Nguyen Khanh Huyen
04/10/2025

Task ##P/C – Spike: Climbing Wall Scoring App

Goals: This task aims to upskill in Android app development by building a multi-layout, multi-language scoring app for a climbing club. It covers UI design, state management, localization, and debugging.

This section is an overview highlighting what the task is aiming to teach or upskill.

- Designing responsive layouts for portrait and landscape orientations
- Implementing scoring logic with zone-based rules
- Preserving app state across device rotations
- Supporting multiple languages
- Using logs for debugging

The following list outlines the goal broken down into more specific knowledge gaps involved in the goal.

- Android layout management (ConstraintLayout, LinearLayout)
- State persistence with onSaveInstanceState
- Localization using string resources
- Dynamic UI updates (changing text color based on zone)
- Button click handling and logic implementation
- Logging for debugging (Log.d)
- Resource management (colors, drawables)

Tools and Resources Used

This section lists related software, tools, libraries, API's, and other resources used for this knowledge gap.

- Android Studio
- Kotlin programming language
- Android SDK and API documentation (developer.android.com)
- Stack Overflow (https://stackoverflow.com/)
- Android Logcat

Knowledge Gaps and Solutions

This section presents the listed knowledge gaps and their solutions with supporting images, screenshots and captions where appropriate/required.

Gap 1: Responsive Layouts for Different Orientations

Steps:
1, Create activity_main.xml for portrait and activity_main.xml-land for landscape.
2, Use different layout types (ConstraintLayout for portrait, LinearLayout for landscape).
3, Test UI on emulator/device in both orientations.

Gap 2: State Persistence Across Rotations

Steps:
1, Override onSaveInstanceState and onRestoreInstanceState in MainActivity.
2, Save score and current hold in the bundle.
3, Restore values after rotation to prevent reset.

Gap 3: Localization (Multi-language Support)

Steps:
1, Add string resources in res/values/strings.xml and res/values-vi/strings.xml.
2, Reference strings in layouts and code using @string/....
3, Test switching languages in device settings.

Gap 4: Dynamic UI Updates and Zone Logic

Steps:
1, Implement logic to update score and hold based on button clicks.
2, Change score text color depending on the current zone.
3, Prevent invalid actions (falling before hold 1, climbing after fall, etc.).

Gap 5: Debugging with Logs

Steps:
1, Use Log.d statements in button click handlers and state changes.
2, Monitor Logcat for debugging information during development.

Open Issues and Recommendations

1. Issue: Localization may not cover all UI elements or dynamic strings.

➔ Recommendation: Review all strings and ensure they use resource references. Test with both languages.

2. Issue: State persistence may not handle all edge cases (e.g., app killed in background).

➔ Recommendation: Consider using ViewModel or SharedPreferences for more robust state management.

3. Issue: UI may not scale well on tablets or devices with unusual aspect ratios.

➔ Recommendation: Test on multiple device sizes and adjust layouts as needed.

I. Introduction

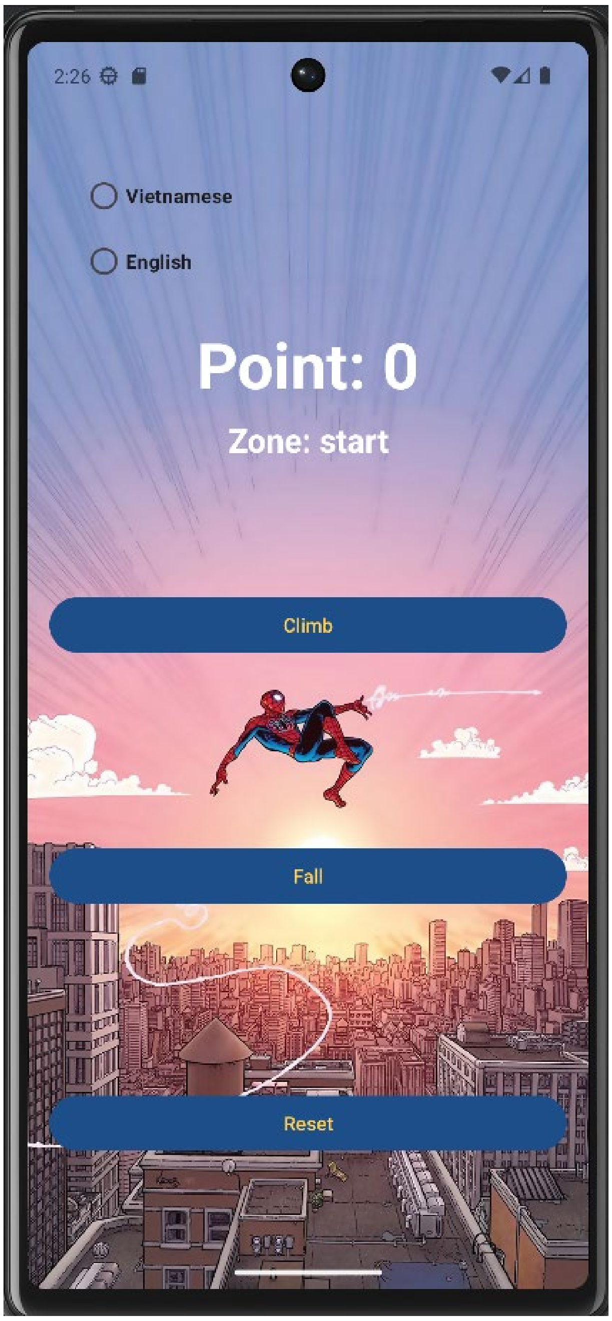
This report details the development of ClimbingApp, a single-activity mobile application created for local climbers to track and manage their climbing progress. Developed using Kotlin and Android Studio, the app demonstrates essential mobile development concepts such as debugging, state preservation, localization, and dynamic user interface updates. ClimbingApp allows users to monitor scores based on hold positions, adjust points according to climbing zones (Blue, Green, Red), handle resets and falls, and switch seamlessly between English and Vietnamese. The application also ensures that all relevant activity states—including scores, selected language, and current zone—are preserved during device rotations, providing a reliable and user-friendly experience.

II. Logic and Layout design

UX/UI:

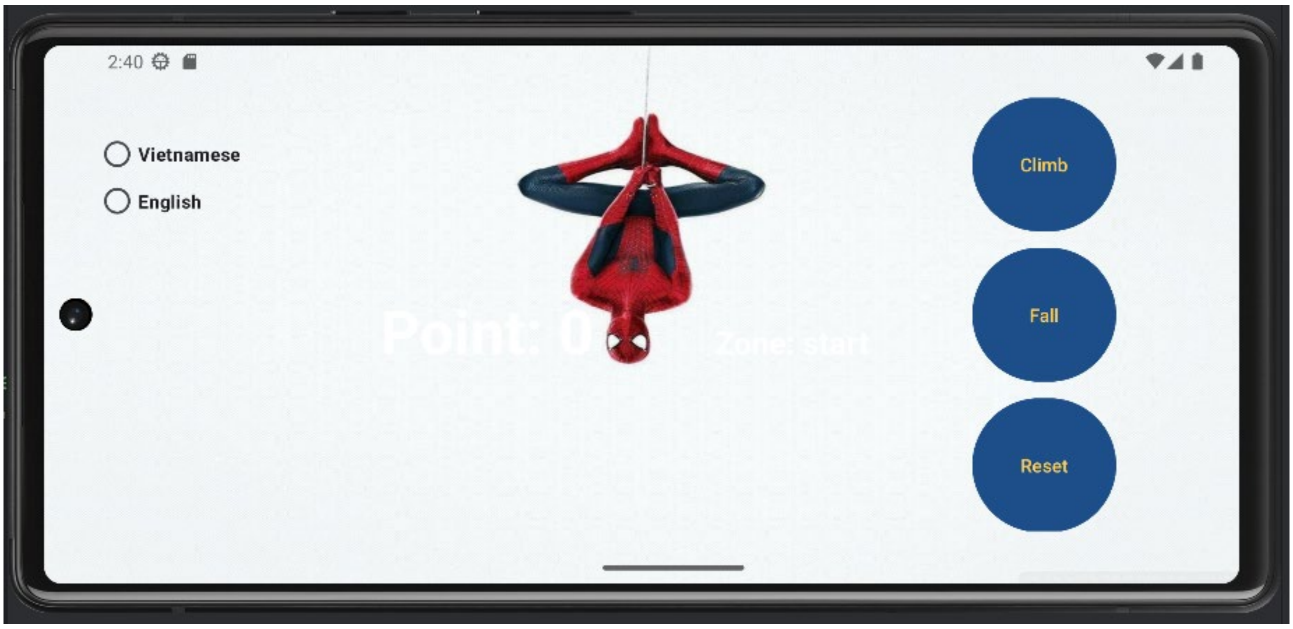
- Portrait:

- Uses a vertical arrangement for controls and information with three main elements (score, zone, buttons) are stacked for easy one-handed use.
- Language selection is placed at the top in form of radio button for quick access.
- Score and zone are displayed centrally with large, bold text for visibility.



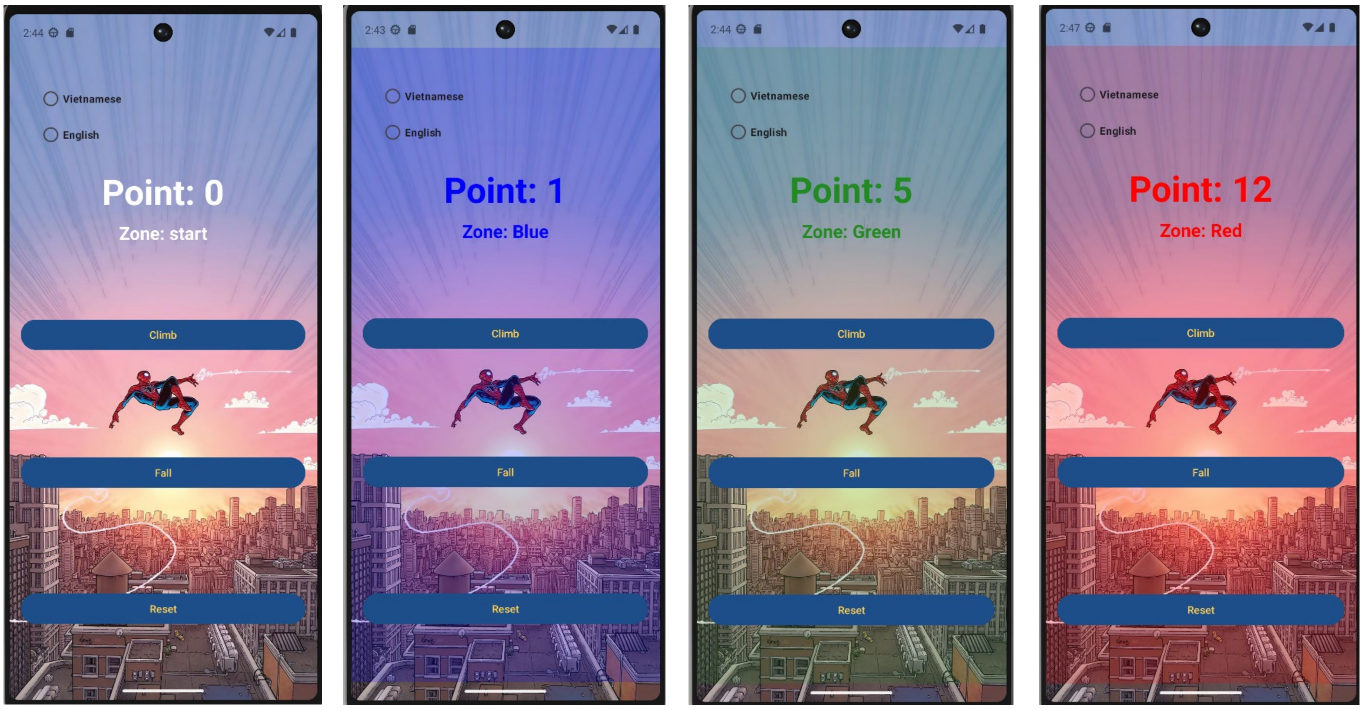
- Landscape:

- Uses a FrameLayout root for flexible layering (background, overlays, controls).
- Controls are organized horizontally in a LinearLayout for better use of wide screens.
- Language selection is on the left, main score and zone display in the center, and action buttons on the right.
- Zone overlay (ImageView) provides dynamic color feedback based on the climber’s current zone.
- Buttons are large and vertically stacked for easy access.



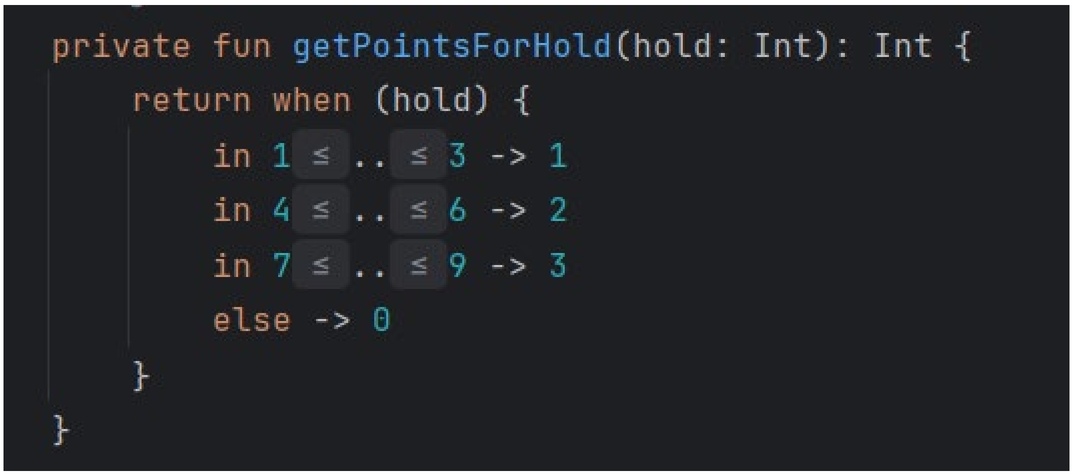
Color:

The application divides the climbing wall into distinct color zones: Blue, Green, and Red, each zone is visually represented by a unique color overlay on the background, helping climbers identify their current position.
- The zoneOverlay ImageView dynamically updates its visibility and color based on the climber's progress:
Ground Zone: No overlay, score is 0.
- Blue Zone: Blue-tinted overlay appears between hole 1 and 3 (scores 1 – 3).
- Green Zone: Green-tinted overlay appears from hole 4 to 6 (scores 5 – 9).
- Red Zone: Red-tinted overlay appears for the last 3 holes (scores 12 – 18).
The current zone name is displayed in the zoneTextView and updates in real-time as the climber moves. Zone changes also affect scoring logic, with each zone potentially offering different point increments or penalties. Immediate visual feedback ensures climbers are always aware of their current zone and its impact on scoring.



Scoring System:

- The score is tracked and displayed in the scoreTextView
- Pressing the Climb button increases the score by 1, moving the climber to the next hold and potentially changing the zone.
- Pressing the Fall button decreases the score by 1 (but not below 0), simulating a fall and possibly moving to a lower zone.
- Pressing the Reset button sets the score back to 0 and resets the zone to start.
- Each zone (Ground, Blue, Green, Red) is determined by the current score and updates both the overlay and zone name.
- All score changes trigger immediate UI updates and a Toast message to inform the user of the new score or action result. The score is preserved during device rotation and configuration changes to maintain user progress.

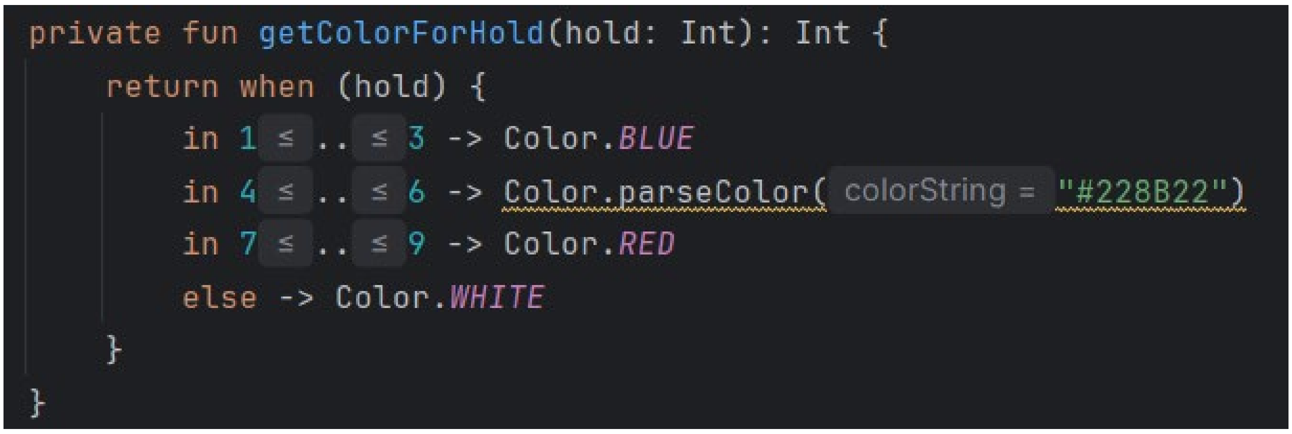


```
private fun getPointsForHold(hold: Int): Int {
    return when (hold) {
        in 1 ≤ .. ≤ 3 -> 1
        in 4 ≤ .. ≤ 6 -> 2
        in 7 ≤ .. ≤ 9 -> 3
        else -> 0
    }
}
```

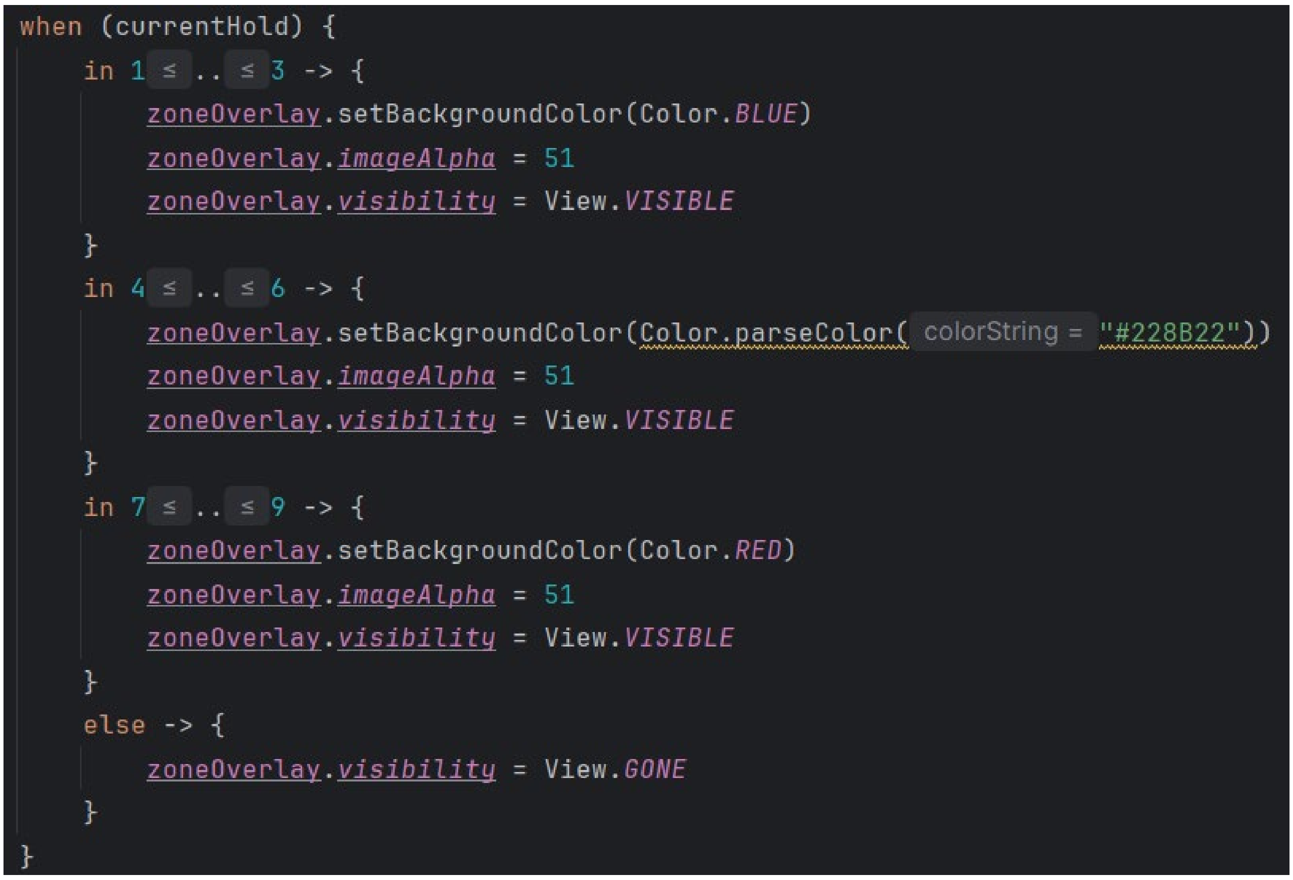
Background color:

The overlay logic in your code uses the climber's current hold (calculated from the score) to determine the overlay's color and visibility:
- The function updateUI() calls getHoldFromScore(score) to get the current hold.
- The overlay's color and transparency are set with setBackgroundColor and imageAlpha.
- The overlay is shown/hidden by setting zoneOverlay.visibility.

➔ The overlay visually represents the climber's zone by changing color and visibility according to the current hold, which is derived from the score.



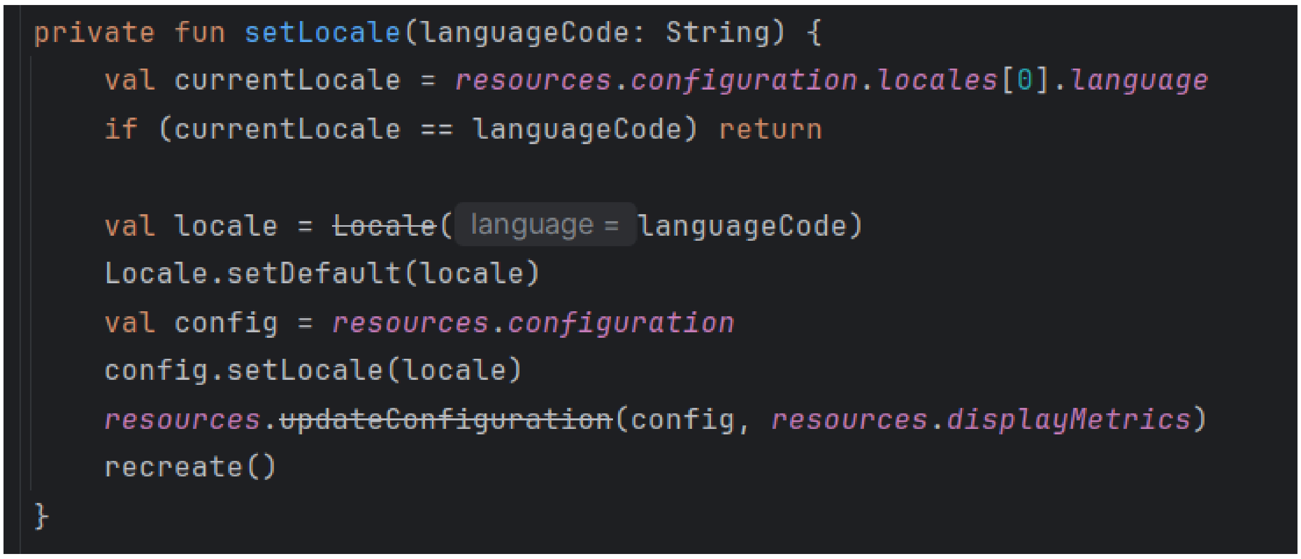
```
private fun getColorForHold(hold: Int): Int {
    return when (hold) {
        in 1 ≤ .. ≤ 3 -> Color.BLUE
        in 4 ≤ .. ≤ 6 -> Color.parseColor( colorString = "#228B22")
        in 7 ≤ .. ≤ 9 -> Color.RED
        else -> Color.WHITE
    }
}
```



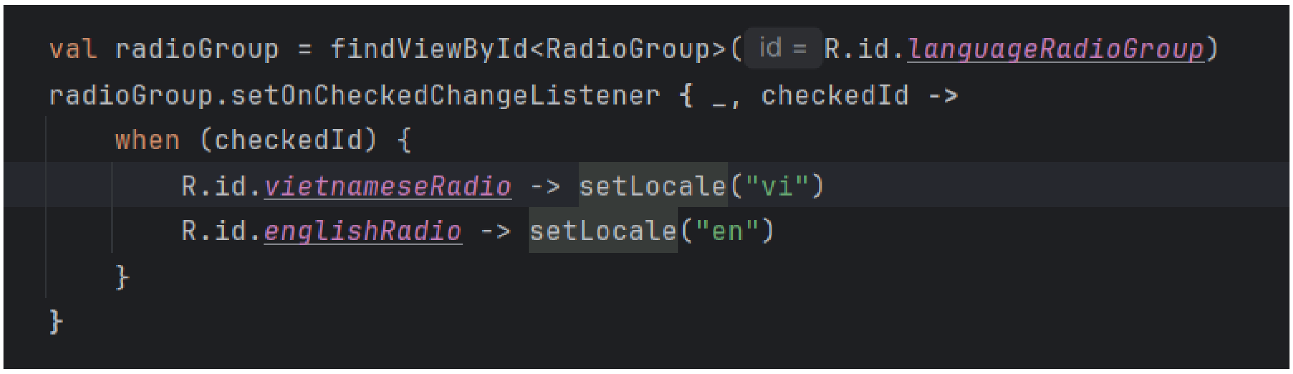
```
when (currentHold) {
    in 1 ≤ .. ≤ 3 -> {
        zoneOverlay.setBackgroundColor(Color.BLUE)
        zoneOverlay.imageAlpha = 51
        zoneOverlay.visibility = View.VISIBLE
    }
    in 4 ≤ .. ≤ 6 -> {
        zoneOverlay.setBackgroundColor(Color.parseColor( colorString = "#228B22"))
        zoneOverlay.imageAlpha = 51
        zoneOverlay.visibility = View.VISIBLE
    }
    in 7 ≤ .. ≤ 9 -> {
        zoneOverlay.setBackgroundColor(Color.RED)
        zoneOverlay.imageAlpha = 51
        zoneOverlay.visibility = View.VISIBLE
    }
    else -> {
        zoneOverlay.visibility = View.GONE
    }
}
```

Localization:

To switch between 2 language (english and vietnamese), I use radio buttons in the top left corner for easy selection. Then Android resource files are used for each language (strings.xml in values and values-vi). And UI text is retrieved with getString(R.string.key), automatically using the current locale.



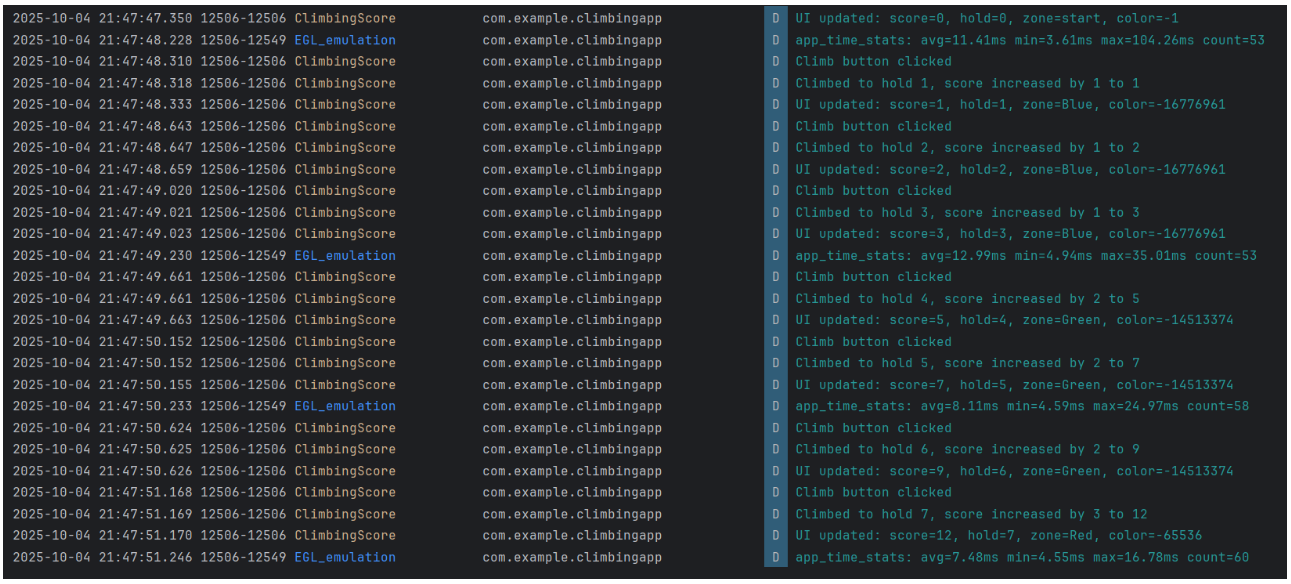
```
private fun setLocale(languageCode: String) {
    val currentLocale = resources.configuration.locales[0].language
    if (currentLocale == languageCode) return

    val locale = ~~Locale~~( language = languageCode)
    Locale.setDefault(locale)
    val config = resources.configuration
    config.setLocale(locale)
    resources.~~updateConfiguration~~(config, resources.displayMetrics)
    recreate()
}
```



```
val radioGroup = findViewById<RadioGroup>( id = R.id.languageRadioGroup)
radioGroup.setOnCheckedChangeListener { _, checkedId ->
    when (checkedId) {
        R.id.vietnameseRadio -> setLocale("vi")
        R.id.englishRadio -> setLocale("en")
    }
}
```

III. Debug and test

- Log statements are used for runtime tracing to check if the app work as expected.



```
2025-10-04 21:47:47.350 12506-12506 ClimbingScore           com.example.climbingapp              D  UI updated: score=0, hold=0, zone=start, color=-1
2025-10-04 21:47:48.228 12506-12549 EGL_emulation           com.example.climbingapp              D  app_time_stats: avg=11.41ms min=3.61ms max=104.26ms count=53
2025-10-04 21:47:48.310 12506-12506 ClimbingScore           com.example.climbingapp              D  Climb button clicked
2025-10-04 21:47:48.318 12506-12506 ClimbingScore           com.example.climbingapp              D  Climbed to hold 1, score increased by 1 to 1
2025-10-04 21:47:48.333 12506-12506 ClimbingScore           com.example.climbingapp              D  UI updated: score=1, hold=1, zone=Blue, color=-16776961
2025-10-04 21:47:48.643 12506-12506 ClimbingScore           com.example.climbingapp              D  Climb button clicked
2025-10-04 21:47:48.647 12506-12506 ClimbingScore           com.example.climbingapp              D  Climbed to hold 2, score increased by 1 to 2
2025-10-04 21:47:48.659 12506-12506 ClimbingScore           com.example.climbingapp              D  UI updated: score=2, hold=2, zone=Blue, color=-16776961
2025-10-04 21:47:49.020 12506-12506 ClimbingScore           com.example.climbingapp              D  Climb button clicked
2025-10-04 21:47:49.021 12506-12506 ClimbingScore           com.example.climbingapp              D  Climbed to hold 3, score increased by 1 to 3
2025-10-04 21:47:49.023 12506-12506 ClimbingScore           com.example.climbingapp              D  UI updated: score=3, hold=3, zone=Blue, color=-16776961
2025-10-04 21:47:49.230 12506-12549 EGL_emulation           com.example.climbingapp              D  app_time_stats: avg=12.99ms min=4.94ms max=35.01ms count=53
2025-10-04 21:47:49.661 12506-12506 ClimbingScore           com.example.climbingapp              D  Climb button clicked
2025-10-04 21:47:49.661 12506-12506 ClimbingScore           com.example.climbingapp              D  Climbed to hold 4, score increased by 2 to 5
2025-10-04 21:47:49.663 12506-12506 ClimbingScore           com.example.climbingapp              D  UI updated: score=5, hold=4, zone=Green, color=-14513374
2025-10-04 21:47:50.152 12506-12506 ClimbingScore           com.example.climbingapp              D  Climb button clicked
2025-10-04 21:47:50.152 12506-12506 ClimbingScore           com.example.climbingapp              D  Climbed to hold 5, score increased by 2 to 7
2025-10-04 21:47:50.155 12506-12506 ClimbingScore           com.example.climbingapp              D  UI updated: score=7, hold=5, zone=Green, color=-14513374
2025-10-04 21:47:50.233 12506-12549 EGL_emulation           com.example.climbingapp              D  app_time_stats: avg=8.11ms min=4.59ms max=24.97ms count=58
2025-10-04 21:47:50.624 12506-12506 ClimbingScore           com.example.climbingapp              D  Climb button clicked
2025-10-04 21:47:50.625 12506-12506 ClimbingScore           com.example.climbingapp              D  Climbed to hold 6, score increased by 2 to 9
2025-10-04 21:47:50.626 12506-12506 ClimbingScore           com.example.climbingapp              D  UI updated: score=9, hold=6, zone=Green, color=-14513374
2025-10-04 21:47:51.168 12506-12506 ClimbingScore           com.example.climbingapp              D  Climb button clicked
2025-10-04 21:47:51.169 12506-12506 ClimbingScore           com.example.climbingapp              D  Climbed to hold 7, score increased by 3 to 12
2025-10-04 21:47:51.170 12506-12506 ClimbingScore           com.example.climbingapp              D  UI updated: score=12, hold=7, zone=Red, color=-65536
2025-10-04 21:47:51.246 12506-12549 EGL_emulation           com.example.climbingapp              D  app_time_stats: avg=7.48ms min=4.55ms max=16.78ms count=60
```