<div align="center">

COSC354 Assignment 3
Luke Piper, Eszter Scarlett-Herbert, Riley Flanagan, Kristie Koorts

</div>

Testing and Coverage reports

Our group has been using the coverage reports generated by Codecov to reassess and improve the testing of our program. Initially, we created base unit tests for each of the functions in our program; as we implemented these tests, the generated Codecov reports outlined specifically which cases in each of our functions were being missed in testing. We used this information to create additional unit tests, where possible, to ensure the program was being rigorously tested.

Using the information generated by the coverage reports, we got our code coverage up to 89%. Some lines of code remain untested because of their difficulty to reach, for example, in process_raw.cpp, four error cases require invalid JSON data formats from the server response, and finding these cases proved incredibly difficult. Therefore, we decided it would not be worth our limited time to try to cover these niche error cases.

Static analysis using Codacy

Our group integrated Codacy into our development workflow as we developed our program. This enabled us to easily double-check the quality of each commit pushed to the repository. We managed to get our code to an A-grade Codacy standard, however initially we had Codacy reporting 50+ issues in our code related mostly to the style of our ReadMe. Therefore, we opted to exclude both the Readme and gitignore files from Codacy's analysis as we found the style issues it was highlighting to be irrelevant. The Codacy analysis has made it very easy for us to continually ensure that we uphold the quality of our code as we implement new changes.

Initial User Testing

Our group has begun the initial phase of user testing, by asking some of our peers, friends, and flatmates to play around with our app and answer our questionnaire. Some of the feedback we have received so far includes

    Positive feedback:
- The user interface is visually appealing.
- Intuitive to navigate around the map.
- Event markers are clearly displayed.

    Constructive feedback*:*
- Unclear how many events are in a given location.
- The app search feature is not intuitive enough.
- Would like the app to search based on the current map position. (i.e. events update with map movement)
- Unclear how to search by multiple fields.

As we continue development, we will actively seek to address the feedback we have received. Our group will continue user testing throughout the upcoming stages of development in order to refine and enhance the user experience of our application.

Development Progress
Our group is very close to finishing our application, and we have established a template code for the implementation of the remaining features. Additionally, we have established QML to C++ communication channels, implemented a basic search functionality, and created placeholders in the GUI for future search fields. The basic navigation features for the map have been implemented, as well as an auto-panning feature. Additionally, to optimise performance and reduce memory consumption, we have introduced marker grouping. These additions will mean that it will be efficient for us to implement the remaining features of our app, including the improvements identified as necessary from our user testing.

Cross-platform development
Our group has structured our application to streamline cross-platform development. The project generation code will adjust the build setup process depending on computer architecture. While this is not flawless, we have the foundational elements in place that act as guidelines for achieving this.

Retrospective
We have also included a retrospective of our group work so far, which is included in the GitHub repository.

ArcGIS License issues
We have implemented the use of the ArcGIS SDK in our project; however, because of this, we cannot currently release the application to the public. This is because our current ArcGIS license is only sufficient for development and testing. As we cannot distribute the ArcGIS SDK over GitHub due to the licensing, building and running of the application currently requires individual ArcGIS download. Qt Creator offers 'handsfree' library linking, meaning running the application through QT is the most straightforward option, as there would be no need to copy, move, or reference the ArcGIS SDK.

If we wish to expand the distribution of Eventures in the future, we could streamline this process by incorporating a commercial ArcGIS license. This license would allow for the distribution of 'point and click' ArcGIS software installation.