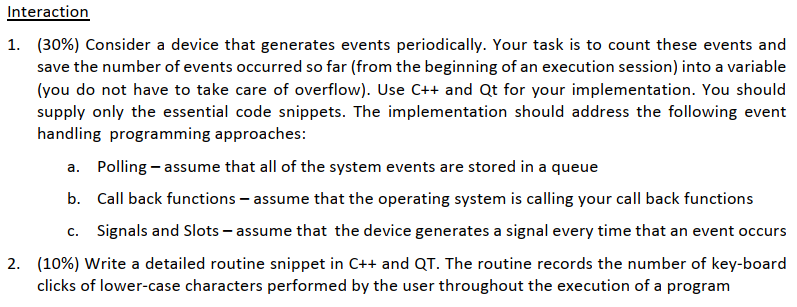
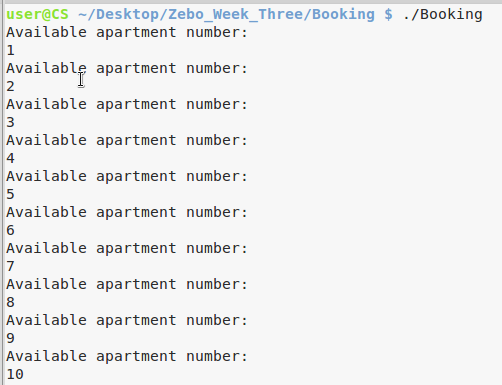
**Mid-Term Exam**

My student ID is A04907051, thus:

* 1) Your task is to develop a computer managed interactive system that enables users to find rental apartments online.

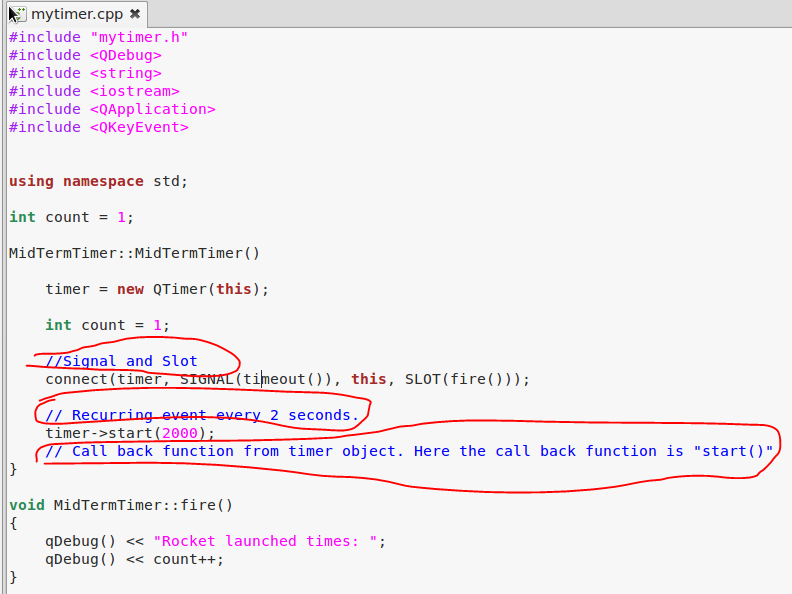


1. I wrote a system using timer to record the events:

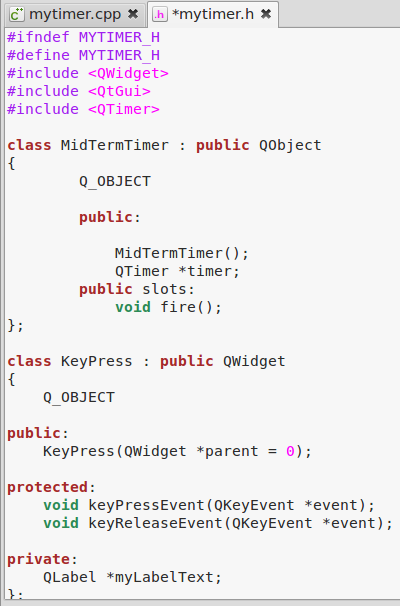


1. mytimer.cpp

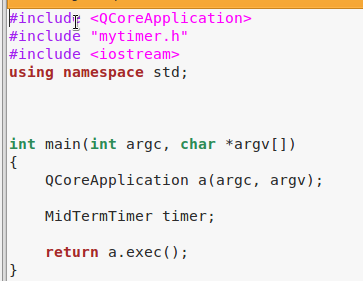
* please refer the red circled code
* there are answer the point a b c respectively.
* I use “count” variable to record the event times.



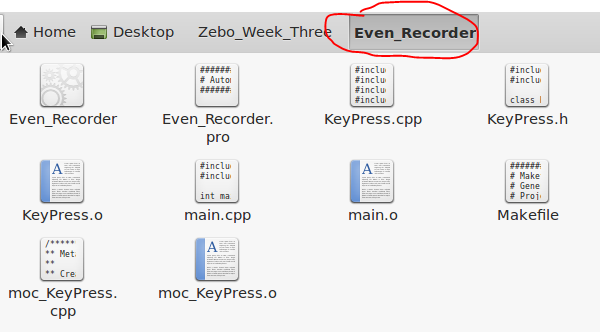
1. Head file:



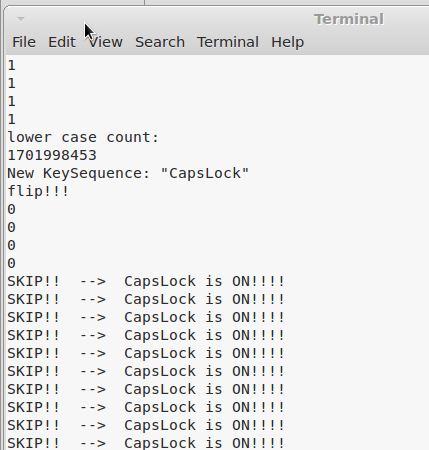
1. Main function:

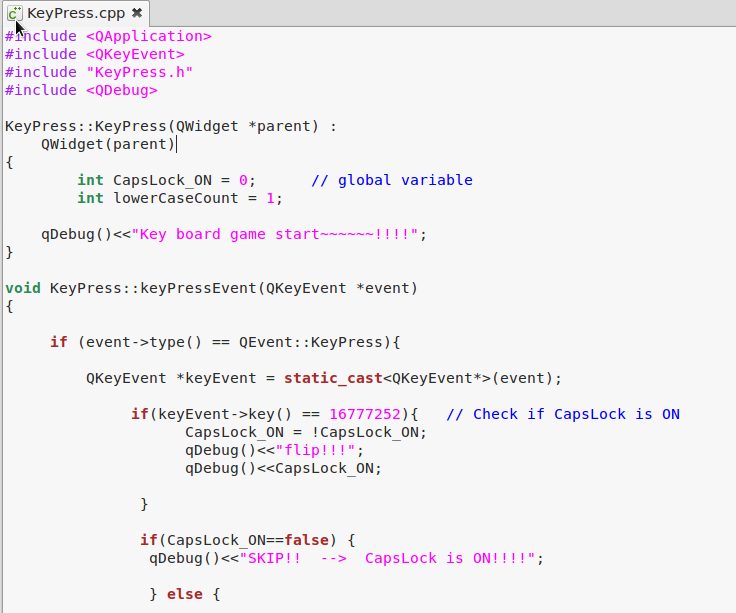


1. I make a system to record the lower case character input

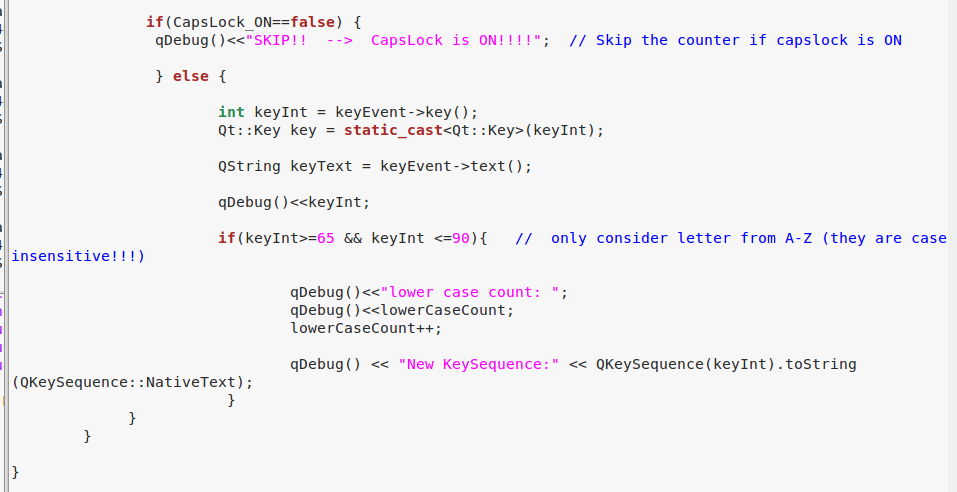


1. When the CapsLock is on, we skip the counter:



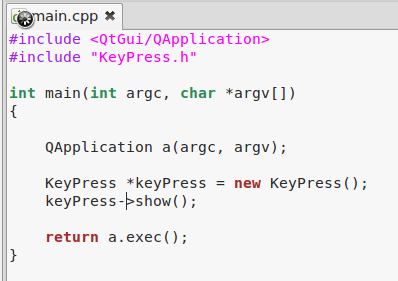


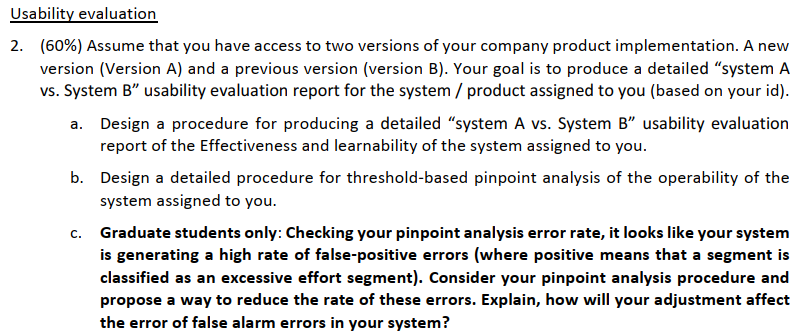
Here we use **keyEvent->key()** to check if CapsLock ON!!



Then later we use **keyInt>=65 && keyInt<=90** to check if there are in range.

1. Main.cpp





1. How to evaluation effectiveness?
2. Method one: considering using SUMI evaluation tool

* Define the functions we are going to test for: text areas, text boxes, radio button, checkboxes, drop-down radio buttons
* Check the valid input from the users
* Generate the correct input rate for system B. We will use it as a base rate B
* Then we test system A and get the rate A. Compare rate A and rate B

1. Method two:

* Define 20 **identical independent tasks**.
* Each task followed the same general workflow, but the function blocks, parameters, and properties being worked on were varied.
* Ask 20 users to complete the 20 tasks respectively, for system B. And generate an efficiency report including criteria such as time consumed, correct hotels and rooms booked, correct location with WiFi, correct City, and wrong booking rate. We use this report as “efficiency report B”.
* Generate “efficiency report A”, and compare it to “efficiency report B”.
* Compare the correctness rate, and time consumed.
* Or,
* We use another strategy: ask users to complete certain task within the fixed time (like 7 minutes).

**How to evaluation learnability? (below steps can also help to test operability)**

1. Here we need to evaluation the two online rental system.
2. We identify a set of “scenarios”:

* Find a room with WiFi and free breakfast in October in
* Find a student apartment where most of the residents are student
* Find an apartment close to university, available from August

1. We test for metric learnability
2. Define 20 iid tasks

* Find a room with price lower than 500 USD while with 3 other roommates
* Find a room with price between 600-800 USD while with 1 other roommates
* Find an apartment with 3 rooms which near round rock
* …
* Etc.

1. We hire 20 people (new to the system B and A) to take participant in the experiments
2. Those who complete all task within 1 hours will get two free movie tickets
3. The task procedure is to register the website, find the apartment, finish payment and screen capture the confirmation message
4. Record all the time needed (average time) to finish the tasks for system B. Plot the TOT chart
5. Try to find the curve and the power line
6. Also, we hire 5 people (expert to the system B) to finish all the task. Find the base effort.
7. **Find the gap between base effort line and the power line. The gap is the learnability gap**
8. Do it all over again for system A.
9. Now we have chart B and chart A. We can compare their own learnability gap.
10. Produce data analysis report with explanation of the learnability difference between two systems

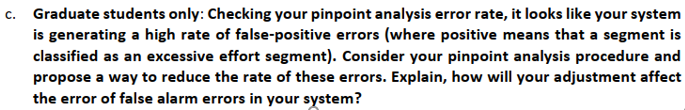
1. Analysis the operability of the system
2. We carry out the evaluation for learnability and operability (how much the users can utilize the system)
3. We identify (pinpoint) usability issues in a way that directs the developers to the actual interface code.
4. In pinpoint analysis we segment the task with **35 seconds uniform time** segments
5. Pattern Recognitions (classify interaction in time segments into several classes based on the level of

effort) EE, NE (excessive effort) identify Excessive effort segments

1. We ask the system expertise to find the metrics and do supervised classification
2. We choose threshold-based method, and **choose only one metrics:** Number of saccades
3. For example, one standard deviation above the average number of saccades (per segment) obtained in the training stage. Get the training subjects to perform the tasks, measure the number of

saccades per segment per task per user and average over segments

1. Label the segments where the result is above the threshold as E. Label the segments where the result is above the threshold as NE. (All segments are classified into Excessive or Non-excessive effort segment)
2. Ask Have an expert to evaluate the segments classified as E.
3. Get the segments that classified as E
4. We focus on the E
5. Solve the error rate problem:



1. Firstly, let we define the two type of errors: (according to the paper we learnt)

* Classifying NE segments as E segments is regarded as false positive or **type-I error**.
* Segments that show excessive effort per manual classification but are Identified excessive effort segment as non-excessive effort segments is regarded as false negative or **type-II error**.

1. Here we say **we get high type-I error**
2. All segments are classified into Excessive or Non-excessive effort segment, which applied threshold “Number of saccades” during the test. On the other hand, we also consider below threshold:

* Average fixation duration
* Number of fixation
* Average saccade amplitude
* Average eye path traversed

1. It is assumed that all the segments classified as Excessive Effort Segments are **due for an additional process of manual evaluation**.
2. If we get high type-I error, we can consider lower the threshold – the number of saccades. Or can we consider less strict threshold on each features when we do the experiments
3. In another words, lower the bar and we may get less false-positive errors.