Usability Testing and Advanced GUI Testing with Tools

Organized by

The Hong Kong Polytechnic University
香港理工大學

Department of Computing
電子計算學系

HKSPIN

PMSG

QA

Supported by

OPEN SOURCE HONG KONG

Internet Society
Hong Kong 香港互聯網協會

Funded by SME Development Fund

工業貿易署
Trade and Industry Department

Any opinions, findings, conclusions or recommendations expressed in this material/event (or by members of the Project team) do not reflect the views of the Government of the Hong Kong Special Administrative Region, Trade and Industry Department or the Vetting Committee for the SME Development Fund and the Dedicated Fund on Branding, Upgrading and Domestic Sales (Organisation Support Programme).

Topics

9:30AM-12:30PM

A/B Testing: Apptimize

Usability Test: Lookback

Break

Card Sorting Test: XSort

GUI Testing:Introduction to Sikuli

2:00PM – 5:00PM

Sikuli: Advanced Techniques

Break

Sikuli: Batch Execution and OCR

Hardware Requirements

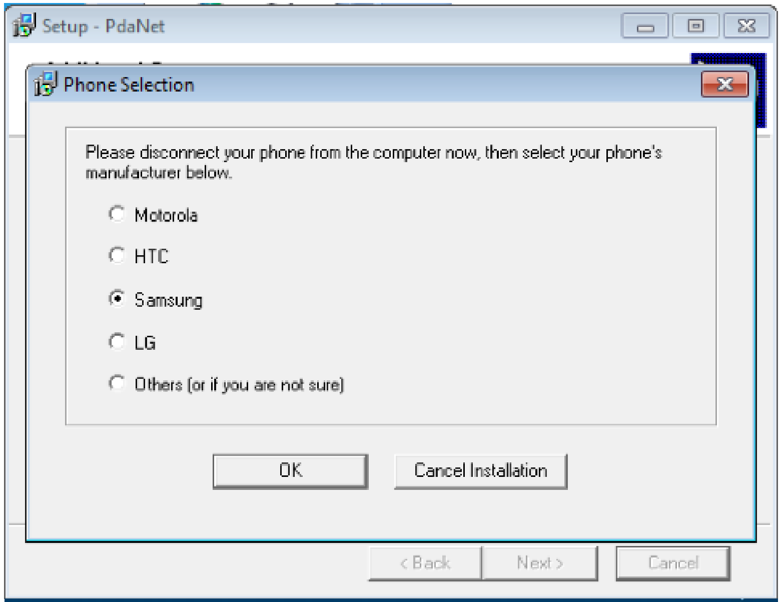
- The participants should have
  - A real android devices
  - A USB cable used to connect to Windows 7

Lab Environment

- Windows 7 (64 bit , 16 GB RAM)
- Select Option [3]SME workshop in the boot menu when you start the computer
- Wifi:
  - Hotspot: SME_workshop or SME_workshop_5G
  - Password: $mew0rk$h0p
- All the files used in the workshop are located:
  - Local Disk D:/mobaptest_files/course_files/Session_2
  - Application APKs: mobaptest_files/course_files/apk
  - Answers to the exercise: mobaptest_files/course_files /Session_2/ANS
  - Passwords to the answers will be released **10 mins after the start** of the exercises

Lab Environment

- Install PdaNet+ on your PC
  - Please install PdaNet+ on your PC for file transfer to your device
  - File available on course_files folder on Desktop
  - Select the brand of your device when installing
- Install PdaNet+ on your Android device


Setup - PdaNet

Phone Selection

Please disconnect your phone from the computer now, then select your phone's manufacturer below.

○ Motorola
○ HTC
● Samsung
○ LG
○ Others (or if you are not sure)

PdaNet+

PdaNet+
June Fabrics Technology Inc.

Table of contents

- Usability Testing
  - A/B testing: Apptimize
  - Usability testing: Lookback
  - Card torting test: Xsort
- GUI Testing
  - GUI testing: Sikuli introduction
  - GUI testing: Sikuli advanced techniques
  - GUI testing: Sikuli batch & OCR

1 A/B Testing: Apptimize

Learning Objective #1

- Improve the efficiency of A/B testing with Apptimize
- Set up the Apptimize for instant A/B testing
- Prepare the application for Apptimize
- Understand the difference between paid and free version

A Brief History of Apptimize

- Founded in 2013
- Code-based A/B Testing solution
- Provide robust A/B testing service without reinstalling the application
- Provide (almost) instant updates to the application on user’s devices

What to test

- The usability of the mobile application user interface
- The turnover rate of a certain button
- The effectiveness of the user interface as a whole

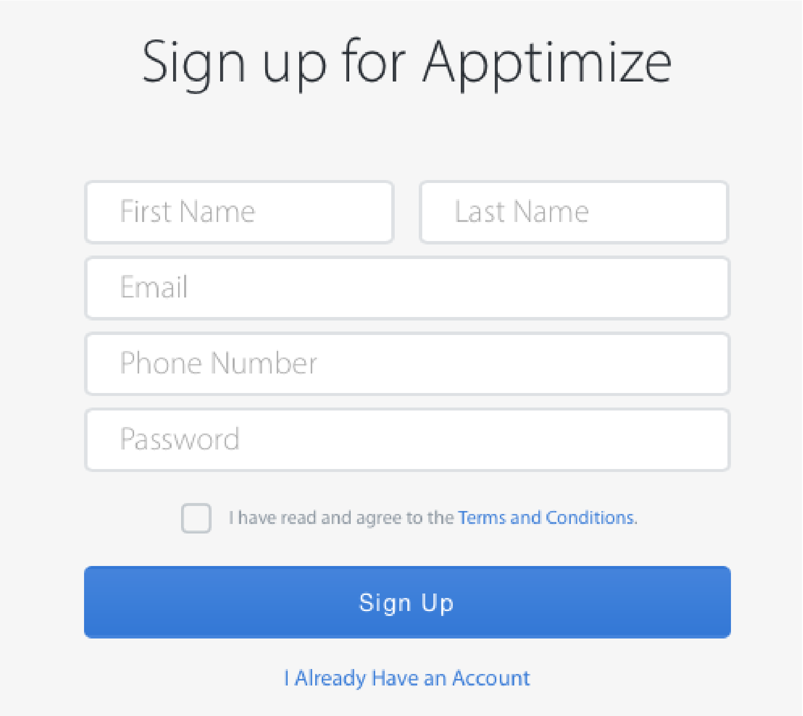
Installation: Apptimize

- Register an Apptimize account:
  - apptimize.com
- Prerequisite:
  - Android Studio

- Download the Apptimize Android SDK
- Open the folder containing apptimize-android-x.jar and copy the file

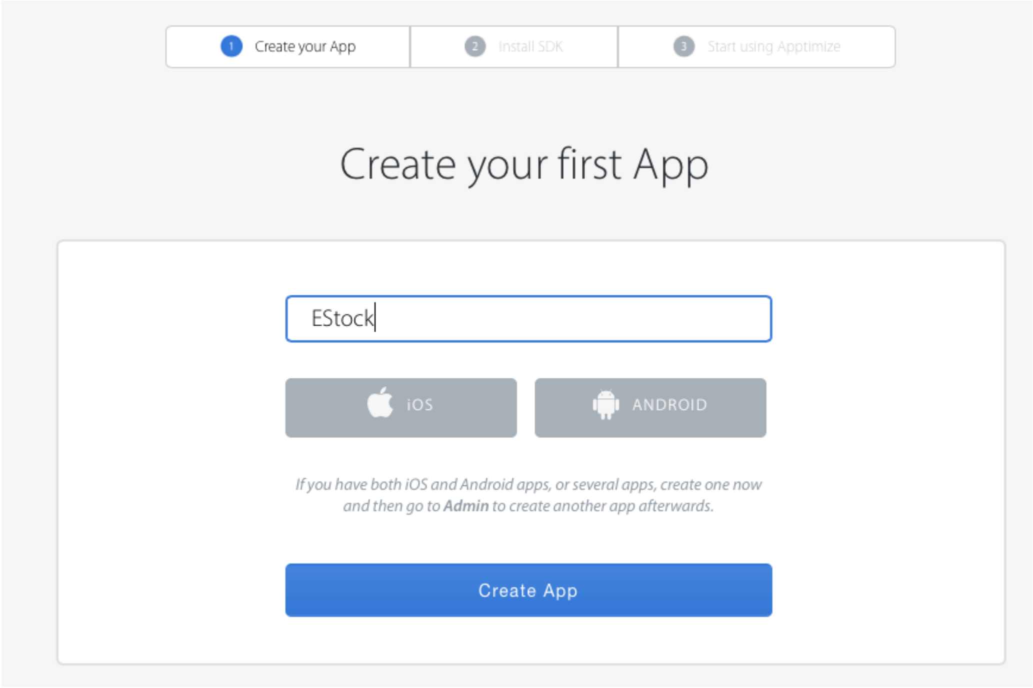
Installation: Apptimize

- Registration of a Apptimize account:
  - apptimize.com



Sign up for Apptimize

☐ I have read and agree to the Terms and Conditions.



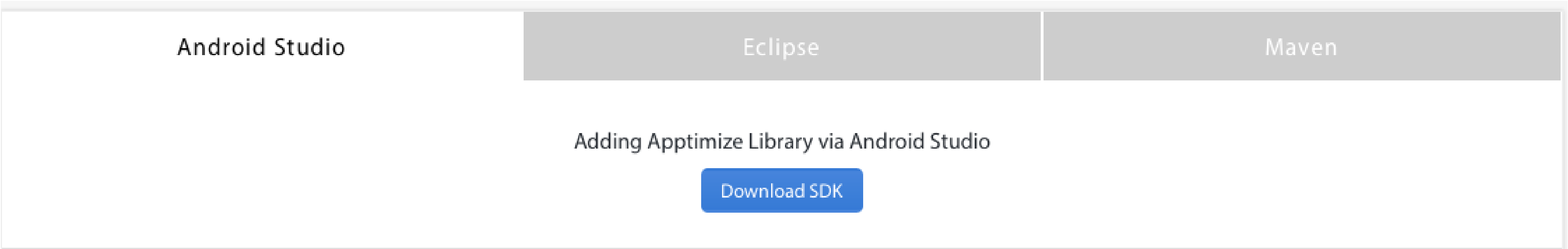
Create your first App

EStock

*If you have both iOS and Android apps, or several apps, create one now and then go to **Admin** to create another app afterwards.*

Installation: Apptimize

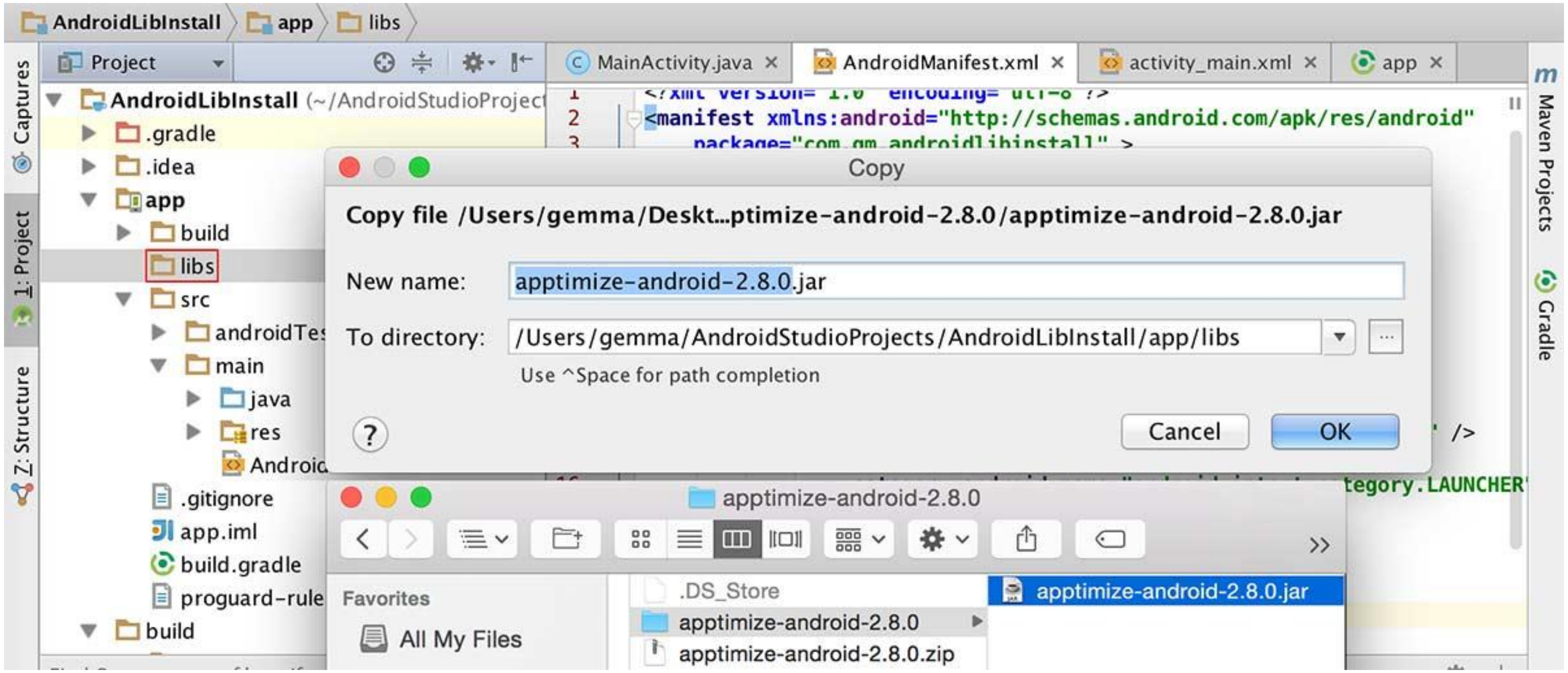
- Open Android studio
  - Android Studio is already installed on your PC
  - Open Android Studio and open the EStock project
  - The app project is in the course_files folder
- Download the Apptimize SDK



Adding Apptimize Library via Android Studio

Installation: Apptimize

- Go to Android Studio, right click on the libs folder (app -> libs), select paste



Copy

Copy file /Users/gemma/Deskt...ptimize-android-2.8.0/apptimize-android-2.8.0.jar

New name: apptimize-android-2.8.0.jar

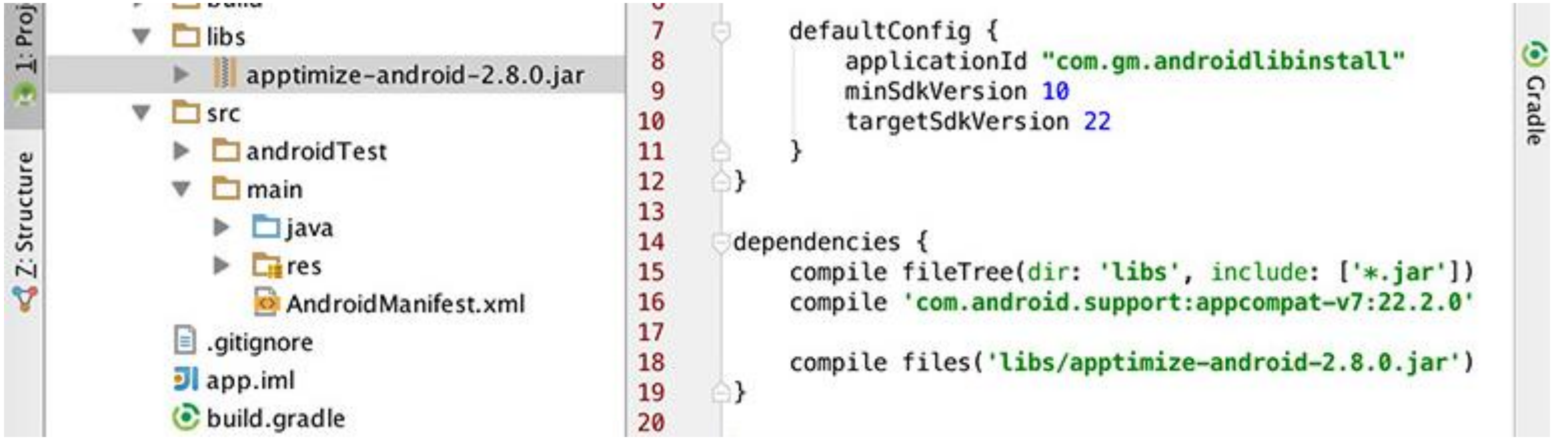
To directory: /Users/gemma/AndroidStudioProjects/AndroidLibInstall/app/libs

Use ^Space for path completion

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.gm.androidlibinstall" >
```

Installation: Apptimize

- The **apptmize-android-x.jar** should appear in the libs folder
- Right click on **apptmize-android-x.jar** in Andorid studio, select Add as Library
- Check the **build.gradle** that the apptimize library is present under dependencies



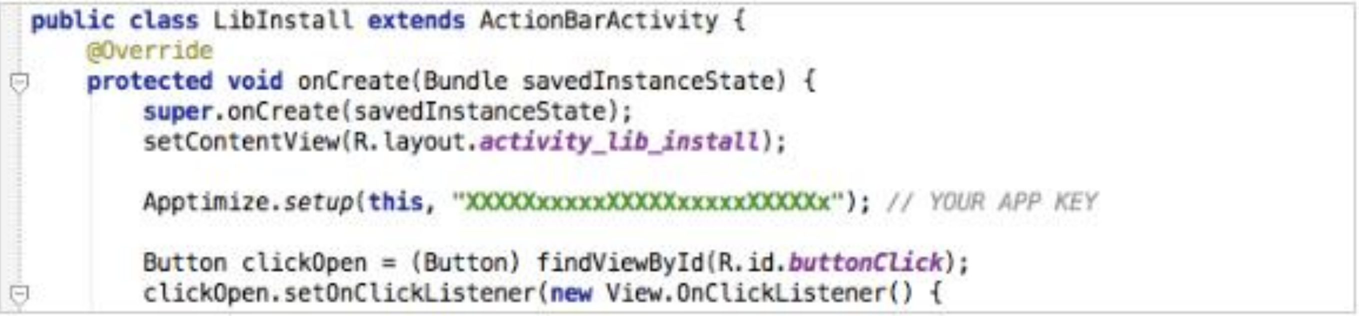
```
    defaultConfig {
        applicationId "com.gm.androidlibinstall"
        minSdkVersion 10
        targetSdkVersion 22
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.2.0'

    compile files('libs/apptimize-android-2.8.0.jar')
}
```

Installation: Apptimize

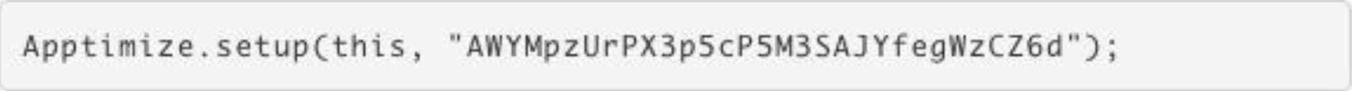
- Prepare the application for Apptimize experiment
  - Add these two statements in all the file (e.g. all the java files) which call Apptimize:
    - import com.apptimize.Apptimize;
    - import com.apptimize.ApptimizeTest;
  - Add setup() method call to all the activities' onCreate() method:
    - See next slide
    - Apptimize.setup(this, "xxxxxxxxxxxxxxxxx");
    - Replace xxxxxxxxxxxxxxxxx with your AppKey
  - Check the permission setting in AndroidManifest.xml
    - Uses-permission android:name="android.permission.INTERNET" />

Installation: Apptimize

- Prepare the application for Apptimize experiment
  - Add setup() method call to all the activities’ onCreate() method:
    - Replace xxxxxxxxxxxxxxxxx with your App Key
    - Your App Key can be found on the dashboard when you sign up to Apptimize



```
public class LibInstall extends ActionBarActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lib_install);

        Apptimize.setup(this, "XXXXXxxxxxXXXXXxxxxxXXXXXx"); // YOUR APP KEY

        Button clickOpen = (Button) findViewById(R.id.buttonClick);
        clickOpen.setOnClickListener(new View.OnClickListener() {
```



```
Apptimize.setup(this, "AWYMpzUrPX3p5cP5M3SAJYfegWzCZ6d");
```

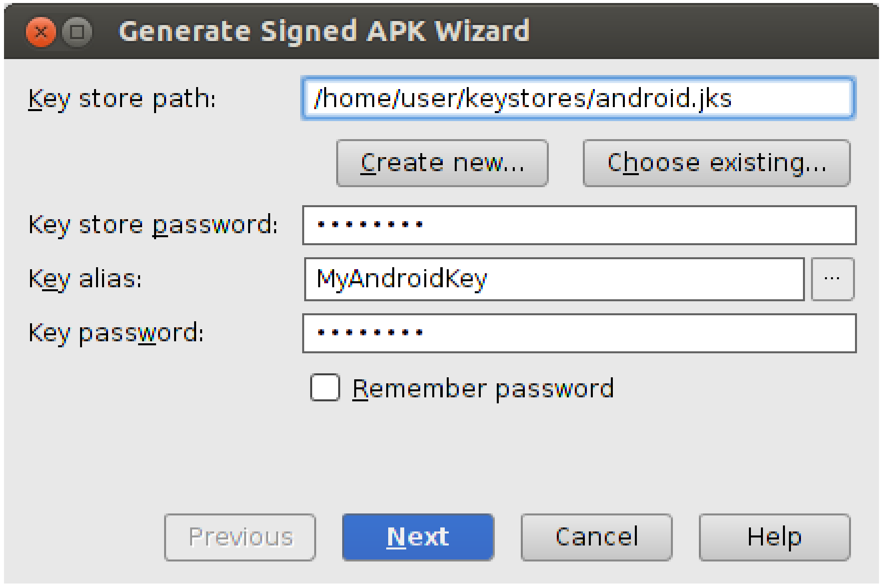
Generating workable build

- A signed debug build is needed for Apptimize to operate
- Generate the signed debug build following the steps:
  1. Set a new Run/Debug Configuration
  2. Set the app to be signed automatically in debug mode

     https://developer.android.com/tools/publishing/app-signing.html

Generating workable build

- Generate the signed debug build:
  - Go to Build > Generate signed APK
  - Click create new to make a new key store



Generate Signed APK Wizard

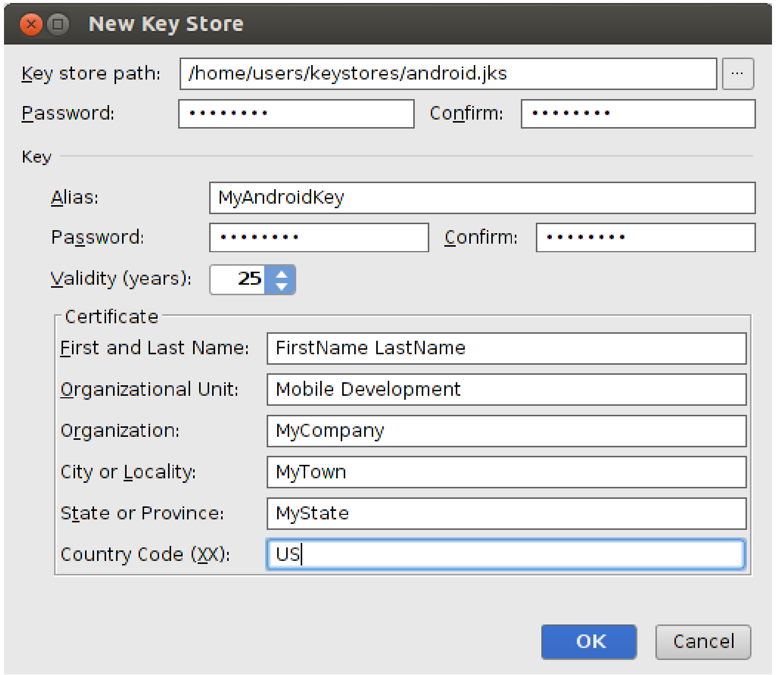
Key store path: /home/user/keystores/android.jks

Key store password: •••••••••

Key alias: MyAndroidKey

Key password: •••••••••

☐ Remember password



New Key Store

Key store path: /home/users/keystores/android.jks

Password: ••••••••• Confirm: •••••••••

Key

Alias: MyAndroidKey

Password: ••••••••• Confirm: •••••••••

Validity (years): 25

Certificate

First and Last Name: FirstName LastName

Organizational Unit: Mobile Development

Organization: MyCompany

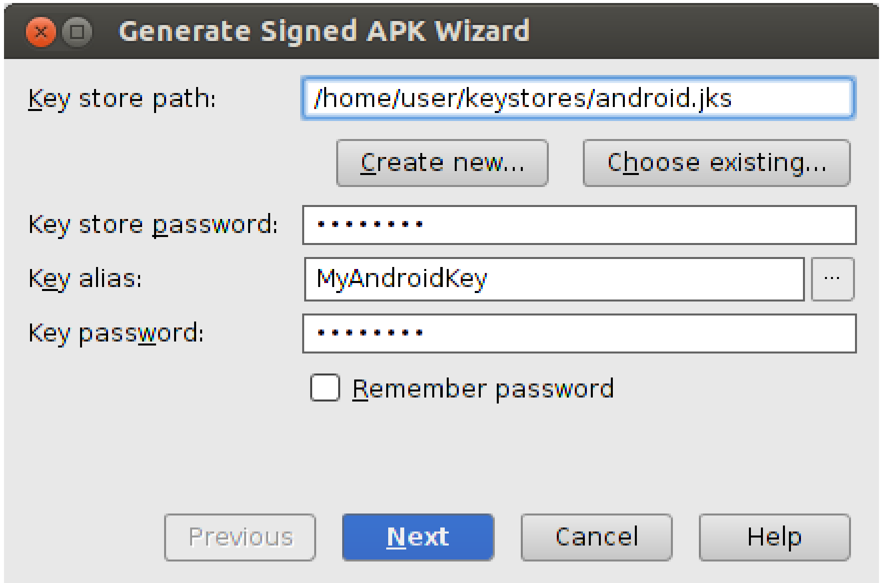
City or Locality: MyTown

State or Province: MyState

Country Code (XX): US

Generating workable build

- Generate the signed debug build:
  - After create the key store, go back and enter the password
  - Finally specific the destination folder, select debug for build type and finish



Generate Signed APK Wizard

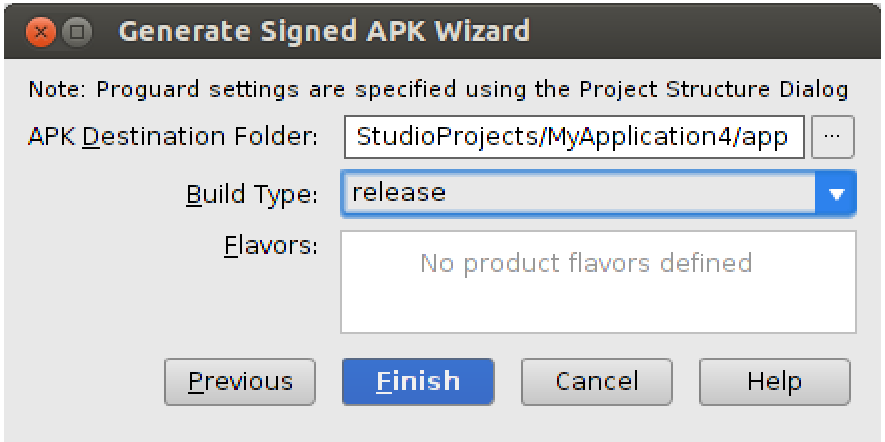
Key store path: /home/user/keystores/android.jks

Key store password: ••••••••

Key alias: MyAndroidKey

Key password: ••••••••

☐ Remember password



Generate Signed APK Wizard

Note: Proguard settings are specified using the Project Structure Dialog

APK Destination Folder: StudioProjects/MyApplication4/app

Build Type: release

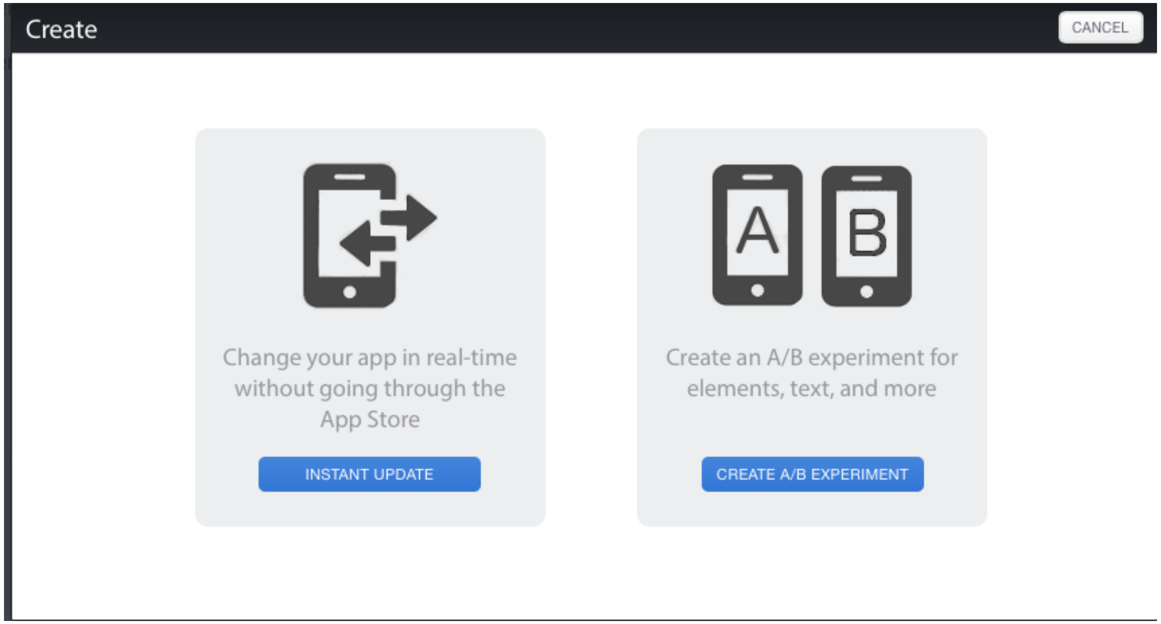
Flavors: No product flavors defined

Starting Apptimize

- Copy the APK generated by Android Studio to your mobile device
- Install the application on the mobile device
- Make sure you have internet connection
- Access the Apptimize dashboard

Creating Apptimize experiment

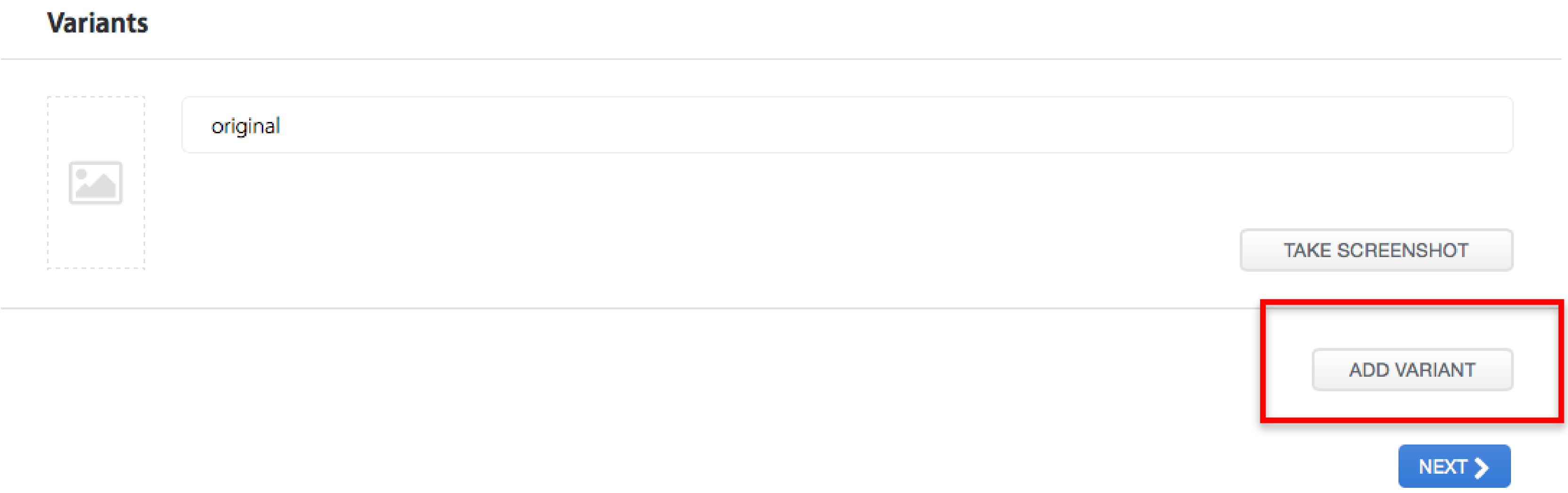
- Click the Create button on the Home page of Apptimize dashboard
- Select Create A/B Experiment



Create
CANCEL
Change your app in real-time without going through the App Store
INSTANT UPDATE
A
B
Create an A/B experiment for elements, text, and more
CREATE A/B EXPERIMENT

Creating Apptimize experiment

- In the variants section, you can click the Add Variant button to create a different version of application



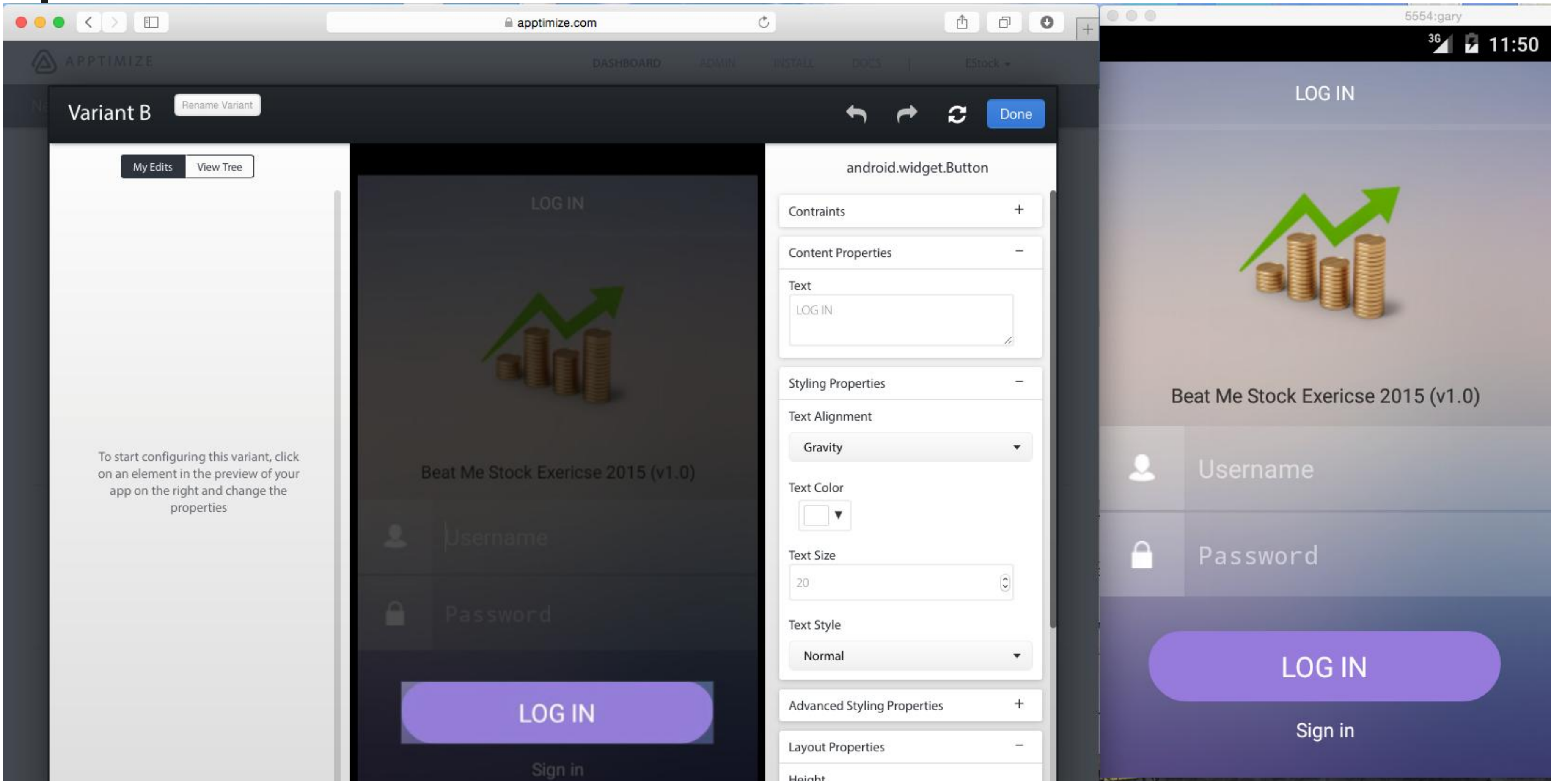
Variants

original

Creating Apptimize experiment

- If you click on the Add Variant button, you should see a mirrored image of your mobile app (see next slide)
- Make sure you already turn on the app on your mobile device
- Change the appearance of the application and save the changes
- Click Next

Creating Apptimize experiment



To start configuring this variant, click on an element in the preview of your app on the right and change the properties

Text: LOG IN

Text Alignment: Gravity

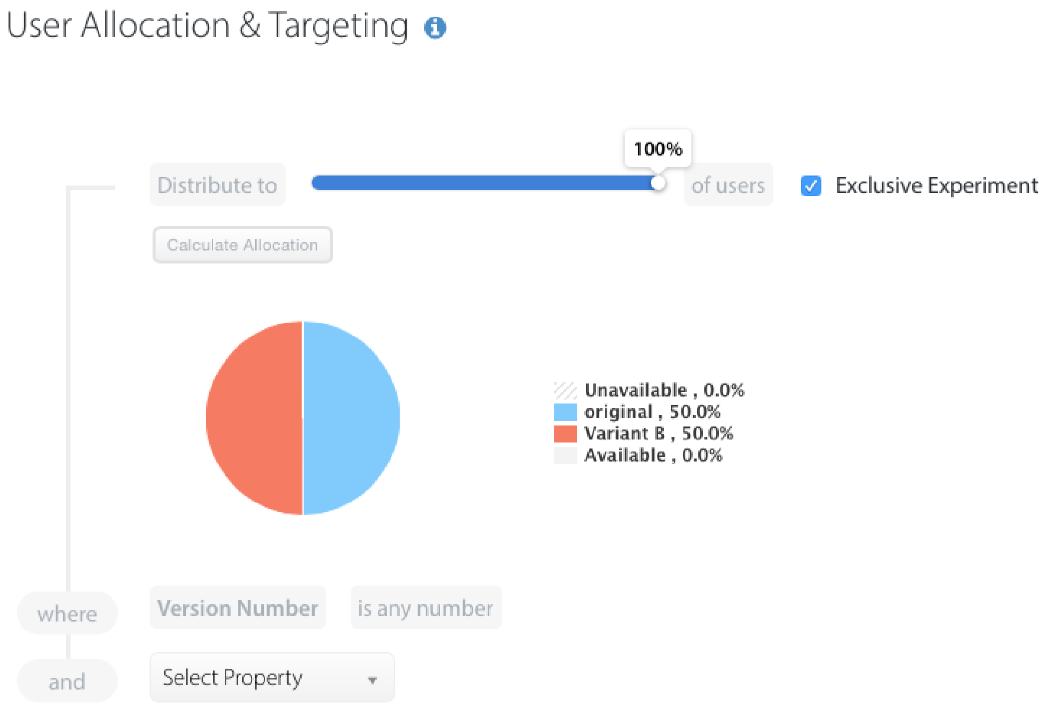
Text Color: □

Text Size: 20

Text Style: Normal

Creating Apptimize experiment

- In the User Allocation & targeting page, leave the setting as default



User Allocation & Targeting

Distribute to 100% of users ☑ Exclusive Experiment

▨ Unavailable , 0.0%
🟦 original , 50.0%
🟥 Variant B , 50.0%
⬜ Available , 0.0%

where Version Number is any number

and Select Property ▾

Creating Apptimize experiment

- The experiment will distribute to 100% of users – meaning if two versions are created, 50% of users will see version A and 50% of users will see version B
- Click Save and Next
- Run the experiment in the Launch page

Free version vs Paid version

- Apptimize is a paid service
- 4 plans available:
  - Starter, Standard, Professional, Enterprise
- First three plans provide 30 days free trial
- Starter only provide Instant Visual changes update features
- Standard provides event tracking statistics and A/B testing suite
- Professional provides Device Targeting, User Targeting in addition to the standard version

Connection and version

- No limited number of created versions
- Allow you to allocate a percentage of users that use a particular version
- E.g. 4 versions of the application, every 25% of users use one version
- One build, four version, no re-installation needed

Exercise#1 (10 mins)

- Form a group of two
- Prepare the application for Apptimize
- Set up the experiment on Apptimize dashboard
- Create two different versions of the application
- Start the experiment with a 50/50 ratio
- Use real device and also emulators

2 Usability Testing: Lookback

Learning Objective #2

- Use Lookback to record user reactions and user experience of the application being tested

Watch Users Work

- Too frequently, companies basing their designs on user input obtain through misguided methods.
- If the users have not actually tried to use the designs, they'll base their comments on surface features.
- Such input often contrasts strongly with feedback based on real use.

A Brief History of Lookback

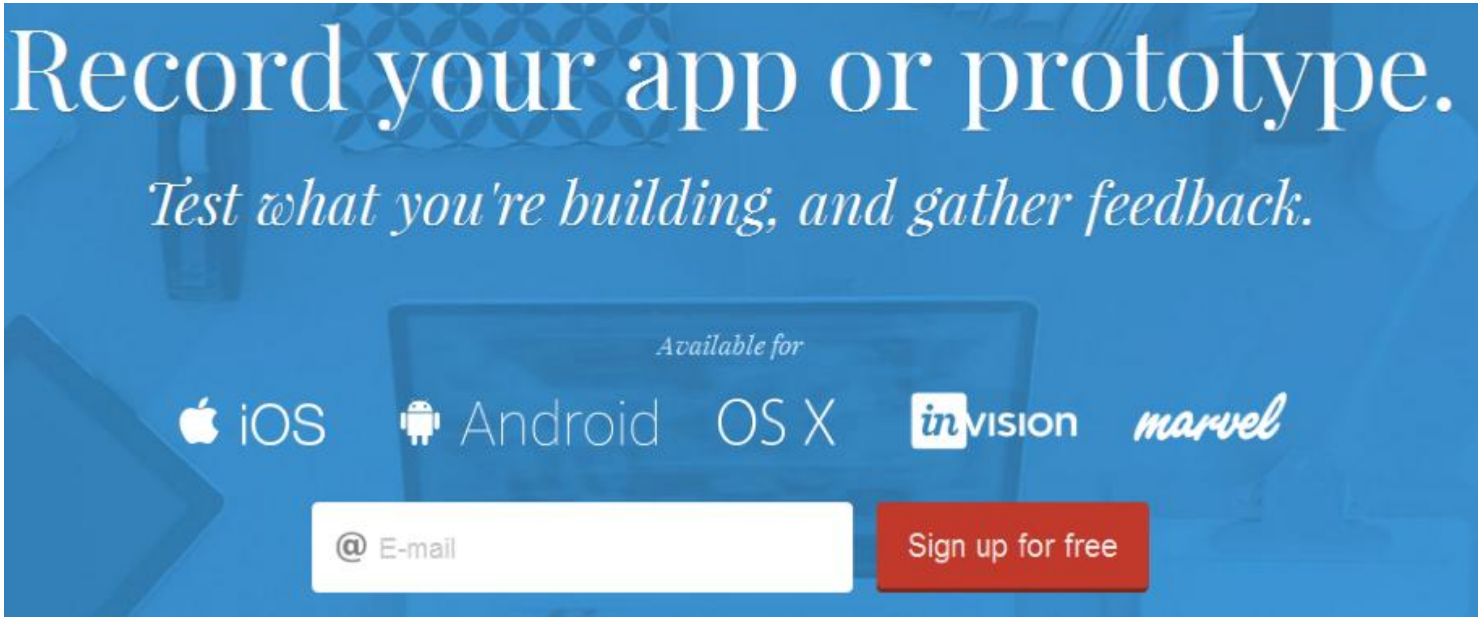
- Nevyn Bengtsson, former iOS lead developer at Spotify, found difficult to see how apps were being used on mobile devices
- No simple way of recording the screen of iOS devices at the time
- Lookback is the solution to the problem
- Available now as an open beta

What to test

- How the target users use the application
- The reactions of the users when they use the application
- Get the user feedback directly from the recording:
  - Audio is also recorded, so user can talk about their comment on the application directly

Installation: Lookback

- Register a Lookback account
  - https://lookback.io



Record your app or prototype.

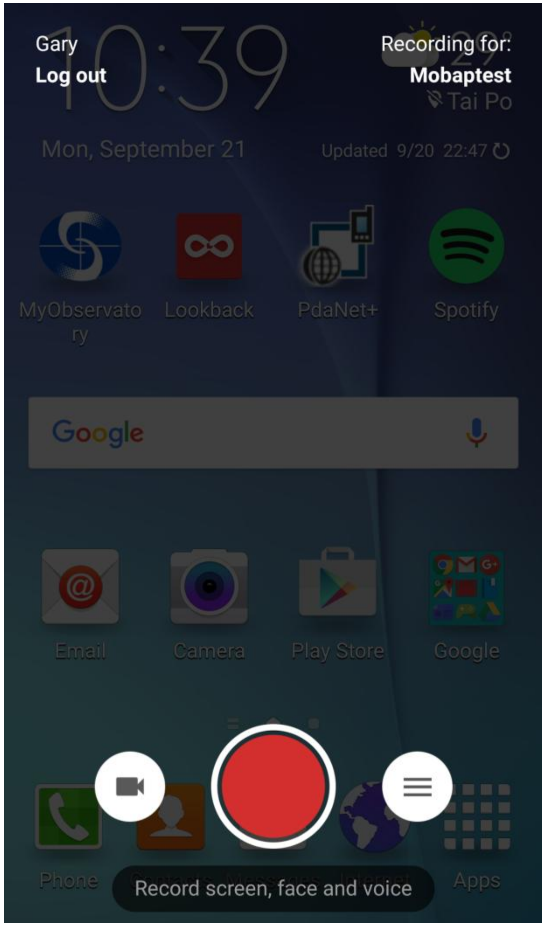
Test what you're building, and gather feedback.

Installation: Lookback

- Using your mobile application, visit Play Store and search for Lookback
- Download and install Lookback
- Open Lookback and connect it to your account

Starting: Lookback

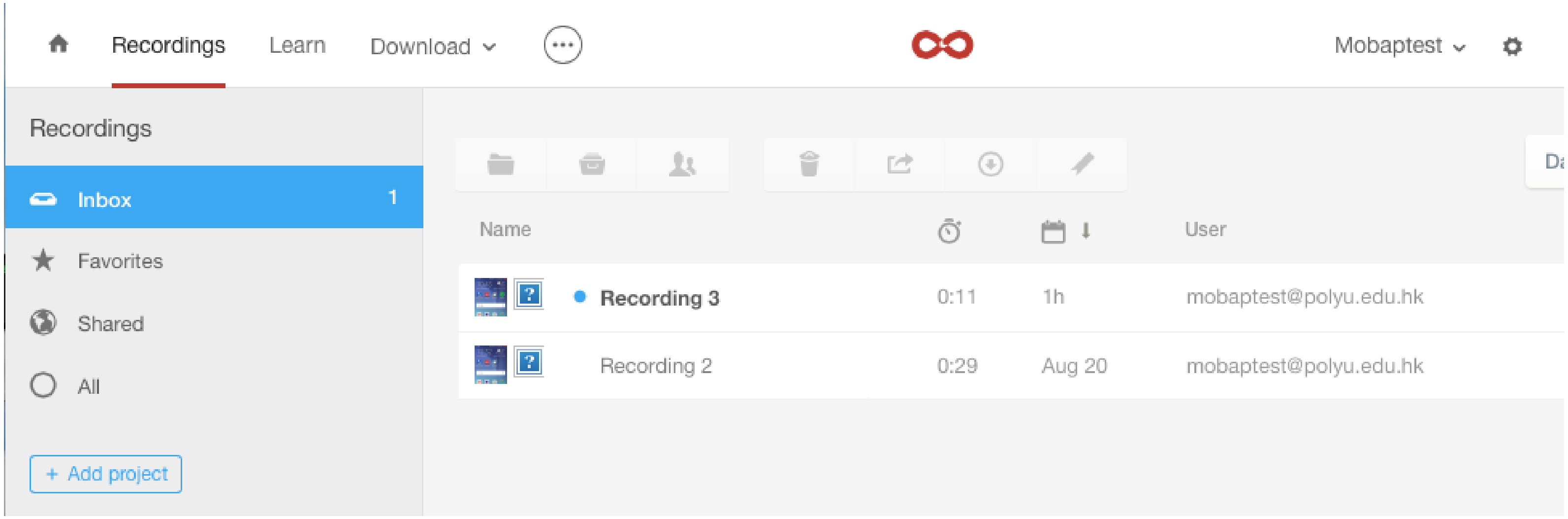
- Open Lookback in your android device
- Press the Red record button to start recording
- Finish the recording and press the upload button to upload the footage to the Lookback server



Record screen, face and voice

Starting: Lookback

- It takes a while for the footage to upload
- Once the uploading is finished, you should be able to see it on the dashboard



Name	⏱	📅 ↓	User
● Recording 3	0:11	1h	mobaptest@polyu.edu.hk
Recording 2	0:29	Aug 20	mobaptest@polyu.edu.hk

Exercise #2 (10 mins)

- Start using the application and record using Lookback
- Upload the recording after the test
- Try to watch the recording on the Lookback dashboard after the uploading is finished

3 Card Sorting Test:XSort

Learning objective #3

- Use Xsort to find out the best configuration of the navigation menu of your application

A Brief History of XSort

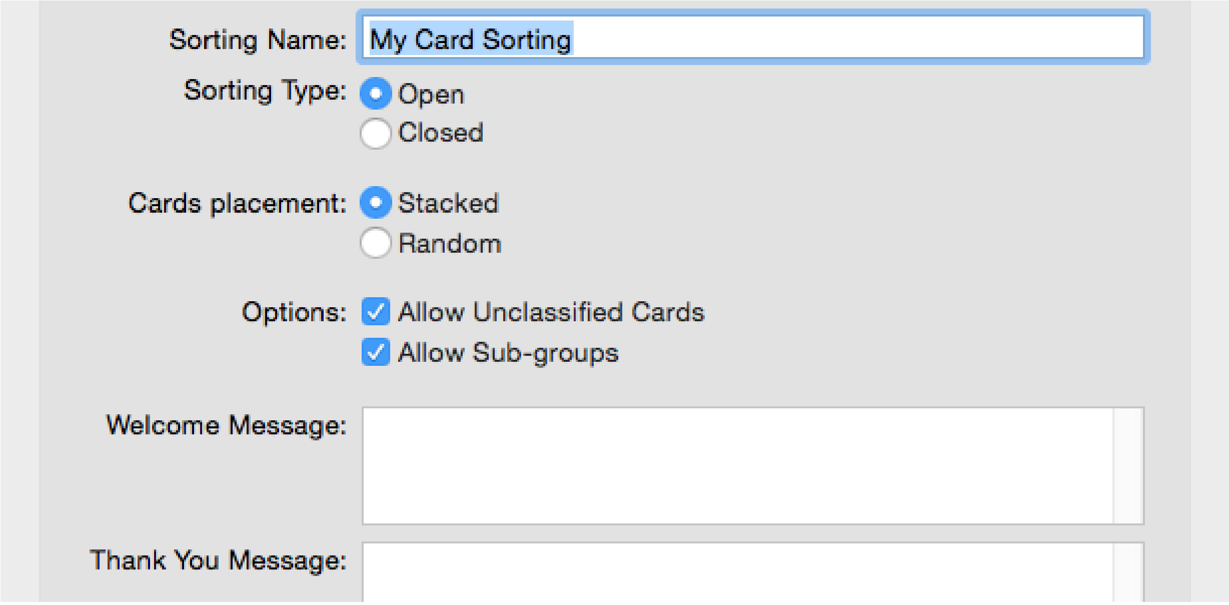
- Xsort is developed by Enough Pepper, a development company from Portugal
- Available only to Mac OS
- Free card sorting application aimed at simplifying the investigation of user experience

What to test

- The preference of difference users on the navigation menu of the application
- The card sort can be used when web sites are created “in a vacuum,” without little input.

Set up an experiment

- You can set up the experiment by entering the name of the sorting, selecting the sorting type or how to place the cards, whether it is stacked or random placed.
- The setup process is intuitive



Sorting Name: My Card Sorting

Sorting Type: ● Open ○ Closed

Cards placement: ● Stacked ○ Random

Options: ☑ Allow Unclassified Cards ☑ Allow Sub-groups

Welcome Message:

Thank You Message:

Features of the App

- Let say you have plenty of pages or functions in the application and you want to organize them into the way that the users want.
- Use XSort to collect users preference by inviting them to sort it for you

國際夥伴
會員計劃
養生食譜
全部活動
最新資訊
最新推廣

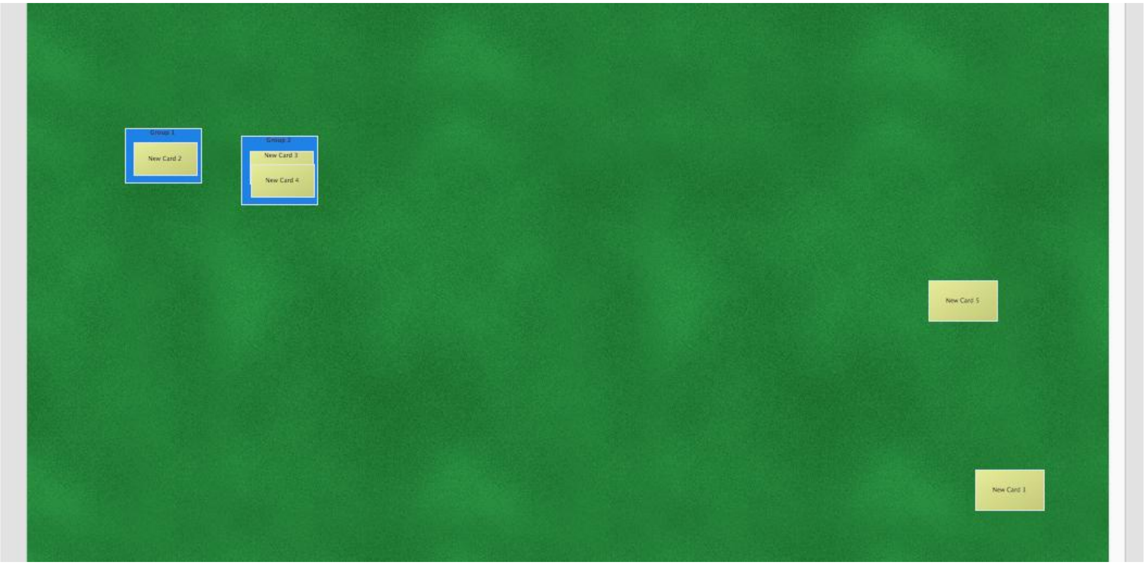
公司簡介
發展里程碑
最新活動
獎項
人才招聘
養生資訊
其他産品

健脾胃烏雞湯
聯絡我們
投資者關係
中醫團隊
網上購物
香港網絡

中國網絡
最新廣告
皇牌産品
公司聲明
人才培訓
研發製藥

Execute an experiment

- When you execute the experiment, users will be brought to a board with all the cards with the function name
- Users can start grouping the cards according to their own preference

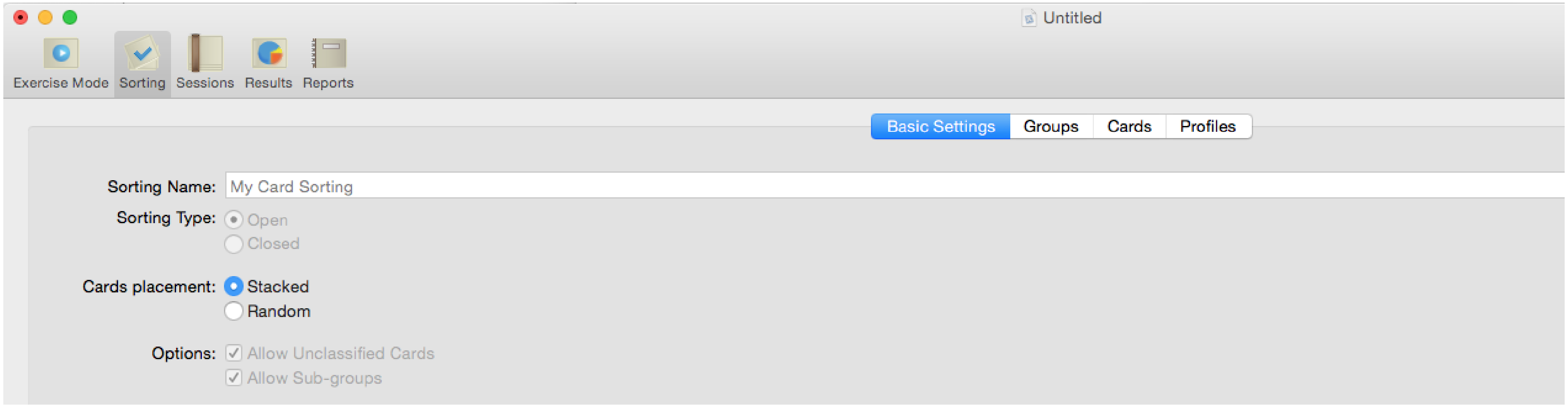


Installation: XSort

- Download Xsort
  - http://www.xsortapp.com/
- Put the downloaded file into Application folder of Finder

Usage: XSort

- Basic Settings
  - Set the name of your experiment
  - Set the type of cards placement



Sorting Name: My Card Sorting

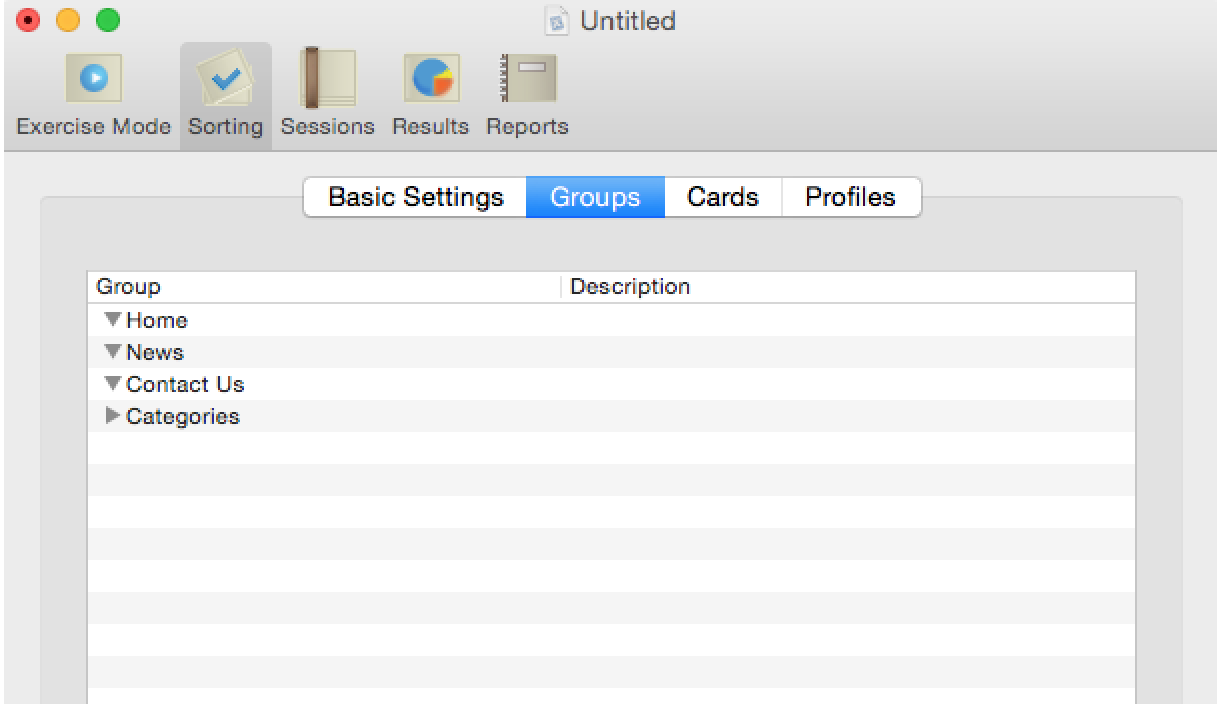
Sorting Type: ● Open ○ Closed

Cards placement: ● Stacked ○ Random

Options: ☑ Allow Unclassified Cards ☑ Allow Sub-groups

Usage: XSort

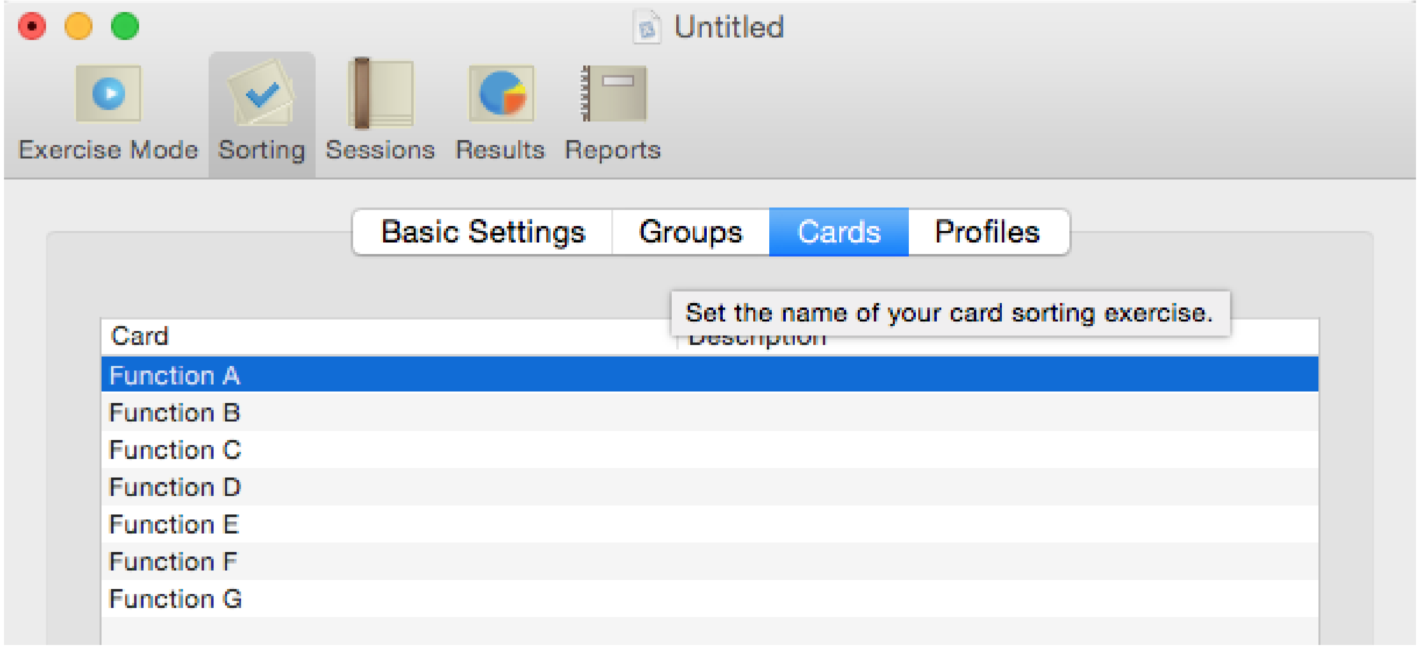
- Groups
  - Set the content of the main navigation



Group	Description
▼Home	
▼News	
▼Contact Us	
▶Categories	

Usage: XSort

- Cards
  - Set the functions that need to be grouped

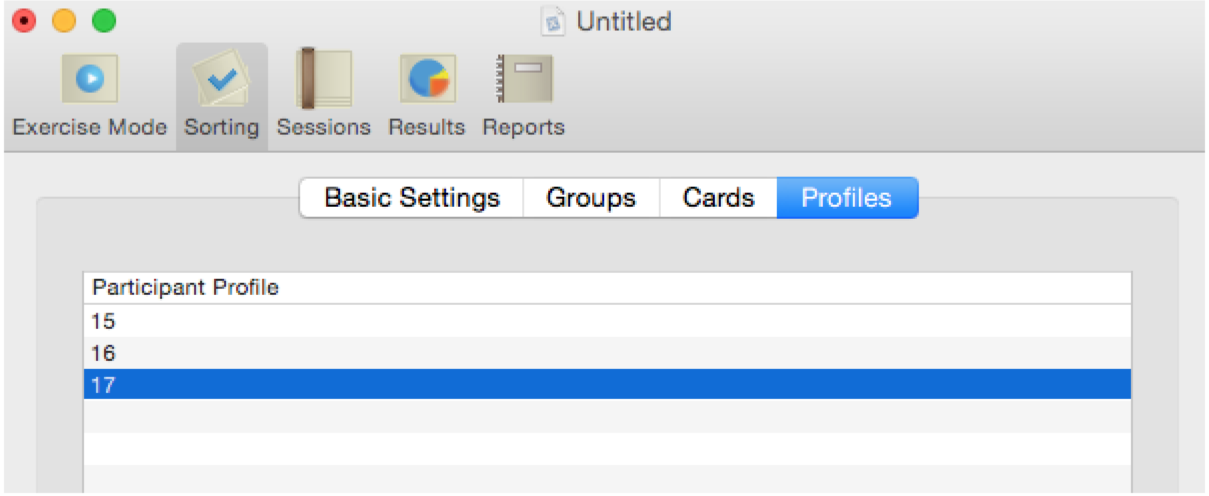


Set the name of your card sorting exercise.

Card	Description
Function A	
Function B	
Function C	
Function D	
Function E	
Function F	
Function G	

Usage: XSort

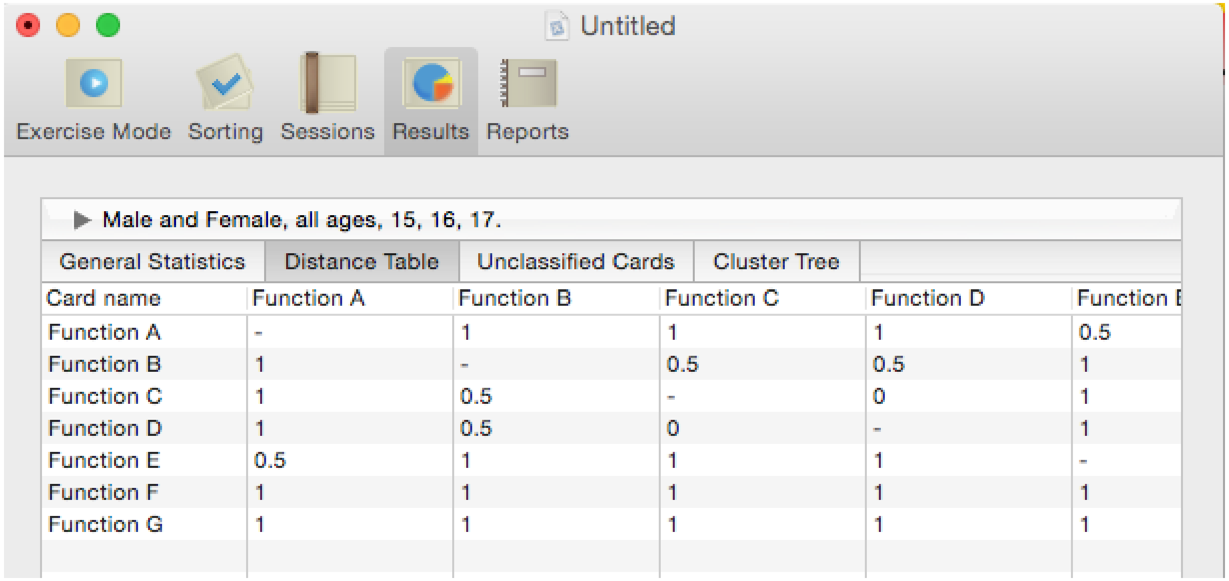
- Profiles
  - Set the profiles of the participants



Participant Profile
15
16
17

Report

- XSort provides statistical data of how the participants grouped the cards



▶ Male and Female, all ages, 15, 16, 17.

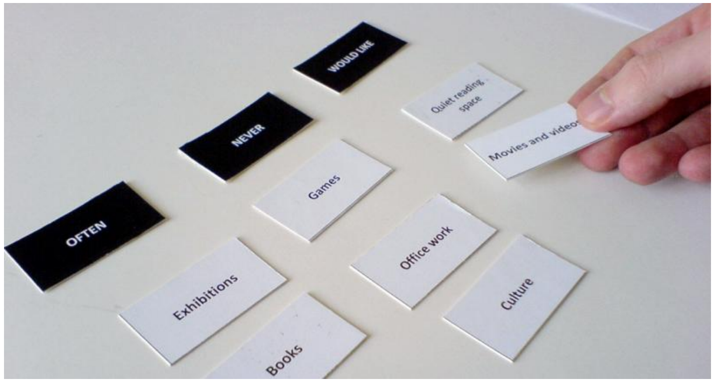
Card name	Function A	Function B	Function C	Function D	Function E
Function A	-	1	1	1	0.5
Function B	1	-	0.5	0.5	1
Function C	1	0.5	-	0	1
Function D	1	0.5	0	-	1
Function E	0.5	1	1	1	-
Function F	1	1	1	1	1
Function G	1	1	1	1	1

Live Demo

- Since Xsort only runs on Mac OS, we will do a short card sorting Demo

Discussion: Card Sorting and Target Users

- Will it be useful if we conduct card sorting by testers or developers?
- How can card sorting help your users to find items they want on your website?



4 GUI Testing: Sikuli Introduction

Learning Objectives #4

- Study an GUI tool which use image recognition to identify GUI components
- Test any Web, Mobile program with GUI
- Basic operation of Sikuli
  - Mouse actions
  - Keyboard actions
  - Find fucntion

GUI Tools Overview for Mobile

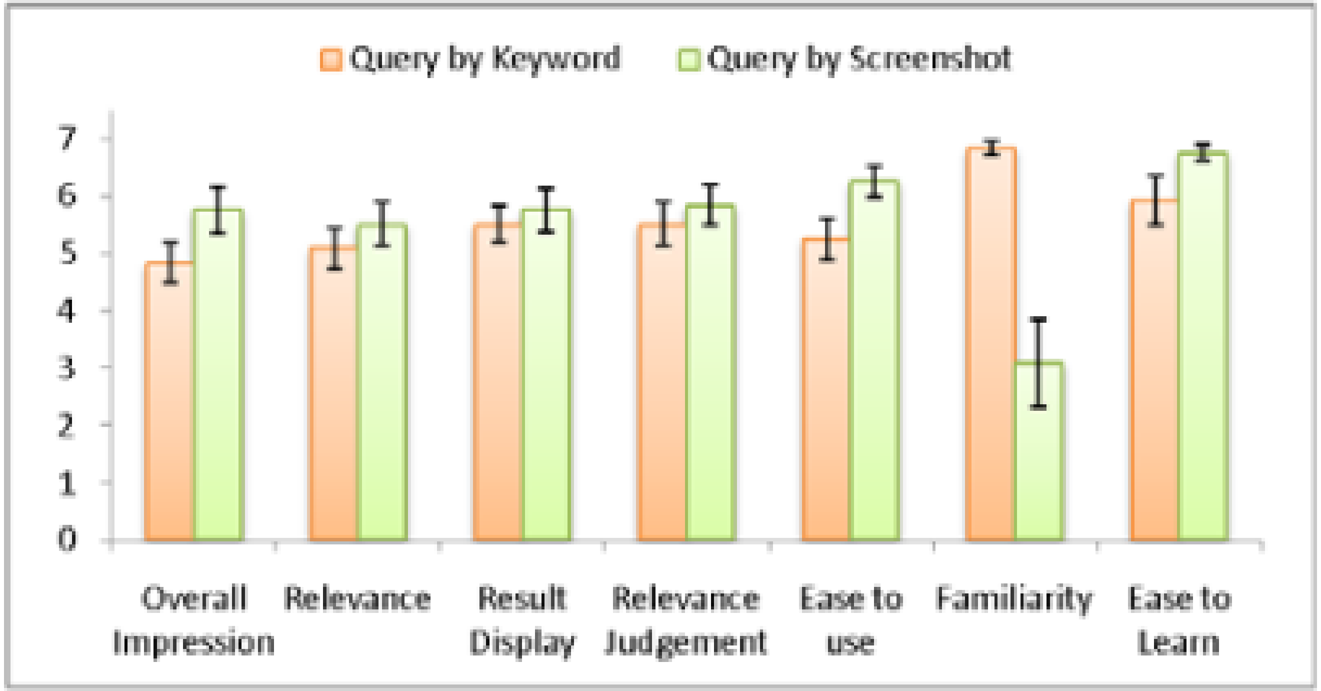
Sikuli	Testdroid	MonkeyRunner
Image recognition automated testing	Recording user-actions and producing JUnit based test cases	Provide API for writing programs that control android device or emulator
Manual scripting using Python	No coding needed	No coding needed
Same testing script can be test on different devices	Same testing script can be test on different devices	Limited to size of the device

A Brief History of Sikuli

- Originally an open-source research project at M.I.T.
- Now it is further developing by Raimund Hocke with the open-source community

Why Sikuli

- Easy for not too complex gui-test



Features

- Image recognition – Control and identify GUI components on screen
- GUI Testing without source code
- Automated GUI testing
- Use Jython as scripting language

python

Jython

Sikuli scripting language

- Jython is the official scripting language of Sikuli
- Jython: Java code to adapt Python environment
- Python and Java can be used in Sikuli script but not pure Java
- Java functionality can be used when Python fails to do so

python

Jython

Sikuli script

- Jython is the official scripting language of Sikuli
- Jython: Java code to adapt Python environment
- Python and Java can be used in Sikuli script but not pure Java
- Java functionality can be used when Python fails to do so

What to test

- Test the GUI to see if the GUI is working as expected by checking the results
- Test the GUI to see if there is any bug arose

Installation: SikuliX 1.1.0

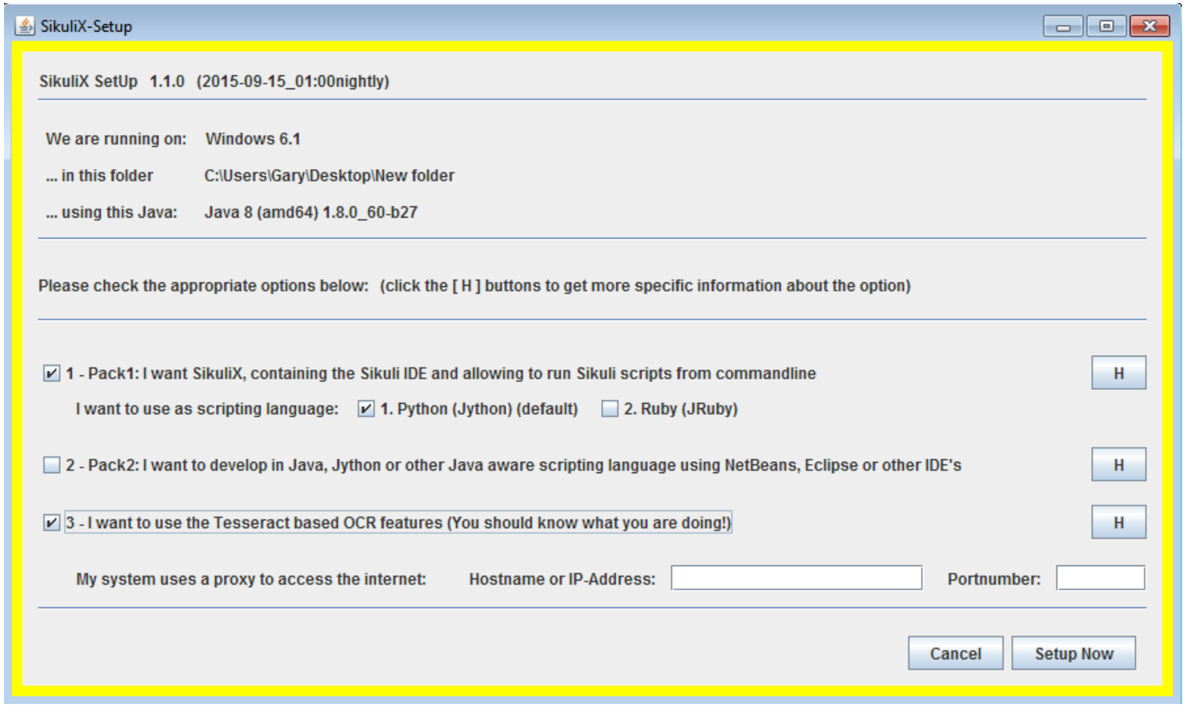
- Installing Java Environment
  - Download JDK 8 at Oracle
    - http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
  - Follow the instructions to install JDK 8
- Installing Python 2.7
  - Download Python 2.7 from python.org
  - Choose the Windows x86-64 MSI installer to download
  - Follow the instructions to install Python 2.7

Installation: SikuliX 1.1.0

- Download Sikuli from
  - https://oss.sonatype.org/content/groups/public/com/sikulix/sikulixsetup/1.1.0-SNAPSHOT/sikulixsetup-1.1.0-20150914.231031-164-forsetup.jar
  - Download the 1.1.0 version
  - Sikulixsetup-1.1.0.jar
  - Put it in a writable folder
- Same file also available in the course_files folder
- Double click the downloaded file to setup

Installation: SikuliX 1.1.0

- You should see the dialogue box in the previous slide
- Check option 1 & option 3
- Click Setup now



SikuliX-Setup

SikuliX SetUp 1.1.0 (2015-09-15_01:00nightly)

We are running on: Windows 6.1

... in this folder C:\Users\Gary\Desktop\New folder

... using this Java: Java 8 (amd64) 1.8.0_60-b27

Please check the appropriate options below: (click the [H] buttons to get more specific information about the option)

☑ 1 - Pack1: I want SikuliX, containing the Sikuli IDE and allowing to run Sikuli scripts from commandline [H]

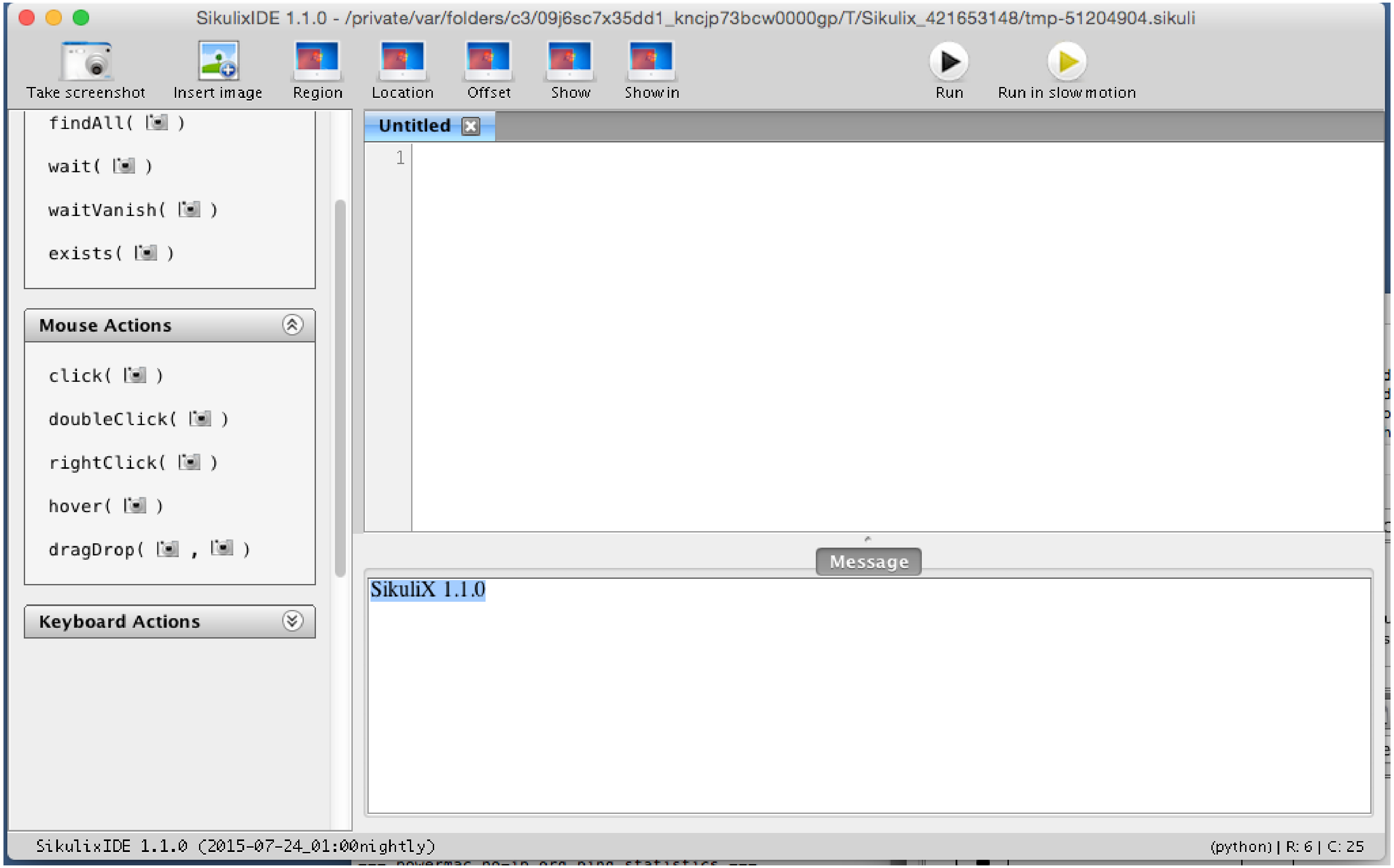
I want to use as scripting language: ☑ 1. Python (Jython) (default) ☐ 2. Ruby (JRuby)

☐ 2 - Pack2: I want to develop in Java, Jython or other Java aware scripting language using NetBeans, Eclipse or other IDE's [H]

☑ 3 - I want to use the Tesseract based OCR features (You should know what you are doing!) [H]

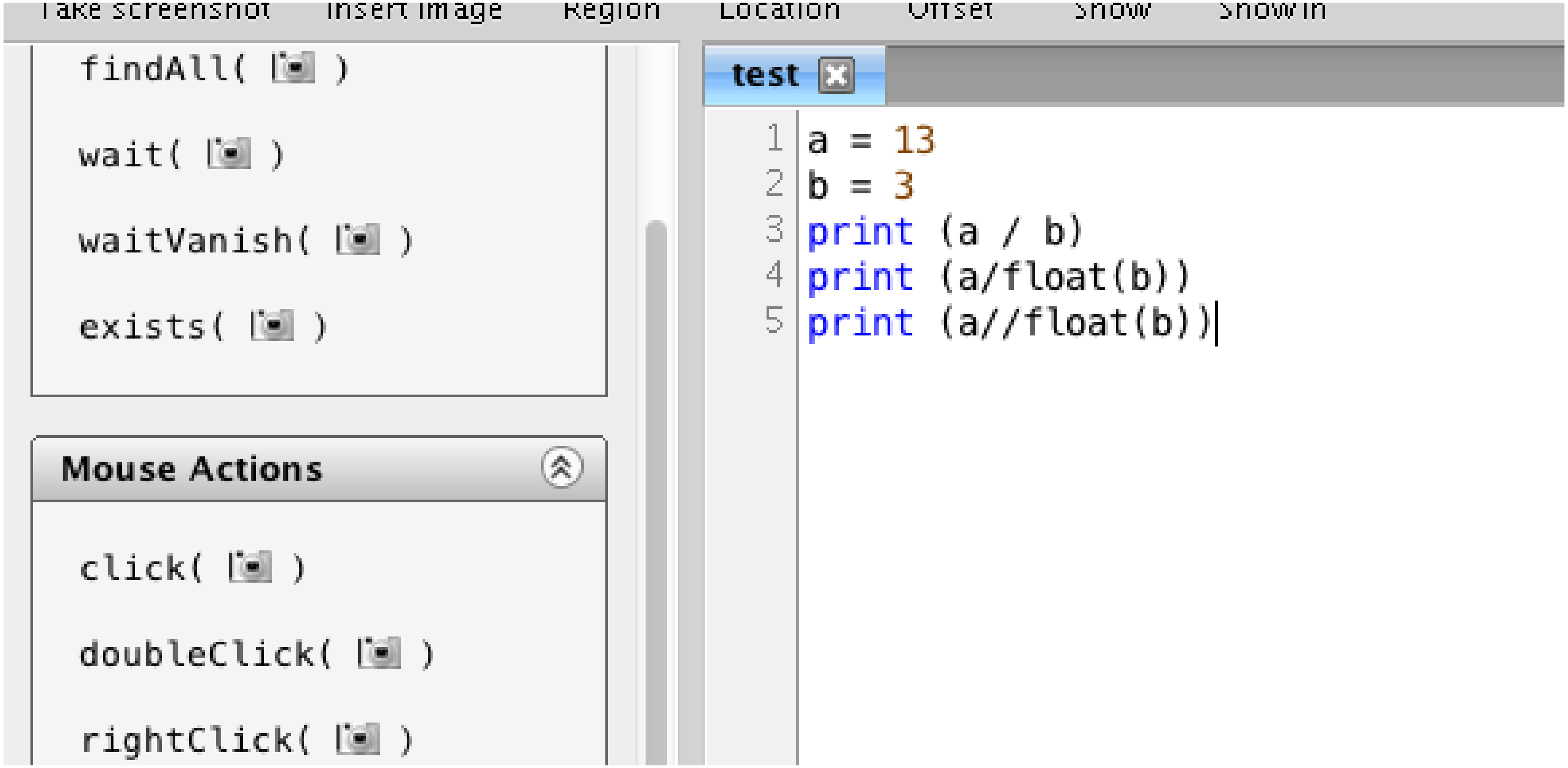
My system uses a proxy to access the internet: Hostname or IP-Address: Portnumber:

Sikuli IDE



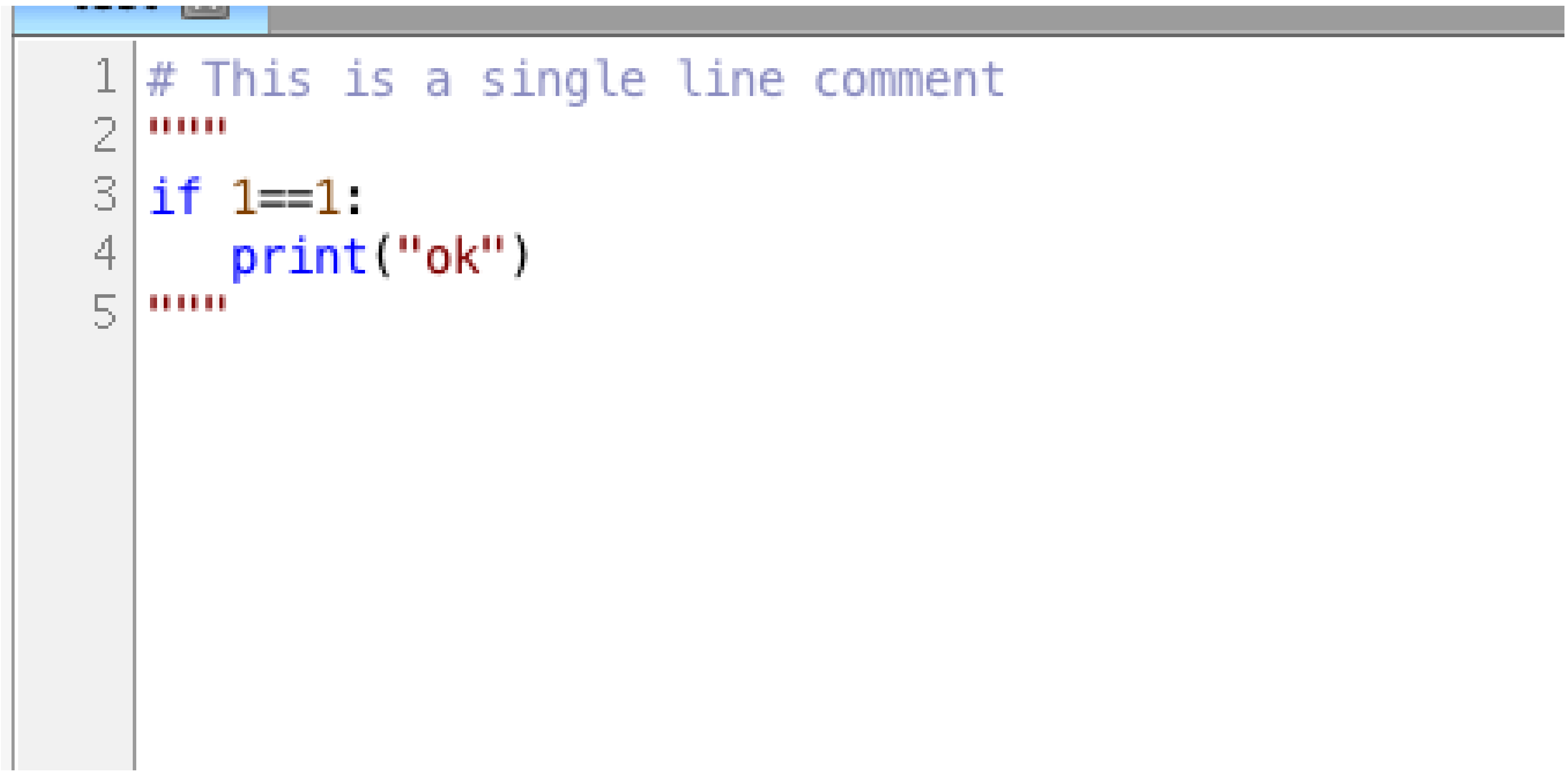
```
SikuliX 1.1.0
```

Python Environment



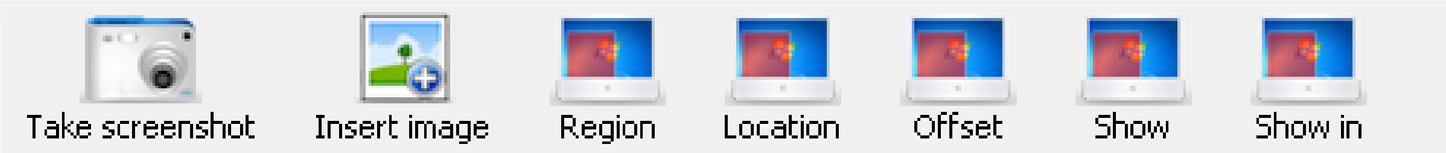
```
a = 13
b = 3
print (a / b)
print (a/float(b))
print (a//float(b))
```

Python Environment



```
# This is a single line comment
"""
if 1==1:
    print("ok")
"""
```

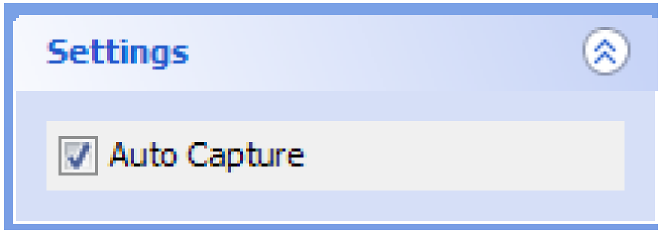
Menu Item



- Take Screenshot: Take a screenshot
- Insert Image: Insert an image from disks
- Region: Create a region to restrict screen matching
- Location: Select a location as the centre of selection
- Offset: Select offset as topLeft or bottomRight of selection
- Show: Show the item at the cursor
- Showin: Show the item at the cursor in the selected region

Setting

- With auto capture is checked, you will be directed to the screen capturing mode
  - when you insert a function
  - E.g. find()

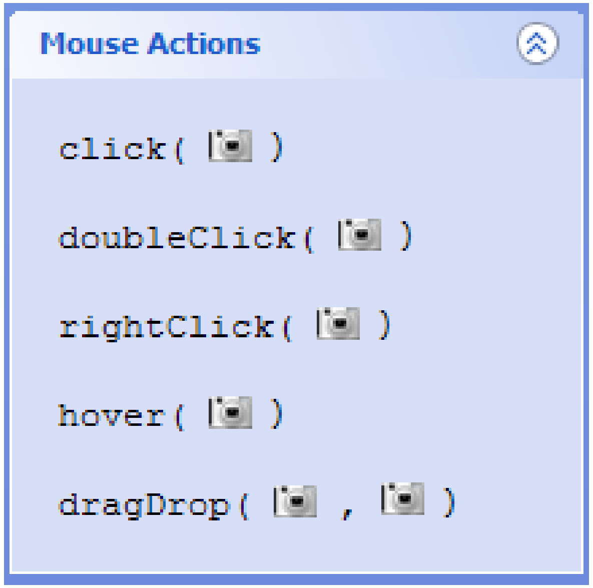


Settings

☑ Auto Capture

Mouse Action

- click: Clicks on the **best matched** GUI components on the screen or in the region
- doubleClick: Double clicks on the best matched GUI components on the screen or in the region
- rightClick: Right click on the best matched GUI components on the screen or in the region

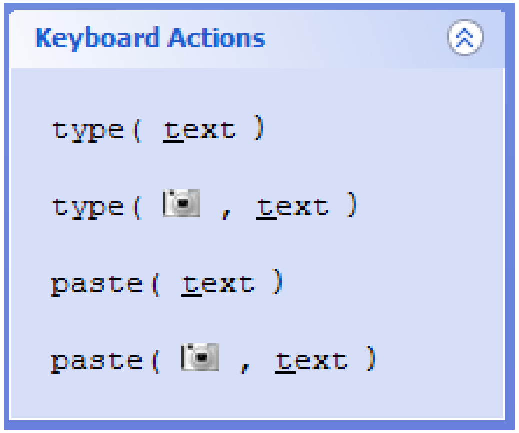


Mouse Actions

```
click( 📷 )
doubleClick( 📷 )
rightClick( 📷 )
hover( 📷 )
dragDrop( 📷 , 📷 )
```

Keyboard Action

- Type: Type the given text to the current focused window
- Type with click: Clicks on the given components to gain the focus and then types the given text
- Paste: Paste the given text to the current focused window
- Paste with click: Clicks on the given components to gain the focus and then pastes the given text

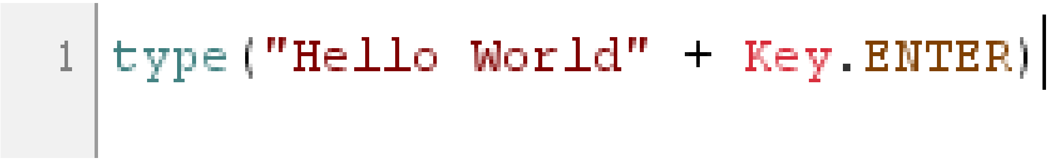


Keyboard Actions

```
type( text )
type( 📷 , text )
paste( text )
paste( 📷 , text )
```

Keyboard Action

- Using the Type function with Key class, you can program Sikuli to enter keystroke when running
  - E.g. type(“Hello World” + Key.ENTER)
  - Sikuli will then input Hello World to the field and press the Enter key on keyboard
  - Other keys: TAB, ESC, BACKSPACE, DELETE, INSERT…



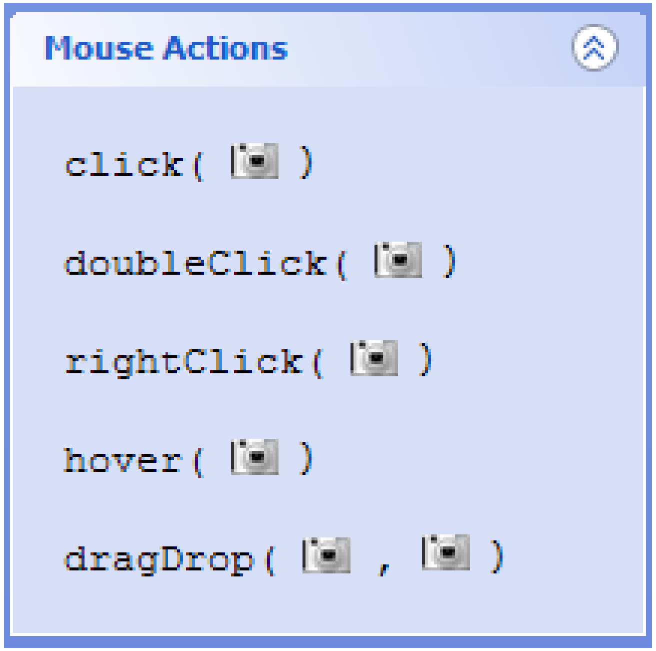
```
type("Hello World" + Key.ENTER)
```

Exercise #3 (15 mins)

- Automate the following task:
  - Click the firefox icon, open it,
  - type www.baidu.com
  - type your company name and click the button search

Mouse Action

- hover: Move the mouse cursor to the best matched GUI component
- dragDrop: Drag the first components and drops at the second components

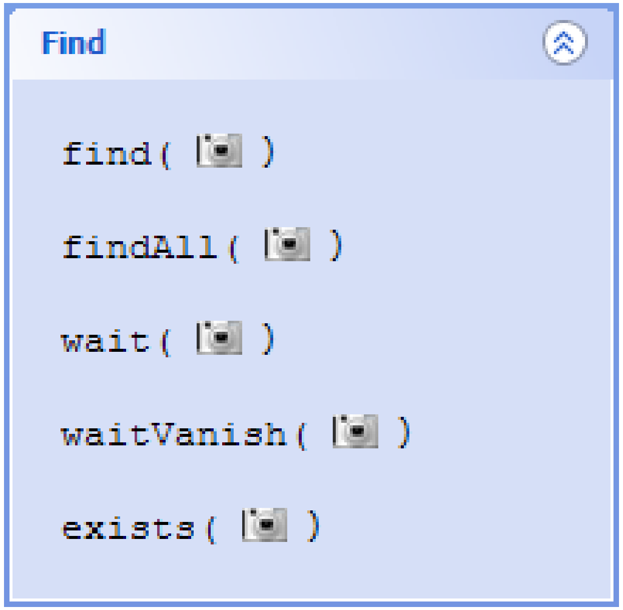


Mouse Actions

```
click(  )
doubleClick(  )
rightClick(  )
hover(  )
dragDrop(  ,  )
```

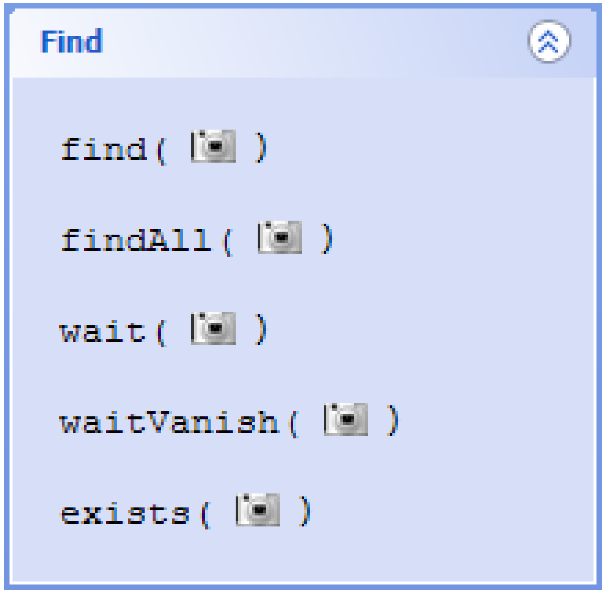
Find

- find: Looks for the best matched GUI components on screen or in the region
- findAll: Looks for all matched GUI components on screen or in the region
- wait: waits the given GUI components appear on screen or in the region for up to timeout seconds



Find

- waitVanish: waits the given GUI components disappear on the screen or in the region for up to timeout seconds
- exists: checks if the given GUI components exists on the screen or in the region, if not, none is returned without throwing an exception

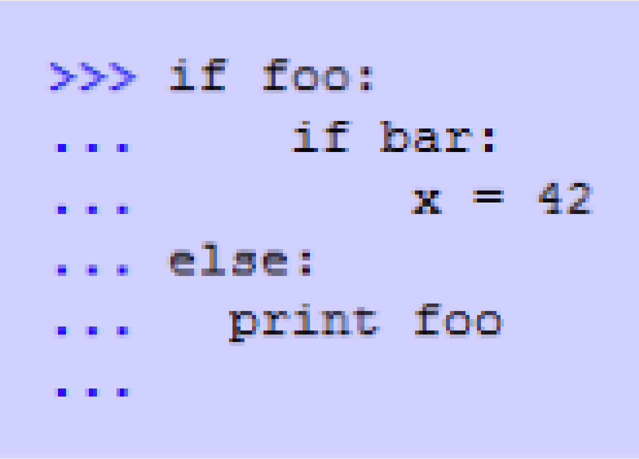


Find

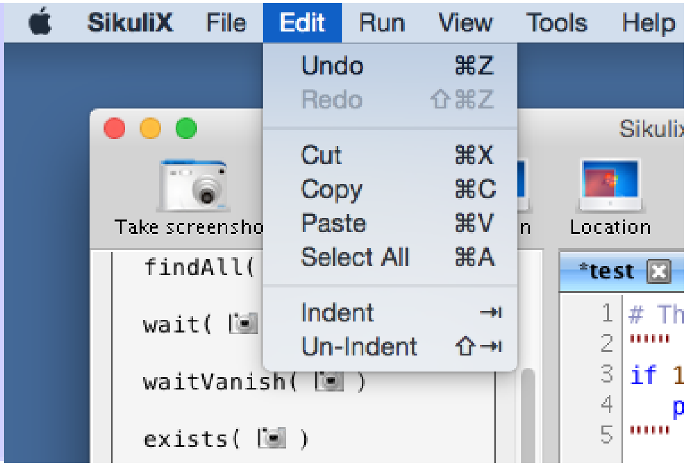
```
find( )
findAll( )
wait( )
waitVanish( )
exists( )
```

Indentation

- Indentation is important in python
- Different level of indentation divides different function of the code



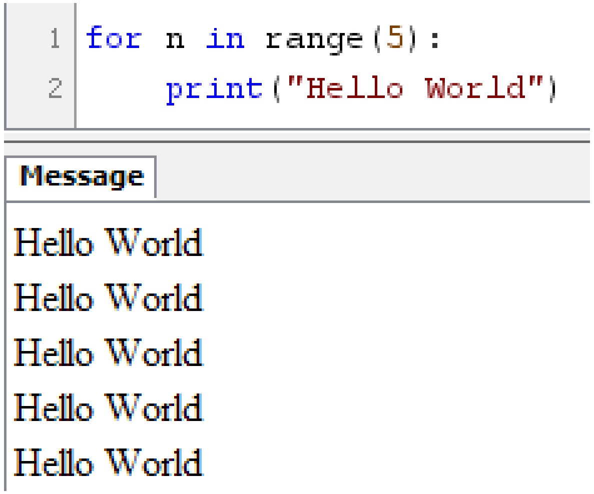
```
>>> if foo:
...       if bar:
...             x = 42
... else:
...    print foo
...
```



For loop

- Using for loop in Sikuli with python allows you to repeat the scripted actions
- for n in range(5)

  print(“Hello World”)
- The above code will print Hello World 5 times



```
for n in range(5):
    print("Hello World")
```

```
Hello World
Hello World
Hello World
Hello World
Hello World
```

Wait Action

- wait(x)
  - Suspends execution for the given number of seconds
  - The number x can be a floating point (decimal) number to indicate a more precise sleep time
- Very useful when dealing with slow internet connection

Exercise #4: Testing the Ads (20 mins)

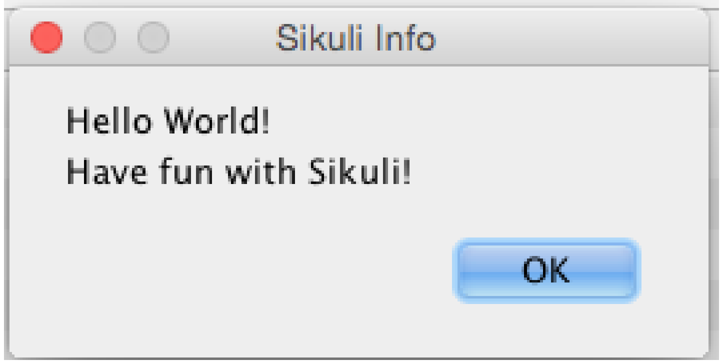
- Open a website with advertisement
  - E.g. etnet.com.hk
- Test the number of times the same ads appeared after 20 refreshes

LUNCH BREAK

4 GUI Testing: Sikuli Introduction

popup

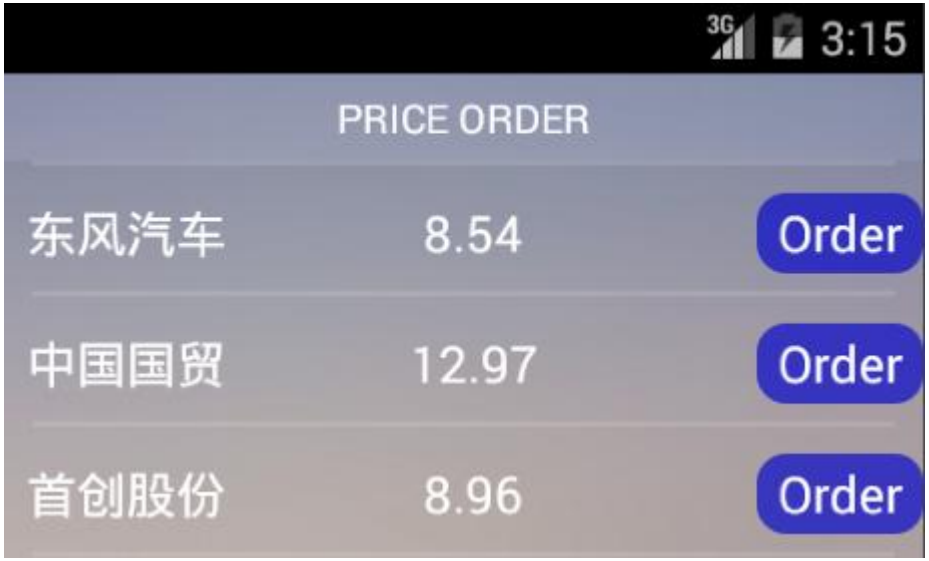
- popup("Hello World!\nHave fun with Sikuli!")
- Sikuli will send a popup window to print the given message



```
Hello World!
Have fun with Sikuli!
```

Target offset

- Sometimes there are identical GUI components, how to target a specific one?
- Let say we want to click on the first Order button in the below screen

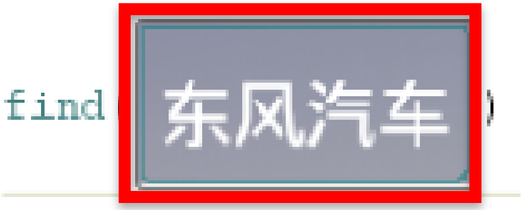


3G 3:15

PRICE ORDER

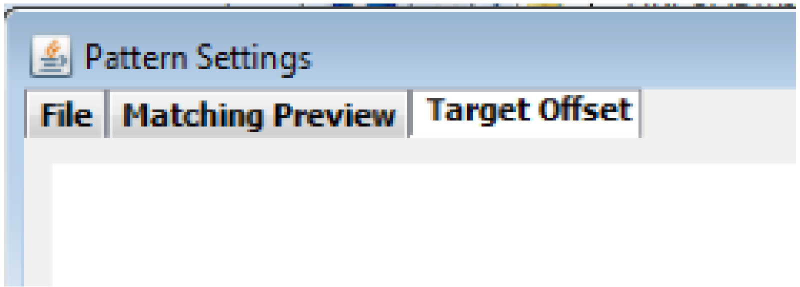
东风汽车	8.54	Order
中国国贸	12.97	Order
首创股份	8.96	Order

Target offset

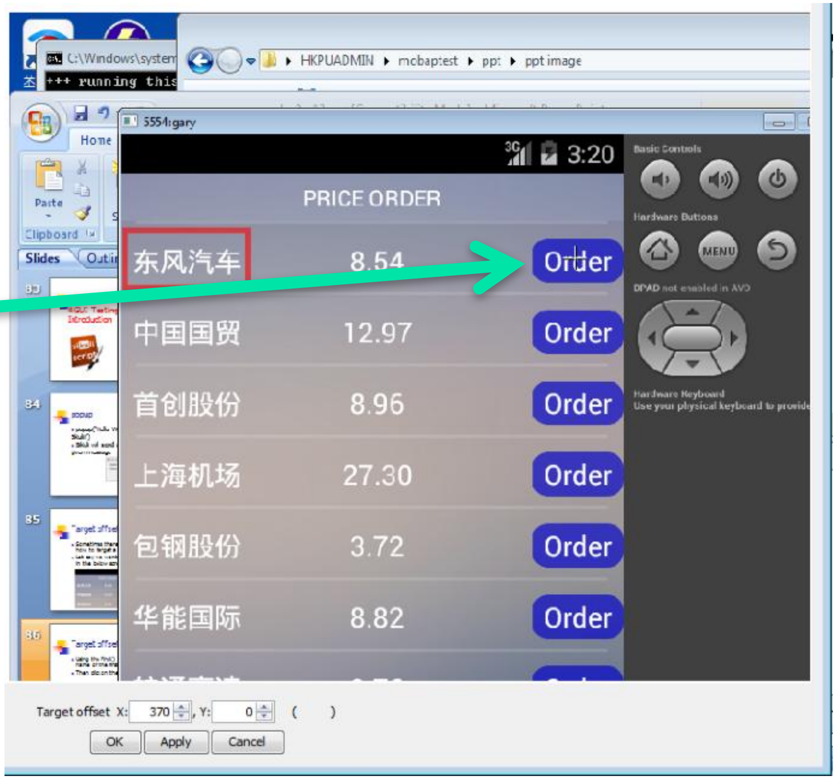


find(东风汽车)

- Using the find() function, we can capture the name of the first stock
- Then click on the image captured
- Select the Target offset tab
- Click on the Order button to set the offset value
- Apply the setting



Pattern Settings



PRICE ORDER

东风汽车	8.54	Order
中国国贸	12.97	Order
首创股份	8.96	Order
上海机场	27.30	Order
包钢股份	3.72	Order
华能国际	8.82	Order

Target offset X: 370, Y: 0 ( )

Starting Android emulator

- Setup an android emulator using Android studio (See slide 13)
- Start the Estock app using the emulator
- The project files can be found in the course_files folder

Key control

- Use Sikuli to start emulator
- In Windows 7, you can switch the orientation of the emulator by pressing ctrl and F11 together
  - type(Key.CTRL,Key.F11)

Exercise #5 (20 mins)

- Use Sikuli to test the Estock app
- Create an account then login to the app
- Buy 100 stock to check if it is successfully transact
- Buy 10000 stock and check if it is rejected by the system
- Sell 50 stock and check if the remaining sum is equal to 50

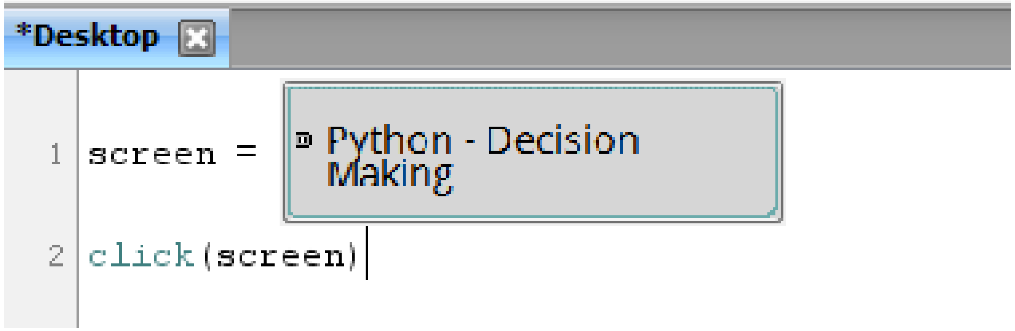
5 Sikuli: Advanced Techniques

Learning Objectives #5

- Using Sikuli with some Python build-in function for advanced task
    - Catching exception to prevent program from stopping
    - Open an app and switch an app with Sikuli
    - Relative position
    - Comparing screenshots
    - Record the test using a log file

Variable

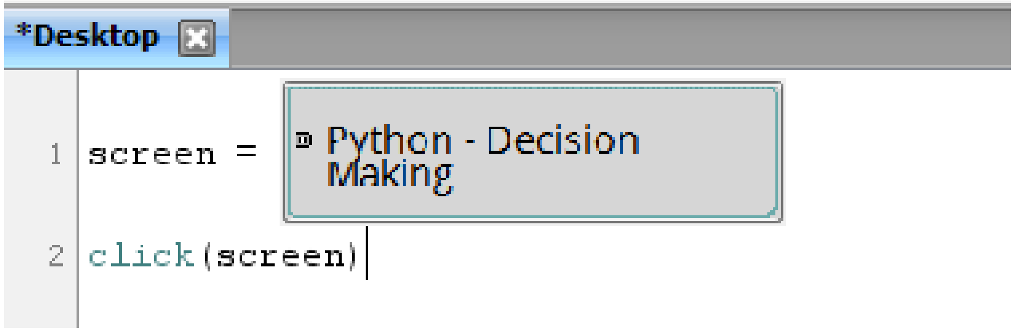
- Python variable do not need explicit declaration to reserve memory space
  - E.g. counter = “John”
- In Sikuli, you can assign a captured screenshot to a variable



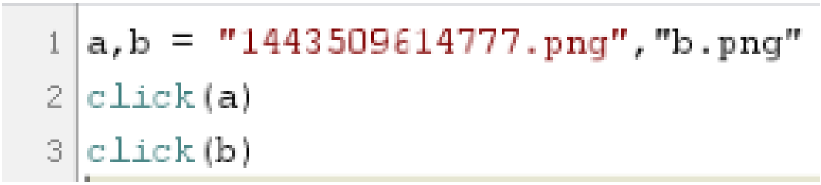
```
screen =
click(screen)
```

Assigning image to variable

- A image can be directly assigned to a variable in sikuli
- Apart from capturing an image when assigning it to a variable, you can also assign it directly using the image’s file name
- Assigning multiple images to multiple variables is also possible

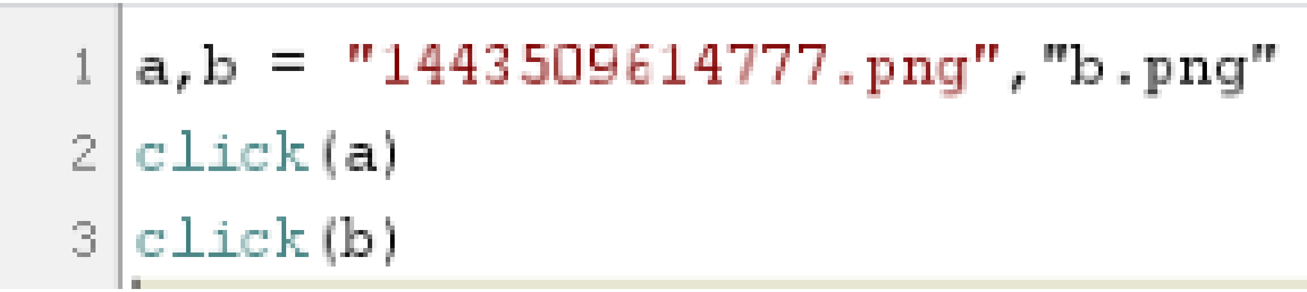


```
screen = 
click(screen)
```



```
a,b = "1443509614777.png","b.png"
click(a)
click(b)
```

Assigning image to variable

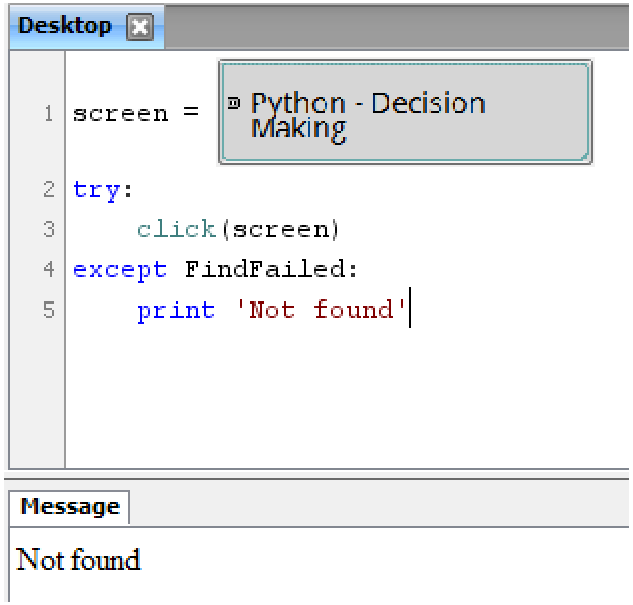


```
a,b = "1443509614777.png","b.png"
click(a)
click(b)
```

- Put the target image in the same folder of the sikuli script
- Assigning the image file name to the variable:
  - For example a is the variable and firefox.png is the file name
  - a = “firefox.png” will assign the image to the variable a

Try and Except

- Python allows users to catch an exception using try and except
- Sikuli include this function so the program will not shut down when Sikuli can't find the button we indicated
  - try:
    - …..
  - except FindFailed:
    - ……….

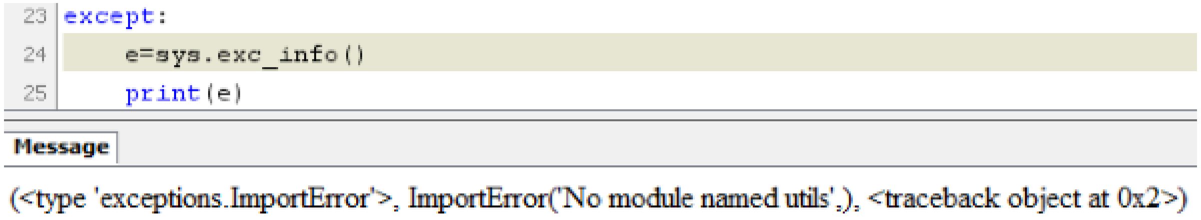


```
screen =
try:
    click(screen)
except FindFailed:
    print 'Not found'
```

```
Not found
```

Try and Except

- Using python you can program the system to print a custom error message
- You can also program it to print the system error information
  - except:
    - e=sys.exc_info()
    - Print(e)



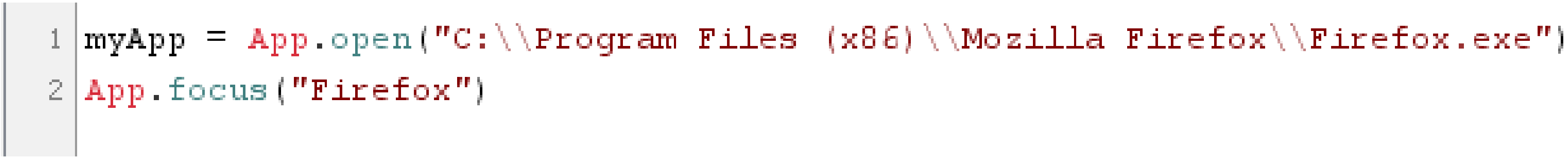
```
except:
    e=sys.exc_info()
    print(e)
```

```
(<type 'exceptions.ImportError'>, ImportError('No module named utils',), <traceback object at 0x2>)
```

- It gives more detailed explanation of the error

Open an App and Switch an App

- Sikuli allows you to use the App class to start an application that is installed
- For example, we want to open the Firefox browser and focus on the browser window:



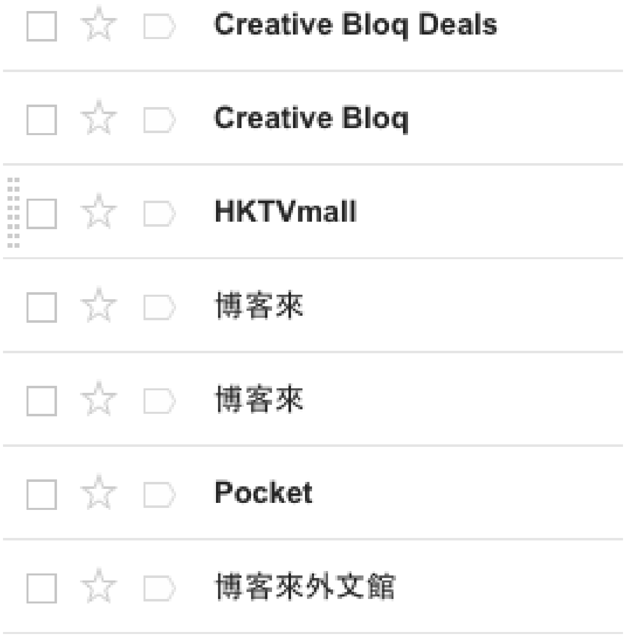
```
myApp = App.open("C:\\Program Files (x86)\\Mozilla Firefox\\Firefox.exe")
App.focus("Firefox")
```

App.open() & App.focus()

- App.open()
  - Open the application you desired by including the path to the app in the curly braces
  - E.g.: App.open(C:\\Program Files(x86)\\Mozilla Firefox\\Firefox.exe)
- App.focus()
  - Switch the focus to the application you desired by including the identifier of the app in the curly braces
  - E.g.: App.focus(Firefox)

Relative Position

- In some situations you can’t simply program Sikuli to click on a given GUI components, like in a web email app
- All the check box looks exactly the same
- Using relative position can tell Sikuli which box to click on



□ ☆ Creative Bloq Deals

□ ☆ Creative Bloq

□ ☆ HKTVmall

□ ☆ 博客來

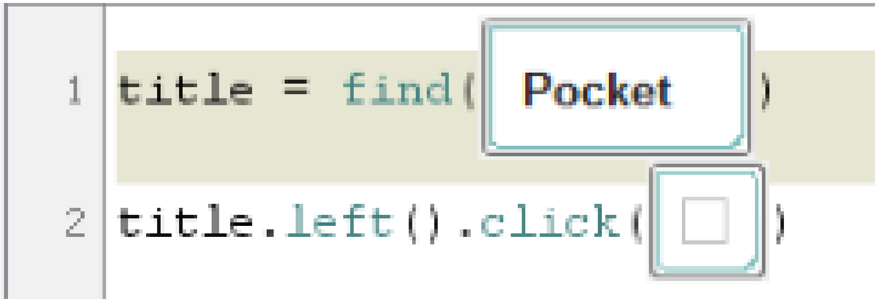
□ ☆ 博客來

□ ☆ Pocket

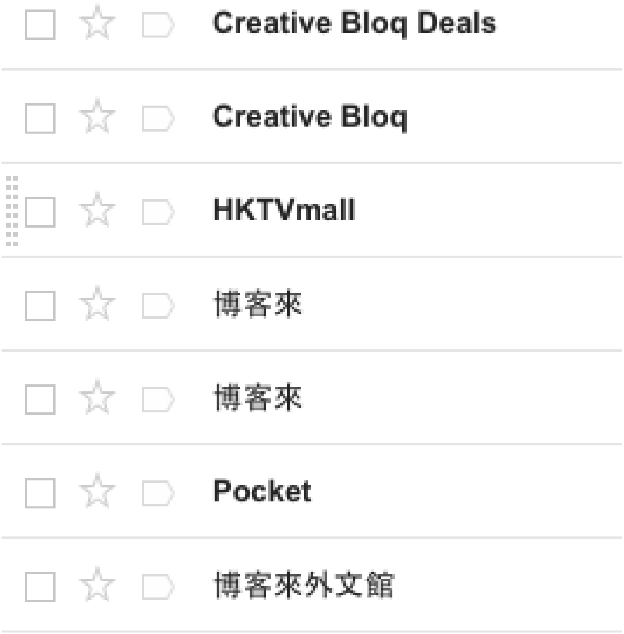
□ ☆ 博客來外文館

Relative Position

- Say we need to click on the box next to "Pocket"
- We can first ask Sikuli to find Pocket and then click the box next to it:



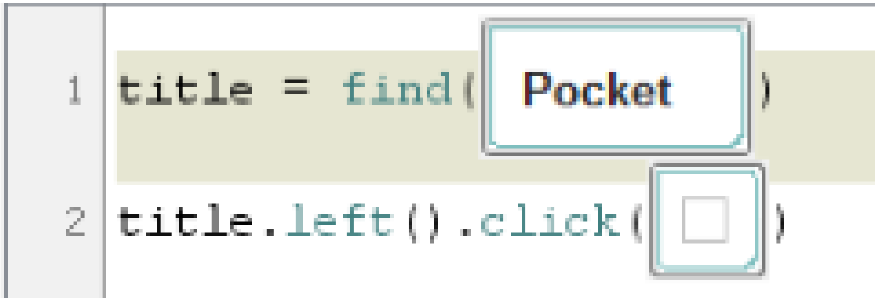
```
title = find( Pocket )
title.left().click( □ )
```



□ ☆ Creative Bloq Deals
□ ☆ Creative Bloq
□ ☆ HKTVmall
□ ☆ 博客來
□ ☆ 博客來
□ ☆ Pocket
□ ☆ 博客來外文館

Relative Position

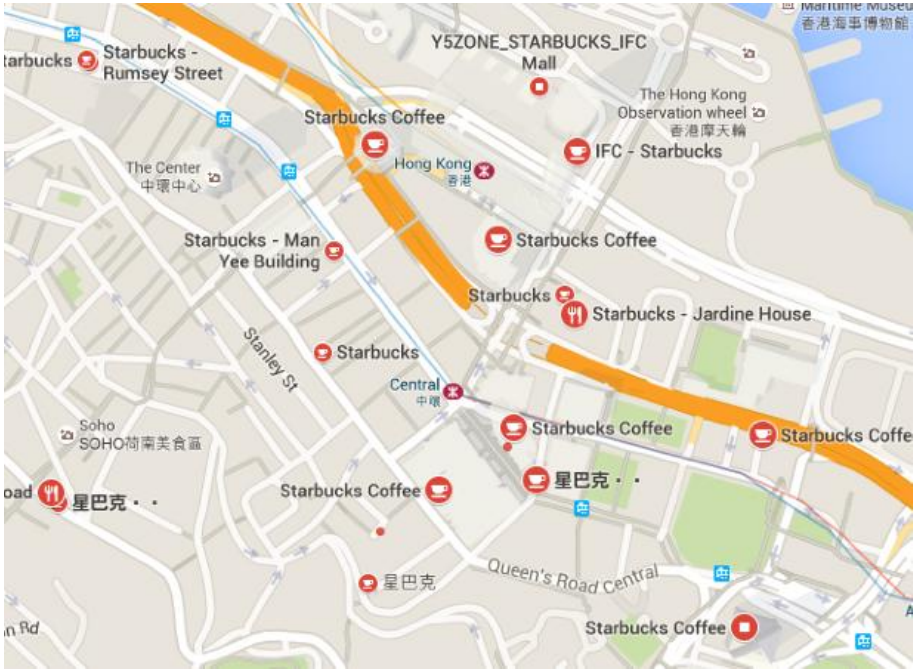
- First assign the position of “Pocket” to a variable
- title.left().click([img]):
  - On the left of title, click the checkbox
- Other directions:
  - right()
  - above()
  - below()



```
title = find( Pocket )
title.left().click( □ )
```

Map

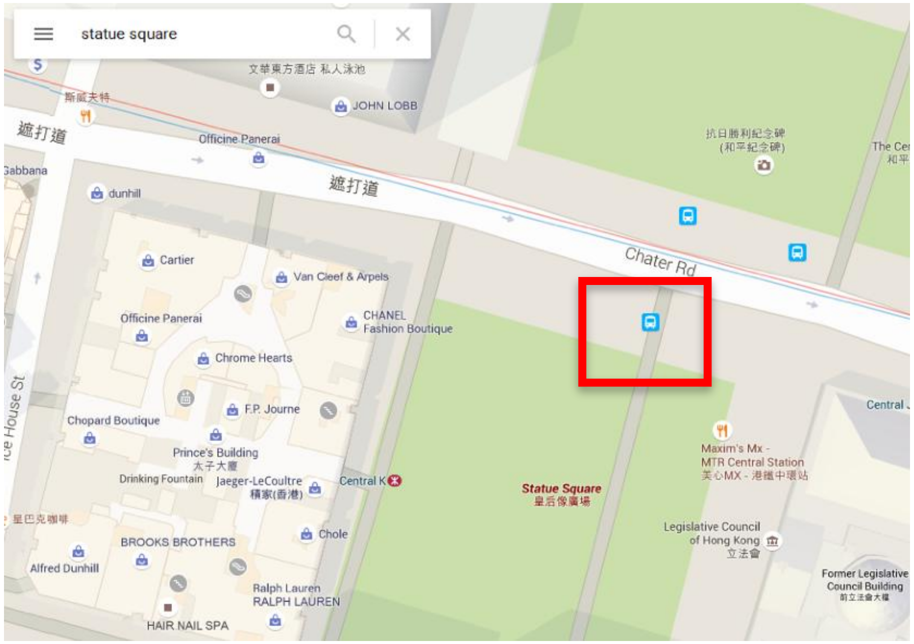
- We can use Sikuli to use the Google map app
- Baidu map will also work as fine
- E.g. we want to find out the number of Starbucks coffee shop in Central



■ Open Google Map and type Starbucks Central
■ The map will mark all the shop using a red cup icon
■ We can then use Sikuli to count the number of icon

Exercise #6 (15 mins)

- Use relative position in Sikuli to automate the following process:
  - Open google map using web browser
  - Click on the bus stop near Statue Square at Charter Road

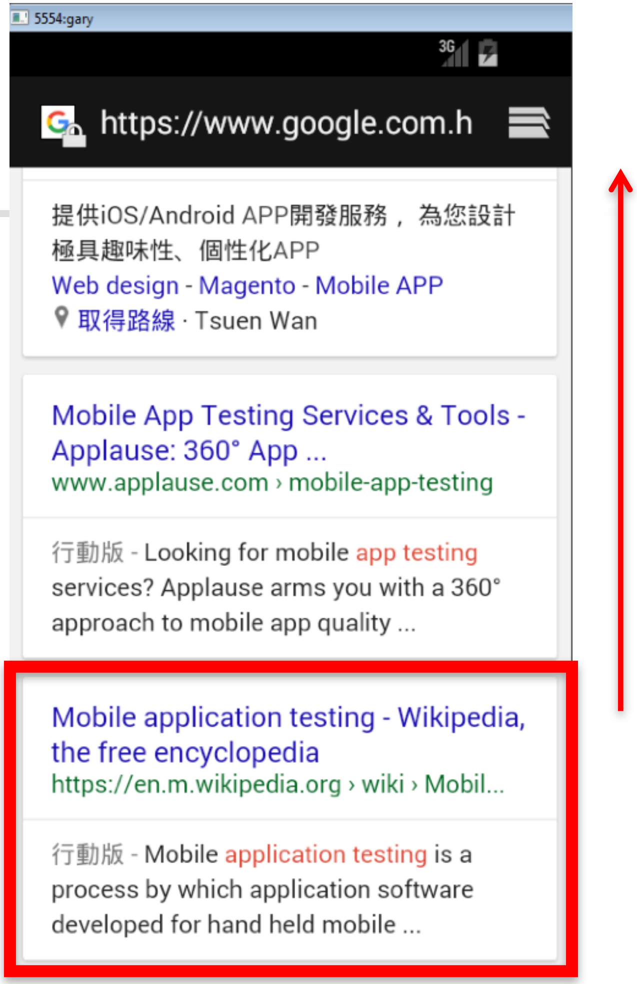


Drag & Drop

- dragDrop(a,b) is a function allows you to set the cursor to hold and drag to a designated point
- The two parameters in the functions are:
  - a = images meaning Sikuli will drag from the first image to;
  - b = the second image before release the cursor

Drag & Drop

- E.g. We want to program Sikuli to drag the page downward
- We can set Sikuli to drag from the wikipedia link upward to the browser url
  - dragDrop(image1, image2)



5554:gary

https://www.google.com.h

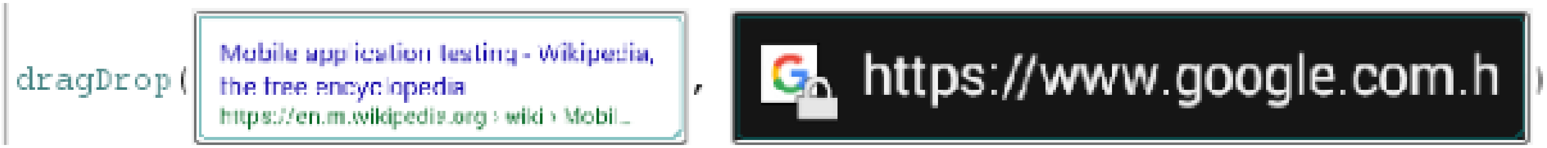
提供iOS/Android APP開發服務，為您設計極具趣味性、個性化APP
Web design - Magento - Mobile APP
取得路線 · Tsuen Wan

Mobile App Testing Services & Tools - Applause: 360° App ...
www.applause.com › mobile-app-testing

行動版 - Looking for mobile app testing services? Applause arms you with a 360° approach to mobile app quality ...

Mobile application testing - Wikipedia, the free encyclopedia
https://en.m.wikipedia.org › wiki › Mobil...

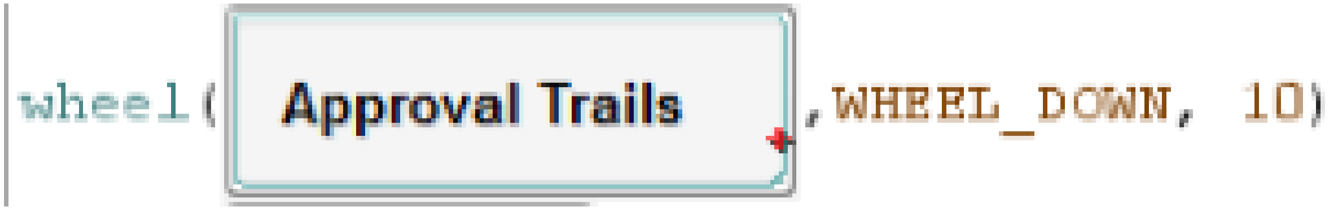
行動版 - Mobile application testing is a process by which application software developed for hand held mobile ...



```
dragDrop( [Mobile application testing - Wikipedia, the free encyclopedia https://en.m.wikipedia.org › wiki › Mobil...] , [https://www.google.com.h] )
```

Wheel

- Wheel(a,b,c) is a function that move the mouse to a specified position and turn the mouse wheel
- Parameters:
  - a: a region/ a match / a location (screen capture)
  - b: WHEEL_UP or WHEEL_DOWN (direction)
  - c: number of steps



```
wheel( ,WHEEL_DOWN, 10)
```

Email Notification

```
#!/usr/bin/python
import smtplib
sender = 'sikuli@test.hk'
receivers = ['ours@email.hk']
message = """From: From Person <sikuli@test.hk>
To: To Person <yours@email.hk>
Subject: SMTP e-mail test
This is a test e-mail message.
"""
try:
  smtpObj = smtplib.SMTP('comp.polyu.edu.hk')
  smtpObj.sendmail(sender, receivers, message)
  print "Successfully sent email"
except:
   print "Error: unable to send email"
```

Sending an email is easy.

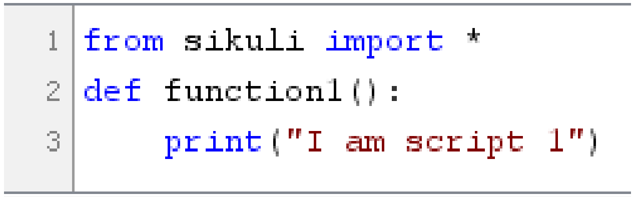
- Sending email with Sikuli is useful when you automate Sikuli to do a sequence of automation testing
- You can program Sikuli to send an email when the testing is successfully finished or error occured

Python: Call function

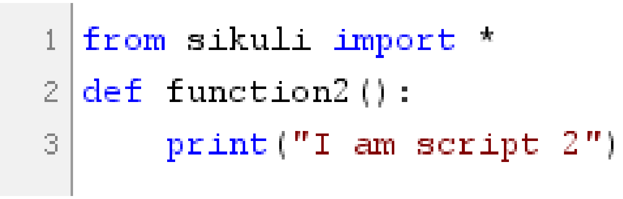
- With function call, a big program can be divided into smaller parts. Moreover, function can be reused when needed
- In Python, you can define a function:
  - Def functionName()
    - print("Hello World")
    - return 1234
- Whenever you need to call a function:
  - functionName()

Import Sikuli

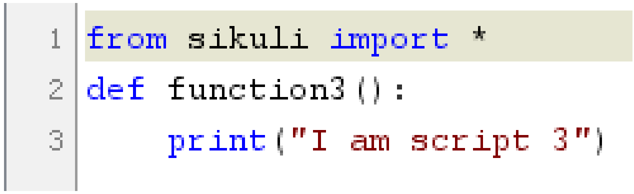
- You can combine several Sikuli scripts by importing a script
- For example, we have 3 Sikuli scripts called
  - script1, script2, script3



```
from sikuli import *
def function1():
    print("I am script 1")
```



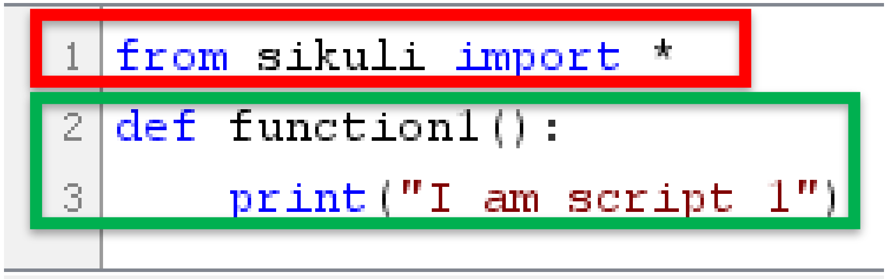
```
from sikuli import *
def function2():
    print("I am script 2")
```



```
from sikuli import *
def function3():
    print("I am script 3")
```

Import Sikuli

- In every script that need to be imported, the line “from sikuli import *” must be included in the script



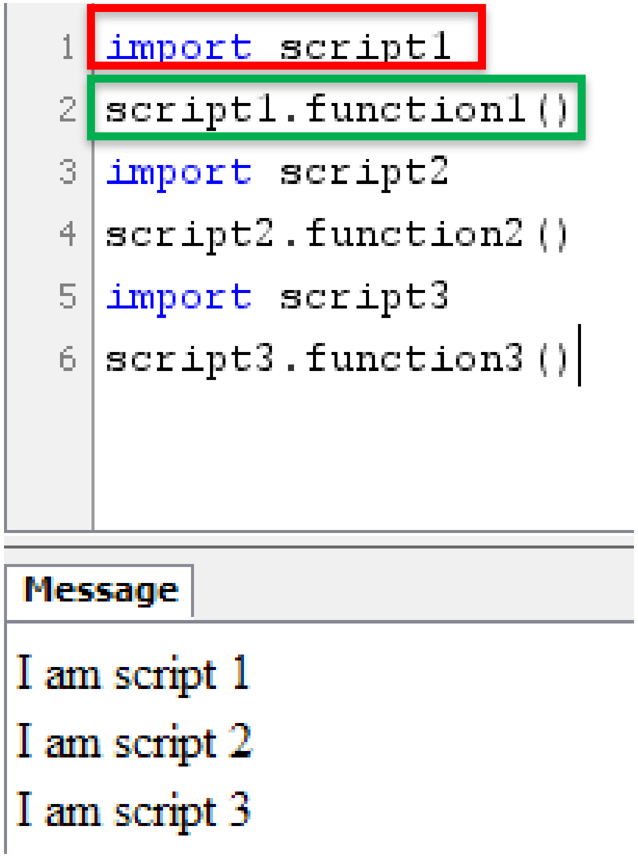
```
from sikuli import *
def function1():
    print("I am script 1")
```

- The content of the script is put inside a function

More explanation: https://answers.launchpad.net/sikuli/+faq/2446

Import Sikuli

- Besides the three scripts, an additional script is needed to combine all the other
  - E.g. combine.sikuli
  - First import the script
  - Then call the function of that script
- If you see an error, try to restart the sikuli IDE and run again



```
import script1
script1.function1()
import script2
script2.function2()
import script3
script3.function3()
```

```
I am script 1
I am script 2
I am script 3
```

Repetitive work

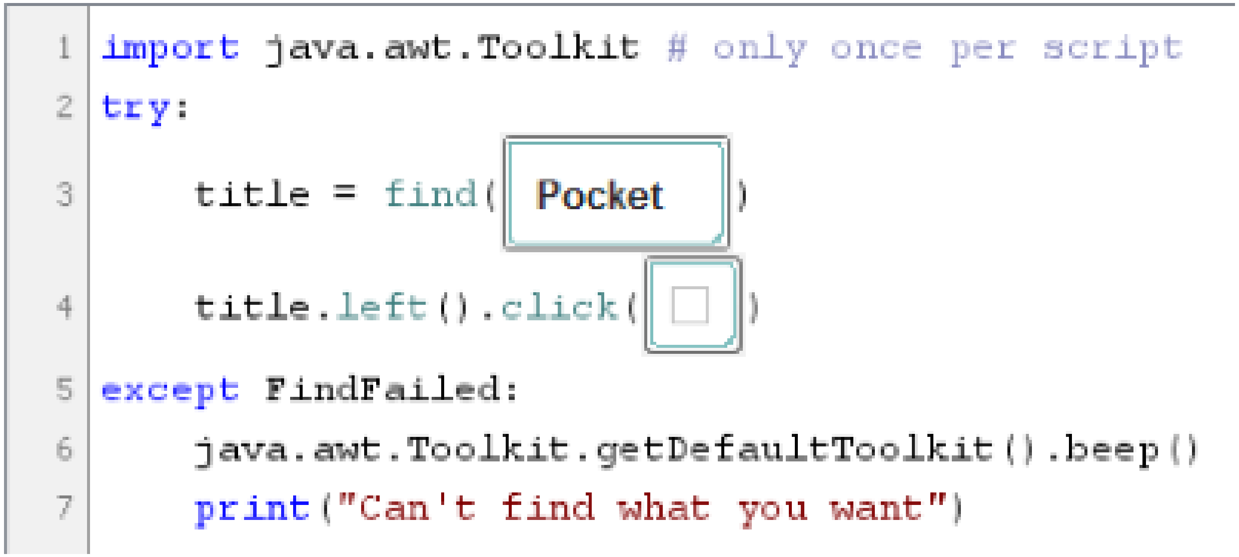
- Sikuli is useful in automating some repetitive tasks
- For example, the Joget App, it is workflow application. It can be used as a leave request application in a company
- Sikuli can be used to answer to request with pre-assigned actions
- E.g. Ask for reason when employees request a leave

Exercise#7 (20 mins)

- Start Joget app, in the course_files > session 2 > Sikuli > Joget_v4 > double click joget-start.bat
- Open the browser and type localhost:7080/jw
- Login as
  - Username: cat   Password: password
- Select the Leave Claim function
- Request a leave claim
- Logout as cat then Login as
  - Username: clark   Password: password
- Reject the leave request
- Automate the above process using sikuli
- **Split the script into parts and call the parts using function when needed, different parts can be reused in real situation**

Play a sound

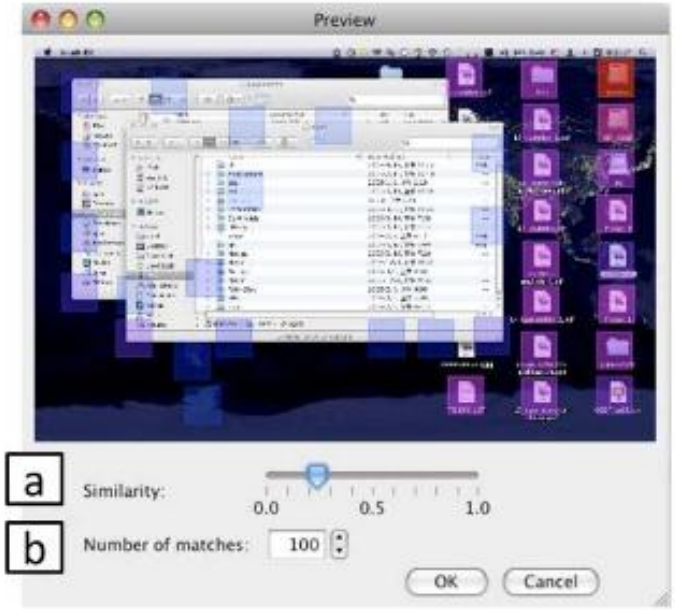
- Sikuli allows importing of Java class
- We can import Toolkit class and program Sikuli to produce a beep sound when an exception is catched
  - import java.awt.Toolkit # only once per script
  - java.awt.Toolkit.getDefaultToolkit().beep()



```
import java.awt.Toolkit # only once per script
try:
    title = find( Pocket )
    title.left().click( □ )
except FindFailed:
    java.awt.Toolkit.getDefaultToolkit().beep()
    print("Can't find what you want")
```

Similarity Threshold

- Adjust the similarity threshold
- Example is given as the threshold (0.25) which is too low, resulting in many false positives

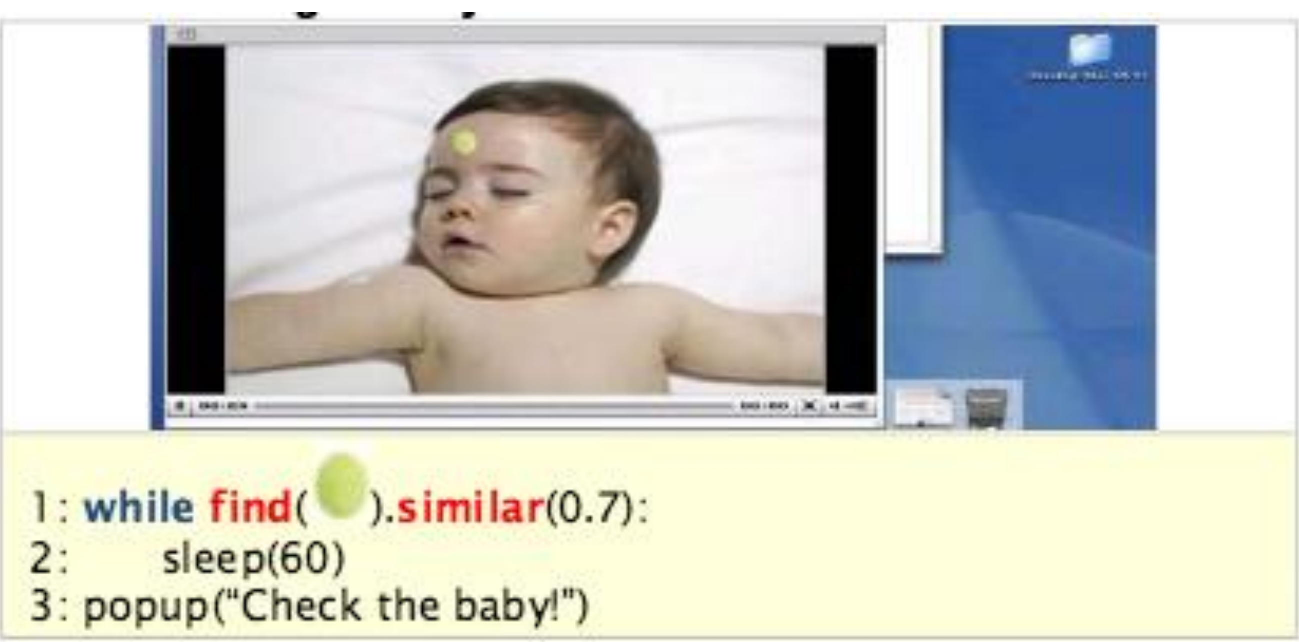


Similarity:

Number of matches: 100

Similarity Application

- Similarity can be use for detection of movement
- e.g. Baby monitoring: when the green spot on the baby’s head is not in the monitor, Sikuli will fire an alarm



```
while find(●).similar(0.7):
    sleep(60)
popup("Check the baby!")
```

Different Scale

- Since Sikuli mainly use image recognition, using the same image to test different devices with different size might not work
- To solve this, we can set the similarity threshold so Sikuli can still recognize the GUI component

Sikuli Log

- You can program Sikuli to record in a log file
  - *# when run in IDE with reruns to avoid errors use*
    ```
    import logging
    ```
  - *#set the file to log to and setting the logging level*
    ```
    logging.basicConfig(filename='example.log',level=logging.DEBUG)
    ```
  - *#writing to log file*
    ```
    logger.debug('We are starting now')
    logger.debug('We are finished now')
    ```
- Logging is useful when you have a long program
- We can set the program to log at different stage of the program so we can keep track on whether the program is running as we expected

Exercise #8 (20 mins)

- Install the Tic Tac Toe game to the emulator
- Try to use the skills you learnt today to test the application
- Use Sikuli to play the game automatically for 5 times
- Try and catch if any error arise
- Record in a log file

Exercise #8 (20 mins)

- Installing apk into a emulator:
  - Use adb tools
- Copy the apk to android-sdk/tools directory
- Then open command prompt
  - adb install tac.apk

Compare Two Screenshots

- We can use Sikuli to compare two screenshots
  - First copy the image to clipboard
  - click(“image1.png")
    type("c", KeyModifier.CTRL)
    x = Env.getClipboard()
  - # path to the folder where your script is stored

    dir = os.path.dirname(getBundlePath())

     # capture a screenshot

    img = capture(SCREEN)

    # save the screenshot as image2

    shutil.move(img, os.path.join(dir, “image2.png"))

Compare Two Screenshots

- Then we can start comparing

  -
    ```
    f = Finder(“image2.png")
    f.find(x) #x is the image we already have
    if exists("x"):
        popup(“image found")
    print x;
    ```
- If same image is found, “image found” will popup

Use pre-captured image

- Sikuli allows inserting images that are pre-captured, so you don’t need to capture the same GUI components that appear repeatedly
- E.g. we captured the Firefox icon as firefox.png
- Every time we need to locate the icon in the screen:
    - find(“firefox.png”)
- No need to capture the icon every single time
- Place the file in the same folder with other files of the same scripts

Exercise #9 (10 mins)

- Use Sikuli insert image function
- Automate the following process using Estock app:
  - Buy 100 quantity of the first 3 stocks
  - Use insert image using the captured file for repeatedly appear GUI components

6 Sikuli: Batch Execution and OCR

Learning objective #6

- Call function in Sikuli script
- Combining several Sikuli scripts and run together
- Run Sikuli script by batch
- Use OCR to identify words

Environment Information

- Sikuli allows user to get the environment information of the computer that sikuli is running on
  - Operating system: Env.getOS()
  - Operating system version: Env.getOSVersion()
  - Sikuli version: Env.getSikuliVersion()

Run Sikuli from Command Line

- Sikuli can be run without the IDE but using command line
  - 1. Open command prompt
  - 2. [path to location of your sikuli]/runsikulix –r yourScriptName.sikuli
- The log message will appear in the command prompt

Run Sikuli by batch

- Sikuli can be run by batch
- Create a bat file or a cmd file includes:
  - @ECHO-OFF
  - CD %USERPROFILE%\Desktop\sikulix\
  - runsikulix -r combine.sikuli
  - PAUSE
- Double click or use command prompt to run the file

Exercise #10 (20 mins)

- A group of three
- Modify the script in exercise #5
- Divide the original script into three separated scripts
- Combine the script into one by function call
- Write the batch script to run it
- Perform the test

MySQL and Sikuli

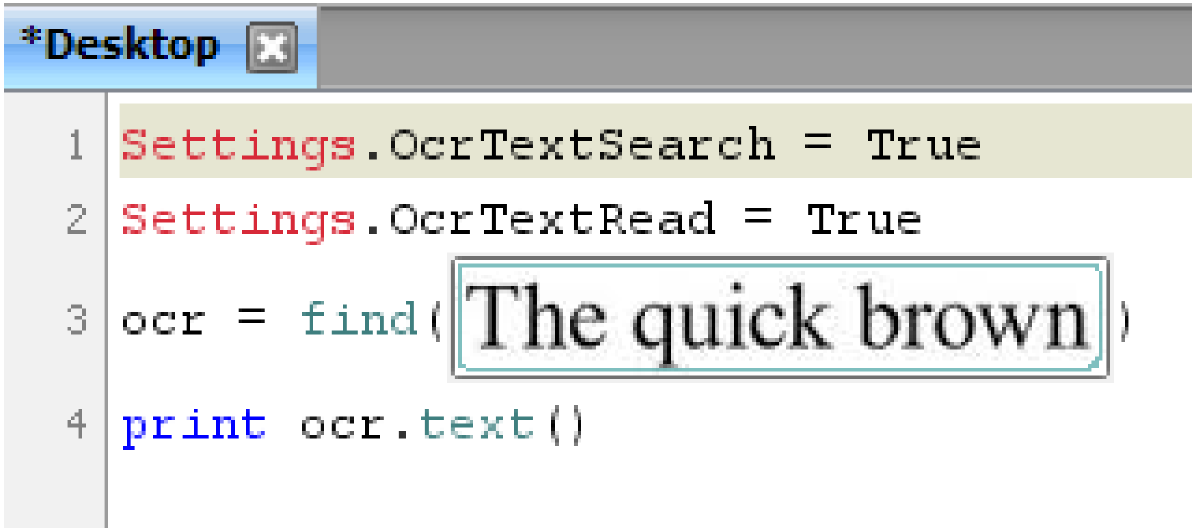
- Sikuli can be programmed to fetch data from MySQL database
- It makes the automation of testing much more efficient
- A sample code file included in the course file

OCR

- Sikuli can perform basic Optical Character Recognition (OCR)
- To use OCR in Sikuli,
  - Insert these two lines at the beginning:
    - Settings.OcrTextSearch = True
    - Settings.OcrTextRead = True
- Then you can assign the words you want Sikuli to recognize to a variable

OCR

- When the same line of character appears on screen, Sikuli can recognize and perform the action you desired
- For some occasions, you have to encode the translated text with ascii to display correctly
  - print ocr.text().encode(‘ascii’,’ignore’)



```
Settings.OcrTextSearch = True
Settings.OcrTextRead = True
ocr = find(The quick brown)
print ocr.text()
```

Exercise #11 (20 mins)

- Use Sikuli to operate a calculator
  - E.g. Program the calculator to 2+2
  - Then use OCR to check if the answer is 4
  - If correct, popup a successful message

Q&A session

- *Any questions?*

Thank you

- *Your feedback is important to us. Please fill in the evaluation form.*