Assignment 1

Name : Li Pei

Andrew ID : lip

1. Contrast an Android Project created with and without an Activity.

In most cases, an Android project would have an Activities. An Activity is an application component that provides a screen with which users can interact in order to do something, such as input some text, take a photo or send some voice. Each activity is given a window in which to draw its user interfaces. The window typically fills the screen, but may be smaller than the screen and float on top of their windows.

An application usually have more than one Activities, and typically one is the main activity, which is presented to the user when launching the application. Each time a new activity start, the previous will be stopped. But the stopped activity will be preserved in a stack. When you type back, it will pop the previous activity and resume it.

When an activity is stopped because a new activity starts, it is notified of this change in state through the activity's lifecycle callback methods. Due to a change in its state, there are several callback method that an activity might receive, and each callback offer you opportunity to do specific work.

The activity transitions into and out of different states is notified through various callback methods. All of the callback method are hooks that you can override to do appropriate work

when the state of your activity changes. The fundamental method for lifecycle is onCreate, onStart, onResume, onPause, onStop, onDestroy, onRestart.

However, sometimes you only need the app run as service and do not need to interact with user. In this case, the project do not need any activity and windows interface could be eliminate. This is called **Android project without Activity.**

Android without activity do not live on the lifecycle of callback because it did not interact with user. For this can of project, it can be start by calling the Context.startService() and stopped by Context.stopService(). It is impressive that only one Context.stopService() need to be called to stop the service, no matter how many call of start service. The service also could stop itself.

Compared with lifecycle of project with activity, a lifecycle for project without have less fundamental method of lifecycle. Only onCreate, onStart and onDestroy. And another difference is that, you could have more than one activity, but only one activity is running and other is stopped and preserved in stack. For service without activity, it only have one instance but can be started multiple times but stopped by only one stop call.

2. After watching video (Pranav Mistry @ MIT Labs http://www.youtube.com/watch?v=YrtANPtnhyg) what assumptions can you make about current mobile computing replacing desktop computing. Comment on current implementation(s) and what are your assumption about how mobile computing will continue to evolve.

Use the following references: ..Android 5.0 features ..Google Glass ..iPhone 5 ..iPad 3 ..Gaming Consoles ..TV and Autoapps.

This video is awesome! I have seen this video several years ago, but now when I saw this video again, this is still amazing and unbelievable. We have to salute to what he have done. From this video, we can see that mobile computing will replace desktop in more and more scenario, and this will be accelerated with the increasing computing ability and decreasing size and power consumption of mobile devices.

Why we can make the assumption that mobile computing will replace desktop computing? There is some reasons.

First of all, mobility. The mobility of device will offer more use scenario for device. For example, if you want to take a photo of a mountain and share with your friend instantly. If you use desktop, it is hard because you have to bring power supply and internet access for your desktop in wild. However, if you use the mobile device, everything will be easy because "mobility" of your device.

Secondly, the mobile device's computing ability is increasing drastically. In the video, we can see the mobile device could take photo, use web browser and edit picture and text without distinction as our PC, so the desktop with similar computing ability but far less mobility will fall out.

However, on the other side, the computing ability of desktop is too strong for some scenario. If you only want a device to store some data and then connected into a Internet of Things, apply every device of object with desktop is a waste of computing ability.

Speaking of Internet of Things, you could see that in the video, everything could be used as interface and interact with user. Placing desktop everywhere to handle this issue is not wise because the high expense. However, when you could do mobile computing and bring computing ability with you, it could be easy to apply. And this is only one application of

Things of Internet. If you could offer everything with computing ability, you could access everything in the world and interact with them, that is where mobile computing show off its ability.

Current mobile applications are evolving really fast. In 18641, let us start with **Android 5.0 Lollipop**. Lollipop comes with several new features, like Material Design, new notifications, device sharing, etc. In these features, OK Google and Device sharing and Tap & Pay is something brand-new for Android and could compete with other mobile device system.

OK Google is a voice support for user, which combines machine learning, AI and signal processing technology to interact with user. With the support, user could not only get quick access some application in your device, but also go social like asking weather, get suggestion for lunch. This is like an interface between user and the whole world, because you could access the information of world by this application, instead of do it yourself.

Tap & Pay is a similar interface to interact with world, but it is specified on help you easily pay. With one tap you could pay your money from your device, instead of your card or cash.

Device Sharing is not new. In PC time it allow multiple user on one device with different user information and context. But Lollipop's sharing is more attracting because Android could be applied in many devices and embedded as operating system. People could log in all these devices based on Android with their user information and it makes the mobile computing more powerful, because you could implement computing on any kind of devices with all your information context.

Smart TV is something new, because traditional TV without interaction is out of date and cannot satisfy the demand of user. The change of TV is that it could access Internet and offer content of what you want. Some TV device could even interact with user like video game device, which offer more using scenario. Google's **AutoApps** is one of the application on TV

devices. It is a central hub for all the Tasker plugins that help you automate your life. You could add app with it on your smart TV and it will make move according to your pre-offered instructions. It makes it possible that you could see everything as soon as you wanted, automatically. It will ease your life tremendously, and you do not need to passively receive information from the world, you could ask for what you want to know.

Google Glass is something recent and something really big. When we talk about virtual reality and augmented reality, we used to refer some huge devices and without connection to internet, but Google Glass change it, drastically. The Google Glass is small, looks like regular glasses, but it comes with amazing computing ability and could connect with network. It offered similar, or better realization of offered video, because it is more simple. The Google Glass offered a new interface to interact with world, with one device, you could get the information of what you see, and in the future when everything is connected, it may act as the interface for you access the world instead of your eye.

iPhone 5 and iPad 3 are something similar, only the size of screen make them different. It is said that at first place, iPad with larger screen and higher power is designed to replace the desktop, and iPhone is designed to take over the smartphone market. Apple is good at reinvent the thing and make breakthrough, and iPhone 5 and iPad 3 is awesome among their competitors,however, with the development of mobile computing, battery and screen, the function of iPhone and iPad are kind of converging. And I think one day iPhone and iPad 3 will decay, but there is one thing will last long, Siri. Siri is mature in iPhone 5 and iPad 3, it is really a smart and widely used digital assistant, I mean, really smart and in some cases you cannot distinguish it with a real person. Like what is done in the video, if you could get mobile computing anywhere, it would be great. And Siri make it in one way that anywhere using iOS could help you get access to the information of world.

Game console with mobile computing is new to me when the first time I say Oculus and Microsoft's HoloLens. In my memory, gaming console is something you connected with TV and interact with the figure in program or people in this game. Now with the mobile computing, VR and augmented reality is something real deal and you could interact with the real world! It enlarge the game and lead it to infinity because no single game could contain the content of whole world. With the power of mobile computing, now the whole world is configured as your game.

After considering this current implementation of mobile computing, how **mobile computing will evolve in the future**?

From my perspective, the future mobile computing will be pervasive, as shown in the video, computing can be anywhere, based on computing devices with you or embedded in other devices.

Secondly, future mobile computing will pay attention on the way you interact with the reality. With the development of current social network, people are get used to interact with other people in social network, but how about share the network with objects? I think like Google Glass, other computing devices would evolve to a scenario that everything could interact you and satisfy your request of its information, or itself.

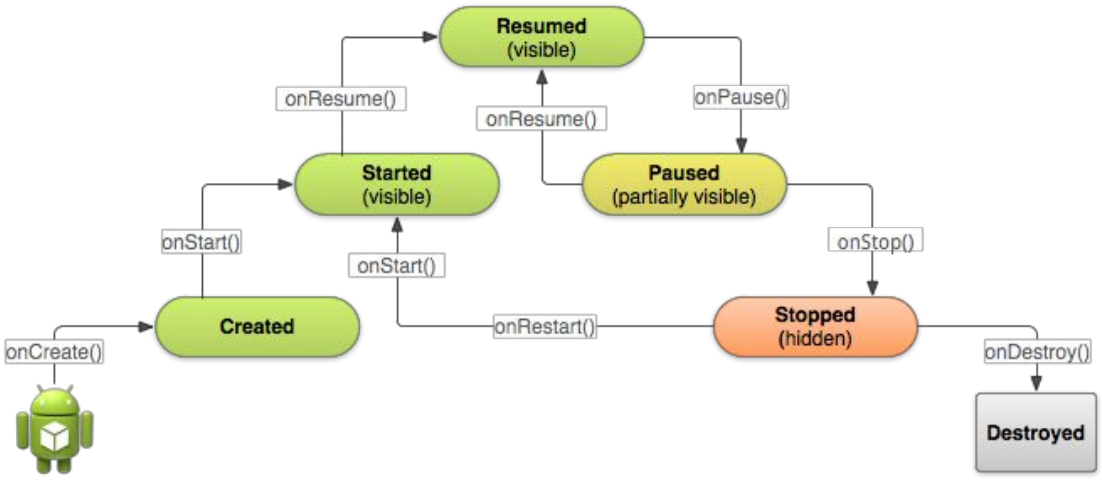
The last is a bold idea, that mobile computing could be separated with screen, like the video shown, you get UI on any surface, and maybe lastly separated with device, because computing could be distributed by the device around you and you do not need any computing device carried but accomplish your mobile computing request.

3. Describe the execution lifecycle for an Android App when running on a physical Android Device.

To describe lifecycle for an Android App, we have to understand the state of Activity in Android. An activity in Android can exist in four states as described below.

1. Active and Running state. This is a state when an activity is in the front and has focus in it. It is completely visible and active to the user.
2. Pause state. In paused state, the activity is partially visible to the user, but not active and lost focus.
3. Stopped state. This is when the Activity is no longer visible in the screen. Another activity is on top of it and completely obscures its view. In this state also the activity is alive and preserves its state, but more likely to be killed by the system to free resources whenever necessary.
4. Destroyed/Dead state. An Activity is said to be dead or destopyed when it no longer exists in the memory.

Based on the Activity state, we could draw an android lifecycle as diagram below.



The above Lifecycle of Android Activity diagram can be explained as follows:

When we launch an Activity in Android, it first calls the onCreate() method. This is where we do User Interface creation and initialization of data elements. This method is provided with a Bundle object as parameter to restore the UI state.

onStart() method is called before the Activity is being visible to the User. Remember that Activity is still not Active.

With the onResume() method, the Activity become visible and Active for the user to interact with. The Activity will be at the top of the Activity stack at this point. Now the Activity is in running /active state and is able to receive user inputs.

In the Active state, onPause() method will be called when the system is about to resume another Activity on top of this one or when the user is about to navigate to some other other parts of the system. It is the last guaranteed call to a method before the Activity can get killed by the system. That is, there's a possibility that your activity may be killed by the system at the paused state without executing any further method calls. Therefore it is important to save the user interface configuration and critical data at this method.

On a physical device, the user's action could trigger the Activity. By default, an Activity can remain in the paused state if user pressed the home button. Another activity or notification which is on top of it doe not completely obscures the visibility of underlying Activity. The device goes to sleep.

There are two possibility for an Activity under paused state:

1. The User resumes the Activity by closing the new Activity or notification and the paused Activity gets Active/Running by calling onResume() method.

2. It gets killed by the system under extremely low memory conditions. In this case there will be no further method calls before the destruction of the Activity and it needs to be re-run from the beginning by calling onCreate() and restoring the previous configuration from bundle object.

An activity under stopped state also has three different scenarios to happen:

1. System kills it to free resources. An activity under stopped state is more likely to be killed by system than one in the paused state. It needs to start the cycle again with onCreate().
2. It get restarted by calling onRestart(), onStart() and onResume() methods in the order if the user navigates back to the Activity again. In this case, the UI is intact and no need to be restored.
3. onDestroy() method is called and the Activity is destroyed. This is the final method we can call before the Activity is destroyed. This occurs either because the Activity is finishing the operation or the system is temporarily destroying it to save place.

Back to see the diagram before, we can see there are three lifecycle loops exist for every Activity and are defined by those callback methods. They are:

1. Entire Lifetime: This is the lifetime between the first cal to the onCreate() and the final call to onDestroy() method. We create all global resources such as screen layout, global variables etc in onCreate() and release all resources with onDestroy() call.
2. Visible Lifetime. It is lifetime of an Activity between onStart() and onStop() method calls. In this the Activity is visible to the user and he may or may not be able to interact with it. During the visible lifetime, an Activity maintains its state intact.

3. Foreground Lifetime. Foreground lifetime starts with onResume() and ends with onPause() method calls. During this, the Activity is completely visible to the user and is on top of all other Activities so that user can interact with it.