This assignment does not count toward the final grade.

# Assignment 1

Start Assignment

- Due 25 Aug by 23:59
- Points 15
- Submitting a file upload
- File types pdf
- Available until 1 Sep at 23:59

## New task for 2024

Please note that assessment has changed for this unit in 2024, and as such revisions might need to be made to the spec. Students are required to be checking this page, the discussion forums and lecture discussions in case of changes.

While this task states "This assignment does not count toward the final grade", it actually will, however students are only permitted to be awarded a single overall mark in portfolio units.

# Assessment overview

- **Title:** Assignment 1: Leading the Way
- This task addresses the following Unit Learning Outcomes:
  - ULO1 Explain the key differences between development of systems to run on mobile devices and on typical personal computing or internet-based environments, and apply this knowledge in the design of mobile device software.
  - ULO2 Design effective applications for a mobile device by taking into consideration the underlying hardware-imposed restrictions such as screen size, memory size and processor capability.
  - ULO3 Build, test and debug graphical applications for mobile devices by using the standard libraries that are bundled as part of the developers' toolkit for the mobile device.
- **Individual assessment**
- **Assessment type:** Written report and code
- **Weighting**: 15%
- **Word count (or equivalent)**: roughly 5-8 pages of content, logs and diagrams
- **Due date**: Please see Canvas submission space.
- **Submission procedure:** For this assignment you may submit as many times as you like before the due date. Your assignment will be checked for plagiarism with a similarity report generated through Turnitin. The teaching staff can see each of your submissions, and the associated Similarity Report. Should you choose to submit more than once, the teaching staff will be marking the final uploaded file.

- **Feedback date**: 2 weeks after due date.
- **Referencing style:** APA 7
- **Who will see your work?** This work will be assessed by your tutor.

# Assessment explanation

This app is a simple one-activity game that makes use of layouts and localisation.

On completion of this task, you will demonstrate that you are able to work with a single activity app, handle activity states, use Logs, create layouts, implement listeners effectively and enable localisation.

**Important** Academic integrity

It is your responsibility to make sure your work is your own. If you provide work to another student you will be penalised the same as the student who uses your work. This includes failure of the assignment, an official penalty notice, and possible exclusion from the University.

Paying someone to do your assignment for you is one of the highest forms of Academic Misconduct and it will not be tolerated. Possible penalties for this include exclusion from the university and revocation of your degree.

## Task

This is an individual task with a fixed deadline -- no submissions will be accepted after the closing date.

If Turnitin indicates 25% or higher similarity (or a teacher detects a similar level of similarity) with other published work and/or work submitted by other students, the submission will be reviewed and a mark potentially withheld or delayed. Similar sections will **not** be assessed, i.e., if the introduction is copied from elsewhere, the introduction criteria will be awarded 0.

- PDF: A completed report for this task -- the report should set out your development plan/time logs, the key design decisions made, issues encountered and the solutions considered/used, and a reflection on what worked well/what could be improved. Please ensure that explanation/decisions relating to the rubric elements are covered. Any use of genAI must be acknowledged, with prompts and outputs included in an appendix. If genAI was not used, the statement "No Generative AI tools were used for this task" must be included in the acknowledgement. Please refer to the Unit Outline for more details around resubmission/redo of work where the teaching staff feel that genAI has been inappropriately used.
- The report **must** contain a link to code in a repository visible to teaching staff on GitHub Classroom (see **GitHub Classroom links**

(https://swinburne.instructure.com/courses/61570/pages/github-classroom-links) for the invite link). Repos are not to be public, nor should tutors be added manually -- **the task will not be assessed until the repo is in a suitable location**.

- Before or after submission, a demo in class is required, otherwise a penalty will be applied. The demo will be assessed according to the **Demo guidelines** (https://swinburne.instructure.com/courses/61570/pages/demo-guidelines) . Note that these will need to be spread over 2 weeks of classes; it is expected that students aiming for high marks will demo in the first week, as this will allow you to take on feedback ahead of submission.

# Instructions

1. Start with planning and researching. This will include working through the tutorial exercises and quizzes. Outline and describe the major things you will need to do to create your app.
2. Develop an app that meets criteria as shown below. Do not go overboard -- make use of your class time to clarify requirements if need be.
3. The report needs to cover key knowledge gaps and process that was undertaken in this task, however note that this should be pitched at a fellow student (e.g., "Go to the Android website and download Android Studio" is not required).
4. Submissions must be uploaded as a single PDF.

# The app

A local climbing club has a modified climbing wall and scoring process. They are seeking a simple app to score climbs.

- The app needs to have three buttons and one score value at a minimum.
- Set the initial score to 0. When the "Climb" button is clicked, the score increases by a number of points as shown below. When the "Fall" button is clicked, the score decreases by 3. The "Reset" button sets the score back to 0.
- The wall has 9 holds.
  - The climber cannot fall until they have reached the first hold (hold 1). Once the climber has fallen, they cannot climb further.
  - Once the climber has reached the top hold (hold 9), they cannot lose points for falling, and the scoring can only reset from there.
  - When the climber is between holds 1-3, this is the blue zone and each hold scores 1 point.
  - When the climber is between holds 4-6, this is the green zone and each hold scores 2 points.
  - When the climber is between holds 7-9, this is the red zone and each hold scores 3 points.
  - Climbers cannot go backwards.
- The scores should be kept between 0 and 18 (so a fall cannot make the score negative). The score colour should change depending on the zone.
- Two layouts are needed, one for portrait and one for landscape. A different layout type must be used for each.
- The scores must be saved so that on rotation they do not reset. This can be done using saveInstanceState.

- The app must also be usable in a second language.
- Logs must be demonstrated in the code for useful debugging purposes.

## Resources

Weeks 1-4 have set you up by isolating different parts of app development needed for this task.

If you would like to know more about the lead discipline in climbing, the following video might help (up to around 2:12, so about 40 seconds). Note our wall has 9 holds, not 40, and we are not timing climbers.


**COS30017 A1 2024**

| Criteria | Ratings | | | | Pts |
|---|---|---|---|---|---|
| UI/UX<br>Does the app look good? Does it provide a good experience for the user? | **2 to >1.5 Pts**<br>**Excellent**<br>Evidence of thought (incl. research/references) incorporated into UI/UX, as shown in report and app. App provides excellent experience for mobile users. | **1.5 to >1.0 Pts**<br>**Good**<br>Thought put into UI/UX, evidenced in report and app. App provides good experience for mobile users, however some minor improvements could be made. | **1 to >0.5 Pts**<br>**Needs improvement**<br>A solid start, however UI/UX require major improvement, in particular thinking about mobile users. Limited valuable discussion in report. | **0.5 to >0 Pts**<br>**No or minimal attempt**<br>UI/UX require major improvement, in particular thinking about mobile users. Limited to no discussion in report. | 2 pts |
| Functionality<br>Is the expected functionality implemented? Have debugging and testing been used to confirm correct behaviour? | **6 to >4.0 Pts**<br>**Excellent**<br>All required functionality implemented correctly, with thorough testing and debugging to confirm this. Extra functionality (commensurate with course progress) implemented and tested. Design/testing decisions clearly documented in report. | **4 to >2.0 Pts**<br>**Good**<br>All required functionality implemented correctly, with some testing and debugging to confirm this. Testing and debugging need minor improvement. Most design/testing decisions documented in report. | **2 to >1.0 Pts**<br>**Needs improvement**<br>Most required functionality implemented correctly, with some testing and debugging used. Testing and debugging need major improvement. Some design/testing decisions documented in report. | **1 to >0 Pts**<br>**No or minimal attempt**<br>Little to no functionality implemented correctly. No use of testing or debugging. | 6 pts |
| Code quality<br>Has care been taken with the code itself? Is the structure of the project sensible? | **3 to >2.0 Pts**<br>**Excellent**<br>Excellent attention to code style, including sufficient comments and documentation. Good use of Kotlin-specific features to produce elegant solution. Project structure conforms to Android style in all aspects. | **2 to >1.0 Pts**<br>**Good**<br>Good attention to code style, including some comments and documentation, however minor improvements could be made. Some elegant code however inconsistently observed. Project structure does not fully conform to Android style. | **1 to >0.5 Pts**<br>**Needs improvement**<br>Some attention to code style, however major improvements could be made wrt e.g., documentation, formatting and variable naming. Project structure does not conform to Android style in some aspects. | **0.5 to >0 Pts**<br>**No or minimal attempt**<br>Little to no attention to code style. Project structure does not conform to Android style. | 3 pts |
| Process and demo<br>Is there evidence of a structured software development process? Has a demo been undertaken with teaching staff? | **4 to >3.0 Pts**<br>**Excellent**<br>Evidence of a sensible and structured process, including time logs and commits with succinct descriptions. Report matches product and commit history. Demo is likely assessed at A/B level. | **3 to >2.0 Pts**<br>**Good**<br>Some evidence of a sensible and structured process, including time logs and commits, however with minor improvements required. Report matches product and commit history. Demo is likely assessed at B/C level. | **2 to >1.0 Pts**<br>**Needs improvement**<br>Little evidence of a sensible and structured process, possibly including a lack of commit history or poor commit messages. Report mostly matches product and commit history. Demo is likely assessed at C/D level. | **1 to >0 Pts**<br>**No or minimal attempt**<br>Little to no evidence of a sensible and structured process, e.g., zip file submitted. Report does not match product and commit history. Demo is likely assessed at D/E/F level. | 4 pts |
| | | | | | Total points: 15 |