset current hold
        hasFallen = true // Set the fall flag
        updateScoreDisplay(scoreText)
    }
    Log.d(tag: "ClimbingApp", msg: "Fall button clicked, score: $score")
}
```

```
resetButton.setOnClickListener { it:View?
    score = 0 // Reset
    currentHold = 0 // Reset
    hasFallen = false // Reset
    updateScoreDisplay(scoreText)
    Log.d(tag: "ClimbingApp", msg: "Reset button clicked, score reset to 0")
}
```

The above images illustrate the functions that define the actions upon click listeners to the climbButton, fallButton and resetButton.

b) setupLanguageSpinner:

- Configures the language spinner to display available languages.
- Sets the spinner's selection based on the current locale.
- Updates the app's language dynamically when a new language is selected from the spinner.

c) changeLanguage:

- Updates the app's locale configuration based on the selected language.
- Using updateConfiguration method to update the existing global configuration to update the language upon user request listened on the spinner. Although this method has been deprecated on Android API level 25 due to its inconsistency, especially triggers misconfiguration when integrating different fragments (pages), this however is fine in this application context, as there only 1 main activity resource.
- Using recreate to apply the language change on run-time (overriding the existing global language configuration).

```
private fun changeLanguage(languageCode: String) {
    val locale = Locale(languageCode)
    Locale.setDefault(locale)
    val config = Configuration(resources.configuration)
    config.setLocale(locale)
    baseContext.resources.updateConfiguration(config, baseContext.resources.displayMetrics)
    // The updateConfiguration() method was deprecated in Android API level 25 (Nougat).
    // This method was deprecated because it manipulates the app's global resources, which can lead to inconsistencies.
    // However, this still works fine in this application as the function allow app's global configuration overriding the language config.
    // Which we only use 1 fragment (MainActivity) so the language switching function wouldn't trigger any misconfigurations.
    recreate() // Restart the activity to apply changes
}
```

d) updateScoreDisplay:

- Updates the score display and changes the color of the score based on the current hold (zone).
- The color changes to blue in holds 1-3, green in holds 4-6, and red in holds 7-9, reflecting the difficulty of each zone.

e) onSaveInstanceState:

- Saves the current score, hold position, and fall status to ensure that the app's state is preserved during orientation changes or other configuration changes.

layout/activity_main.xml:

Defines the app's vertical layout for the portrait mode. It includes a background image 'climb.webp', buttons (Climb, Fall, Reset), and a language spinner.

layout-land/activity_main.xml (landscape):

A constraint layout designed for landscape orientation, with a similar structure to the portrait layout but arranged horizontally. It also sets a different background image from 'drawable/mount.avif'.

drawable/click_effect.xml:

Defines the ripple effect for button clicks, enhancing user feedback.

strings.xml (various languages):

Defines the app's text elements in different languages with English (as default) from values/strings.xml, Spanish from values-es/strings.xml, Vietnamese from values-vi/strings.xml, Chinese from values-zh/strings.xml).

English (values/strings.xml)	Spanish (values-es/strings.xml)	Vietnamese (values-vi/strings.xml)	Chinese (values-zh/strings.xml)
<string name="app_name">Climbing App</string>	<string name="app_name">Aplicación de Escalada</string>	<string name="app_name">攀岩应用</string>	<string name="app_name">Ứng Dụng Leo Núi</string>
<string name="score">Score: </string>	<string name="score">Puntuación: </string>	<string name="score">成绩: </string>	<string name="score">Điểm Số: </string>
<string name="climb">Climb</string>	<string name="climb">Escalar</string>	<string name="climb">爬</string>	<string name="climb">Leo</string>
<string name="fall">Fall</string>	<string name="fall">Caer</string>	<string name="fall">落下</string>	<string name="fall">Té</string>
<string name="reset">Reset</string>	<string name="reset">Restablecer</string>	<string name="reset">重启</string>	<string name="reset">Làm Lại</string>

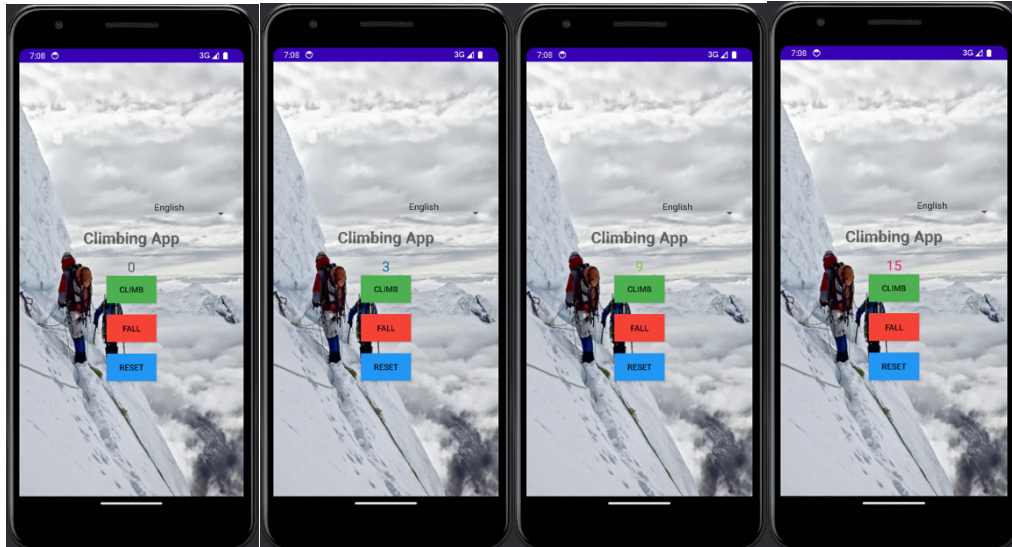
In these 4 strings.xml files above, the string id to "app_name", "score", "climb", "fall" and "reset" have been configured on 4 languages to allow the app update languages on run-time and applying on the app components.

values/colors.xml:

Define colours with an id to be used, in order to enhance the UI.

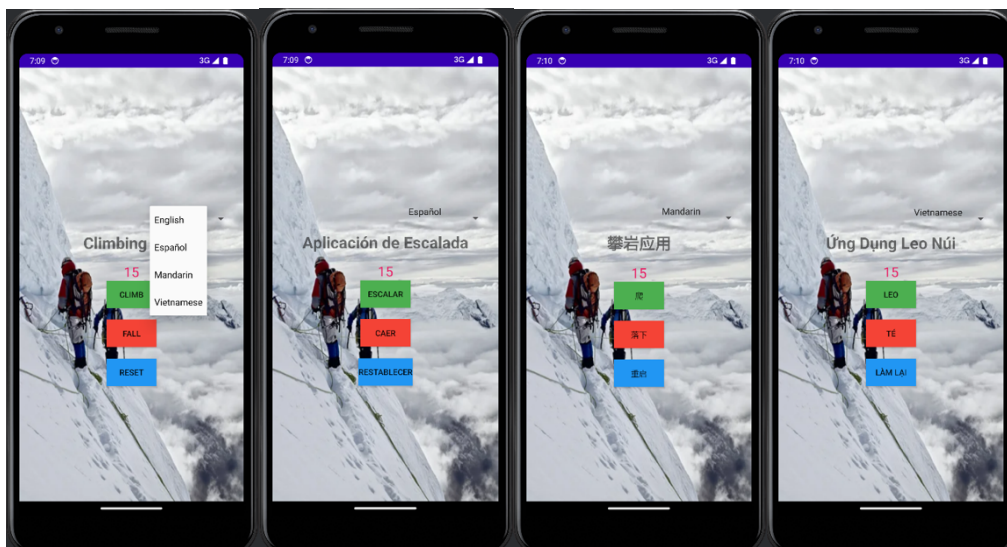
7. App Demo

a. Scoring feature demo



The above figures illustrate the climber's scoring feature functioning with color changing upon different score holds. The app has effectively using blue-green-red color signifying current holding zones as per specified by the client (a local climbing club).

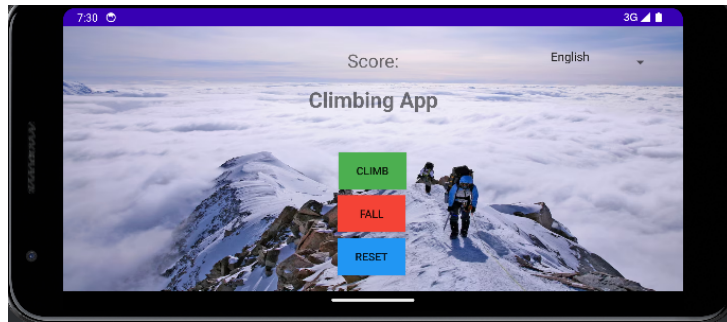
b. Dynamic multi-language switching demo



The above figure representing the language switching feature with user being able to select different languages from the spinner, including English (default), Spanish, Mandarin and Vietnamese.

The reason behind choosing these 4 languages is because these are 4 languages I can speak.

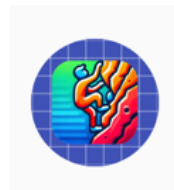
c. Horizontal (landscape / constraint view) orientation demo



The above image illustrates the app displayed with landscape orientation, 2 major differences to the portrait mode is the usage of different background image and

d. Logo

For a better recognition experience of the user about the app, I changed the Image Asset of the app logo to the drawable/logo.png image (instead of using the default Android logo), as well as changing the background color of the logo to purple, which suits the overall theme of the app, enhancing user's interaction experience while using the app.



8. Logs - Testing

Testing and Debug Logs (Log.d method) have been deployed over the program in MainActivity.kt to listen for button triggers and ensure score and currentHold variables are deployed correctly upon event actions.

```

2024-08-22 11:35:07.183 14997-14997 ClimbingApp com.example.climber D Reset button clicked, score reset to 0
2024-08-22 11:35:30.101 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 1, score: 1
2024-08-22 11:35:30.232 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 2, score: 2
2024-08-22 11:35:30.492 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 3, score: 3
2024-08-22 11:35:30.784 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 4, score: 5
2024-08-22 11:35:31.000 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 5, score: 7
2024-08-22 11:35:31.211 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 6, score: 9
2024-08-22 11:35:31.442 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 7, score: 12
2024-08-22 11:35:31.676 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 8, score: 15
2024-08-22 11:35:31.819 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 9, score: 18
2024-08-22 11:35:31.971 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 9, score: 18
2024-08-22 11:35:32.127 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 9, score: 18
2024-08-22 11:35:32.301 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 9, score: 18
2024-08-22 11:35:32.441 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 9, score: 18
2024-08-22 11:35:33.690 14997-14997 ClimbingApp com.example.climber D Reset button clicked, score reset to 0
2024-08-22 11:35:35.774 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 1, score: 1
2024-08-22 11:35:36.111 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 2, score: 2
2024-08-22 11:35:36.376 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 3, score: 3
2024-08-22 11:35:36.580 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 4, score: 5
2024-08-22 11:35:36.842 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 5, score: 7
2024-08-22 11:35:37.019 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 6, score: 9
2024-08-22 11:35:37.258 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 7, score: 12
2024-08-22 11:35:38.039 14997-14997 ClimbingApp com.example.climber D Fall button clicked, score: 9
2024-08-22 11:35:39.849 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 0, score: 9
2024-08-22 11:35:41.281 14997-14997 ClimbingApp com.example.climber D Reset button clicked, score reset to 0
2024-08-22 11:35:42.264 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 1, score: 1
2024-08-22 11:35:42.443 14997-14997 ClimbingApp com.example.climber D Climb button clicked, current hold: 2, score: 2

```

These are the method used to print out logs on different buttons with the log id "ClimbingApp":

```

Log.d("ClimbingApp", "Climb button clicked, current hold: $currentHold, score: $score")
Log.d("ClimbingApp", "Fall button clicked, score: $score")
Log.d("ClimbingApp", "Reset button clicked, score reset to 0")

```

Examining the log events (from Logcat) allowing developer to keep track on triggers and data consistency over the app deployment, identify bugs and navigating issues, especially at the early stage to prevent exhaust effort fixing errors.

9. Acknowledgment

GeeksforGeeks. (2024). Ripple effect on Android button. GeeksforGeeks. <https://www.geeksforgeeks.org/ripple-effect-on-android-button/>

GeeksforGeeks. (2024). How to change the whole app language in Android programmatically?.

GeeksforGeeks. <https://www.geeksforgeeks.org/how-to-change-the-whole-app-language-in-android-programmatically/>

Stack Overflow. (2016). Android context.getResources.updateConfiguration() deprecated [Discussion post]. Stack Overflow. <https://stackoverflow.com/questions/40221711/android-context-getresources-updateconfiguration-deprecated>

No Generative AI (genAI) tools were used for this task.

10. Conclusion

The Climber app successfully meets the requirements of a simple climbing scoring system, providing a user-friendly interface with dynamic multi-language support, robust state management across device orientations, and responsive UI elements.

11. Link to Code Repository

The code for this project can be accessed on GitHub Classroom at:

<https://github.com/SoftDevMobDev-2024-Classrooms/assignment01-Lelekhoa1812/tree/main>.