Outline of paper on GUI Testing

Title page

Abstract (a summary of the paper, write this last)

1. Intro (explain the problem or topic and its importance, give background)

Describe the problem:
• GUI testing: the process of testing a product's graphical user interface to ensure it meets its written specifications.
• This is normally done through the use of a variety of test cases.
• Regression Testing. Ensure a change to the code does not introduce new errors. Usually have a set of test cases that are run for each major (or minor) code change.
• Problems for a GUI: how to specify/write the tests, and how to compare new/actual results to expected results

Briefly describe three methods:
• Manual Testing:
  specifications of test = sequence of steps for test user to take
  expected results: statement (or screenshot) of how GUI should look
  verification: user compares screen to expected results
• Capture playback:
  specifications of test = sequence of steps for test user to take (or script)
  expected results: or screenshot of how GUI should look
  verification: use simple tool to compare actual results screenshot to expected one
• GUI objects state (like unit testing, in code):
  specifications of test = sequence of calls to GUI object methods
  expected results: set of GUI object states, specified for certain objects
  verification: assertion methods in code compare actual objects to expected values

Layout the plan for the rest of the paper
• Explain the methods in more detail:
• Compare the methods, pros and cons
• Future work?
• Conclusion

Body
2. GUI Regression Testing Methods
(Present/explain the various approaches)

2.1 Manual Testing
2.2 Capture Playback
2.3 GUI Objects State
3. Comparison
   (can be organized in different ways:
    pros+cons of each method, OR
    compare how they write the test, then how they compare results,
     etc.)
   (may use a table for the results)

3.1. The test formats
3.2. The verification methods

4. Future directions/possible improvements
   Maybe some way of automatically generating much of the code in the Object State method

5. Conclusion (summarize your conclusions)

References