

Lab 3: Embedding the backbone

Lab Due Date: 7:59AM April 21, 2016

Grade Rubric: You will be graded on the following sections for credit.

Class structure (Create the foundation of the objects applicable to your project)	200
Event handling (incorporate user input specific to your project e.g. keyboard input, main UI buttons, etc. this doesn't have to be complete yet, just start on it)	200
Creativity (Appearance of your project up to this point)	200

Total

600 points

Note: This lab directly applies to your term project that can be team based with one other person. (Only do one if you're working with someone)

What you should have before starting:

At this point you should have a splash screen or opening screen for your project.

What you should have after this lab:

A high level structure for the logic of your project (not complete, just have the classes behind your project in either Qt or Unity). Some interaction with the user specific to your project (begin some of your core functionality). Keep up the appearance of your project, you should be building on the opening screen (splash screen) you made last week. Try and make it look nice and keep your readme up to date with fresh screenshots, and info on any new bugs that may have popped up during this lab.

Submission:

On blackboard, create a text submission that includes the following:

- Link to GitHub repository that includes this lab.
- Briefly explain or show your class structure and justify your decisions
- Briefly explain the event handling behind your project, how do you expect the user to use your application?

Class structure:

Begin to layout the foundation of your project. An example is if you're working on a card game you may have a main Game class, that contains a Deck of Cards. Where "Game", "Deck", and "Card" are all part of the class structure for the project. These classes don't need to include all of the methods / members that your project will need, just the high level class structure that you know that your project will require.

For those using Qt, the class structure and the logic behind each class should be contained in the C++, in .h and .cpp files that interact with the QML user interface side.

For those using Unity, the class structure and logic behind each class should be contained in the C#, in the .cs files.

Event Handling:

Determine how you want your project to be controlled. If you primarily want the user to navigate / use your application using the keyboard then start to work on handling keypress events. If you primarily want the user to navigate using mouse / touch input, begin developing the main UI behind your project, which could include buttons, text boxes, or other clickable regions.

Creativity:

This will be evaluated on a per project basis, based on what you described your project to be in your readme on GitHub. Are you trying new things and incorporating UI techniques that we've discussed so far?

Here's some cool projects that we hope will inspire you: