University of Toronto

CSCD01 Phase 1 Report
Team: Infinity

Reverse Engineering

Andrew Berneshawi	`bernesha` 998425388
Nicholas Dujay	`dujaynic` 999194900
Pirave Eahalaivan	`eahalaiv` 998152136
Josh Hillen	`hillenjo` 999143911
Yunqin Huang	`huangy11` 996445217
Ho-Cheung Lai	`laihoche` 999035020
Wei Li	`liwei18` 999232753
Choi Tak	`choitak` 997905566

February 5, 2014

Abstract

The objective of this report is to reverse engineer a set of models from the `matplotlib` source code to explain its design. The goal is to use UML to highlight the structure and behaviour of the code. The report strategically determines which aspects of the design to model, how much to abstract away from the code base, and which parts of UML to use. The models are designed to explain the most interesting and important aspects of the design.
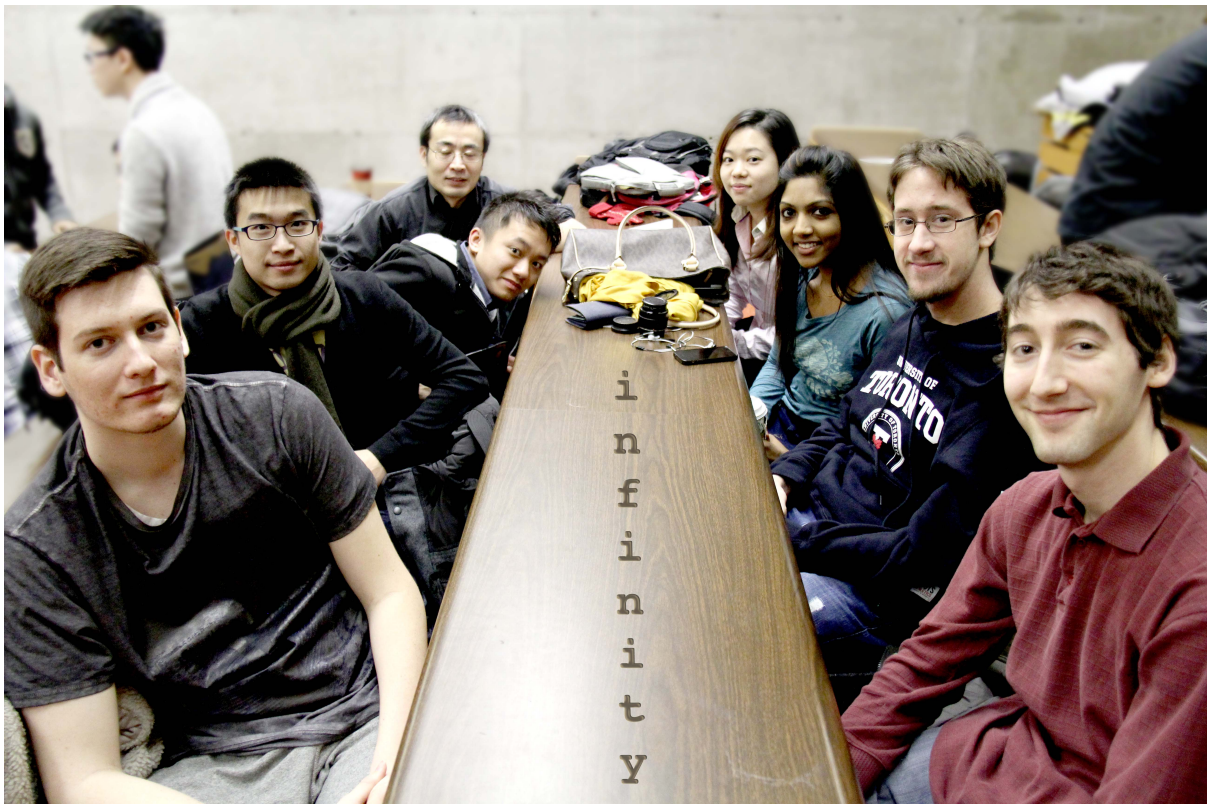
Contents

List of Figures

Introduction

Team

We are team ∞.

Members

Andrew Berneshawi is currently in the 3rd year of his Computer Science Specialist program in the Software Engineering stream. Thanks to his involvement in the co-op program he already has over a year of industry experience in web development. In his first internship at UTSC's IITS department he worked with PHP, followed by Java at IBM, and Java again in his latest job at Amazon; where he also gained experience with agile development. Now that he is back in class he looks forward to strengthening other parts of his software development knowledge. Not just passionate about software, Andrew also enjoys staying up to date on the latest computer hardware, and has built a few computers for his friends and himself. Outside of computer science he is interested in the sciences in general, and although the course load has been prohibitive regarding taking any challenging physics or chemistry courses, he did manage to squeeze in some psychology courses which he found very interesting. He is looking forward to using his experience from previous courses and internships to contribute to the project.

Nicholas Dujay is a 3rd year undergraduate computer science student at the University of Toronto Scarborough Campus. He transferred from Carleton University, where he was studying to become an electrical engineer, but he decided to pursue his passion in computers and study at the University of Toronto. He is interested in learning agile, extreme programming and the software development process so he can have an idea of how software development happens in a real business setting. He is hoping to enter the video game industry after graduation, so this class will help him have an idea of how to work in a team. He spent a couple months developing a game during the summer of 2013, but had to stop development because of school. He has travelled to many places in North America, including Montreal, Moncton, New York, and Canc?n. In his spare time, he enjoys watching television shows such as Breaking Bad.
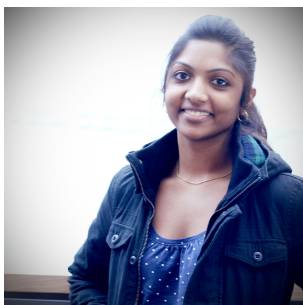
Pirave Eahalaivan is a 4th year student at the University of Toronto, specializing in computer science and minoring in psychology. As a part of the co-op program she has worked as a programmer analyst at Scotiabank and Ontario Teachers'Pension Plan. Her work experience has allowed her to explore web development via ASP.NET. These experiences have also enabled her to work in an agile developing environment. As the web associate for UofT's Co-op Student Association, she is responsible for designing, updating, and maintaining the club's website, using webpress. Having taken Programming on the Web (CSC309) and Databases (CSCC43), she has experience working front and back end with javascript, PHP, and MySQL. She would like to expand her horizons in web developing as well as graphic designing.

She hopes to pursue a graduate degree in applied computing before she begins her career. Some of Pirave's hobbies include photography, reading and fitness.


Joshua Hillen is currently a third year student of computer science at the University of Toronto Scarborough campus. He is specializing in software engineering and plans to join the industry in 2015 directly after he graduates with an Honours Bachelor of Science. Previously to Joshua's current term he has learned and mastered a few languages, most importantly Java and C. He also participated in a group project where he was introduced to agile development and constructed software that modified .csv files for excel spreadsheets. Joshua has also taken on a variety of solo programming projects including an implementation of the Tiling Algorithm and the game Battleship. In his first two years Joshua's studies have primarily required critical thinking and time management; two important skills for group based work. He has broadened his perspective through electives including philosophy, psychology and astronomy. Joshua is a well rounded individual who has traveled overseas to Prague, Vienna, Budapest and South Africa.

Yunqin Huang is a 4th year student of computer science in University of Toronto at Scarborough. He was born in China, and he spent his childhood there before he moved to Canada. His first language is Mandarin, and English is his second language, but he is proficient in both. He is not a very talkative person, and loves reading. He is experienced in C, Java, Python, and shell script, and quite familiar with UNIX platform. He has worked as a developer of applications on Solaris and Linux system in a network security company, and a developer of java EE web application in Medical service company. He loves computer games and movies as well.

Derek Lai was born in Hong Kong and came to Canada at the young age of 2. Ever since he was a kid, he has always wanted a job revolving around technology. Derek attended Unionville High School where he was first introduced to programming in Grade 9, learning his first programing language; Turing. From there, he furthered his computer science knowledge, learning both Visual Basic and Java. Derek is also the former President of his high school?s Robotics Team. Derek Lai is now currently a student at the University of Toronto working towards becoming a Software engineer. If not at school, he can usually be found in front of his computer or out with friends chatting over a cup of coffee.
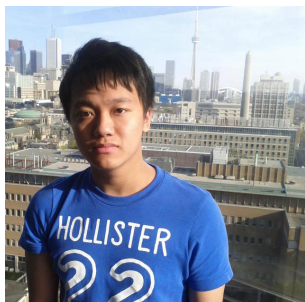
Wei Hang was born in ZhanJiang, China December 27,1993. Wei Hang is prefer to be called Dave. Dave moved to Toronto, Canada at age of 9, studied and graduated from local elementary, and high school. He's a third year student studying at University of Toronto, Scarborough, estimated graduation date: 2014. In his spare time, he likes to play computer games such as League of Legend, Defence of the Ancient, Candy Crush, and Star Craft. Aside from his passion for video games, he also likes to study for fun. He loves to help people around him, sometimes he helps classmates with homework questions, and sometimes he volunteers to be a class note taker. Dave is planning to continue his graduate studies at University of Toronto st.George perusing degree in Master of Science. Dave is very passionate about programming and food, he would just sit on a chair and program for hours with a bag chips on the table, which explains why he's such a fatty!

Candy(Choi) Tak Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
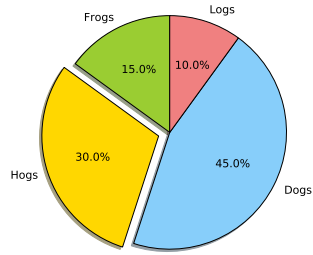
matplotlib

matplotlib is a plotting library that produces quality graphics for 2D(full support) and 3D(limited support) diagrams in Python. The library is most popular in the community for Python-based scientific computing and can be applied in many different applications. matlplotlib comes with an interactive shell that can generate figures via simple python scripts, this includes the iPython shell embded in MATLAB, IDL, or Mathematica.
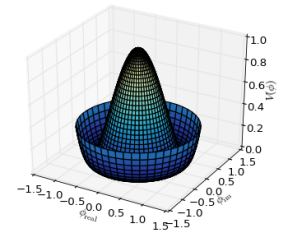
In addition, matplotlib is cross-platform compatible and can be embedded within web application servers and via any of the six GUI interface toolkits within different environments. These produce quality graphics in hardcopy that may be used for publication on the web or on paper. Supported formats include raster based graphics (PNG) and vector based graphics (PDF, PS, SVG). matplotlib can be used by a large range of users, from beginners who can use simple python commands, to power users who have
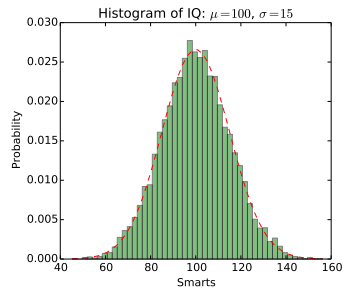
(a) Histogram example   (b) Pie chart example   (c) mplot3d toolkit example

Figure 1.1: Examples of matplotlib graphics

full access to styling and graphical properties. A few examples of figures produced by matplotlib are illustrated in Figure 1.1.

The Process

Reverse Engineering Tools

Our group decided to find a suitable reverse engineering tool to aid us in generating a UML class diagram that represents all the classes, subclass relationships, and associations in the source code. We initially took a look at various applications that would provide us with the nessecary tools to generate diagrams from the matplotlib source code. Some of these applications included visual paradigm, pywebuml, pyreverse, etc. After listing and analyzing some of the pros and cons of each application, we unanimously agreed on using a software called Enterprise Archeticture by Sparx Systems (Sparx).

Many of the applications offered us a way to input our matplotlib source code and return a UML class diagram representing all the relationships within it. However since we still needed to edit the generated model to capture information missed by the tool, and remove unnecessary details, we found Sparx provided us a way to do this with ease.

Abstraction Steps

After generating our intial model from the raw source code, we all took a step back to understand the relationships and associations between different components. After briefly inspecting the overall model in totality, we strategically divided it up into parts for each team member to modify. For example,

System Architecture

Chapter

Section

Subsection

Subsubsection

Figure 4.1: Default Img

Paragraph blah blah blah