Kenneth Chuen and Anthony Krivonos March 30th, 2021

Prof. Gail Kaiser COMSE6156

Final Project Second Progress Report

Team Name: Gren Maju

Team Members: Kenneth Chuen (kc3334) and Anthony Krivonos (ak4483)

Changes

No changes have been made. See the paragraph below for the original response for **Novelty**.

Qubot will be one of the few, if not the only, fully autonomous exploratory testing framework published for Python. It will be able to crash test and simulate page flows for a wide variety of websites that do not require hyper-specific input generation nor difficult login mechanisms. Additionally, testers will have the options to specify their entire tests in a .qu JSON file, largely removing the need for coding knowledge.

Research Questions

Technical Questions:

- What is the average exploration rate (percentage of components visited) during a website test?
- What is the rate that Qubot performs successful website navigation to a target element?
- What is the rate of crashes vs. detections of crashes?
- What are the fastest, slowest, and mean rates of reward accrual by Qubot?
- How does Qubot compare against Selenium-AI in terms of test speed and exploration rate? (Assuming we are able to get Selenium-AI to run.)

User Study Questions:

- What is the average amount of time it takes for the test subjects to set up a test, after having read the instructions in the project's README?
- What are the most commonly used configuration parameters in the .qu.json config file?
- What are the difficulties and criticisms posed by the test subjects?

User Study Demographics:

- Three developers—Danielle B., Joey T., and Jungkyu K.—who are all mutual friends in Computer Engineering, Computer Science, and Electrical Engineering, respectively, who have various technical backgrounds and are undergraduate students in SEAS.

Challenges

Some of the initial challenges we faced, but were able to overcome in the past week, were the development of a configuration scheme to make Qubot work in the command line and the publishing of Qubot as a Pip package. That being said, we still expect to face some challenges in the coming weeks. Primarily, we are unsure how smoothly testing 10 different websites will go, and it is unclear if Selenium-AI will be cooperative to be used as a point of comparison across these 10 tests. We are moreover concerned about being able to find a couple hours in the aforementioned test subjects' schedules to have them undergo our user study, but are hopeful that they will be available.

Demo

We will start by giving a one minute elevator pitch on the inspiration behind Qubot, what Qubot is, and Qubot's technical stack. We then hope to mitigate the risk of the tool not cooperating during a live demo by playing a three-to-four minute pre-recorded sample test and narrating this demo live during class time.

Everything below has been copied from the first progress report, for completeness.

Value to User Community

Qubot will be most valuable to small developer teams working on websites that do not have the resources to hire QA testers, as well as QA testers in larger companies who do not have a lot of technical expertise, but are tasked with exploratorily testing large portions of code. Each of these user demographics will be able to perform tests and accumulate various analytics on the software they are testing, and Qubot may be used as both a command-line tool and a Python library.

Dataset

Qubot learns on the fly by scraping interactive HTML elements off websites and converting user interactions into a UITree. Then, it uses the UITree that it just learned to navigate to the right HTML element or webpage. Hence, it will not be using any external datasets. That being said, the ground truth we will be comparing Qubot against is manual interaction with the website under test. For instance, if it takes 3 button clicks to get from Page A to Page B on a website, we expect Qubot to be able to learn to do this consistently within a reasonable amount of epochs. If it fails to achieve such a goal, then we will mark the test as having failed.

Comparison Subjects

Given the meager amount of source code for Q-Learning testing agents that is currently available online, we will be unable to compare Qubot to techniques written in papers such as DRIFT and Fastbot. That being said, we found a project on GitHub, called Selenium-AI, that we will try to make some comparisons against during our experiments. This work will simply be obtained by pulling the project from GitHub and spinning up a Docker instance with the image from this repository.

Project Delivery

All source code and documentation will be available on this public GitHub repository. The link to the Qubot PyPi project will be posted in the README.md file when it is published.

Open Source Software Used

OpenAI Gym: used for custom RL agent environment

Selenium Webdriver: used for webscraping