UML Activity Diagram

Introduction:

To understand UML Activity Diagrams, we first need to understand what UML Diagram means. UML stands for Unified Modelling Language. It is a standardized set or a collection of diagrams which helps the software developers and software architects to understand the flow of the software. In other words, UML Diagrams are diagrams which depict how the software system is going to function. UML Diagrams are divided into three types:

a. Structure Diagrams
b. Interaction Diagrams,
c. Behavior Diagrams.

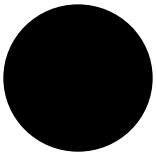
What is an UML Activity Diagram:

An UML Activity Diagram is a behavioral diagram out of the above mentioned three types of UML diagrams. Now, how are they different from UML diagrams? So activity depicts what action is going to take place in the process. It is pictorial representation of how the software system is going to function. At the time of execution, the software system must function according to the required flow, so according to it, activity diagrams depict it with as forward and reverse engineering process / actions.

Symbols and Components of UML Activity Diagram:

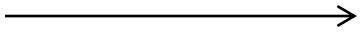
Beforehand, one must have knowledge of how to draw the UML activity diagrams and for it following components and symbols are needed.

1. Starting State: The initial state which is yet to be used or modified in the activity. Activity Diagrams start from this step. Also known as the entry state. Start Node is the starting point of any activity. It is depicted as:

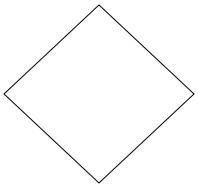


2. Action State: A step in which the users or a software performs a certain task. It represents an action which is going to take place at this stage of the software system. Generally depicted with rounded edged rectangle.
   It is depicted as:

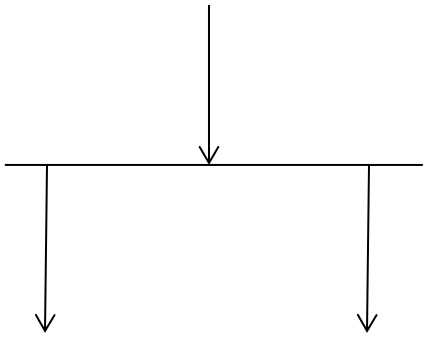
3. Control Flow: Connectors between two states or two actions to depict the flow. Shows the sequence of execution. Also known as paths. One action state can have multiple control flows input and also output to another action state. A single headed arrow is used to depict the control flow.
   It is depicted as:



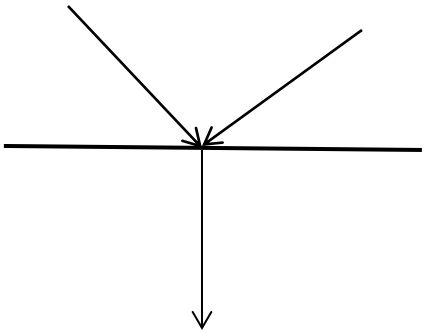
4. Decision Node: A conditional node or a decisional node is one where there are multiple options available. Or there are two or more conditions which can be considered at the point of the software system.
   Ex : There's an ice cream shop. A person enters into that shop and wants to buy one of the many options available. Now, if he chooses, Vanilla flavor, then server 1 has to give it to him, else he chooses other flavor, then others have to serve it to him. So this is the condition in the activity diagram. It is drawn as a diamond shape with multiple inputs and outputs.
   It is depicted as:



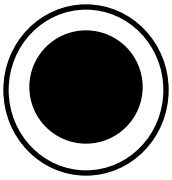
5. Fork: A point from where two concurrent or parallel processes are executed or run or processed. It generally includes a single input, but may or may not get one output.
   Ex: There's one ice cream shop. A person tends to buy ice cream for himself and his friend. Now, both of them buy same vanilla flavor, but one of them wants choco-dip and other wants to have fruities to be put upon it. So, here the input is same, that is of vanilla ice cream but the output results are different. So, this is a perfect example of fork.
   It is depicted as:



6. Join: A join is one where, two results of concurrent activities add and form a single result. In join, there is more than one input, but only one output is obtained. Two results are obtained from two activities and one result it obtained.
   Ex: There is an requirement of sandwich. But first we need to have tomatoes and spinach for it.
   It is depicted as:



7. End State: This is the last stage of the UML activity diagram. This is where the activity ends in a software system ends.
   It is depicted as:



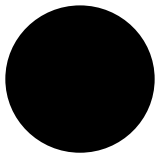
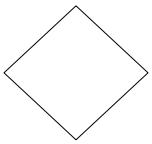
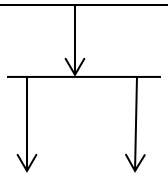
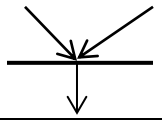
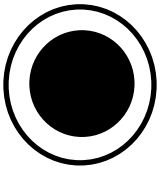
Advantages or Benefits of UML activity diagram:

1. Complex stage or steps in a software system can be explained easily diagrammatically.
2. Dynamic modelling of a software system.
3. Each and every activity flow in the system can be explained as it is.
4. Methods, functions and operations can be explained in details.
5. Business process and flows can be depicted easily.
6. Simplified view, though complex system.
7. Business requirement analysis.
8. Understanding of system requirements is explained in a lucid and simple manner.
9. Workflow of the user and the system and user with the system is explained in detail.

Disadvantages of UML Activity Diagrams:

1. The only drawback in the UML Activity Diagram is the messages or the communications between two components or the user cannot be shown.

The symbols used in the UML activity diagram is explained in the below table:

Sr. No	Name	Symbol
1.	Start Node	
2.	Action State	
3.	Control Flow	
4.	Decision Node	
5.	Fork	
6.	Join	
7.	End State	

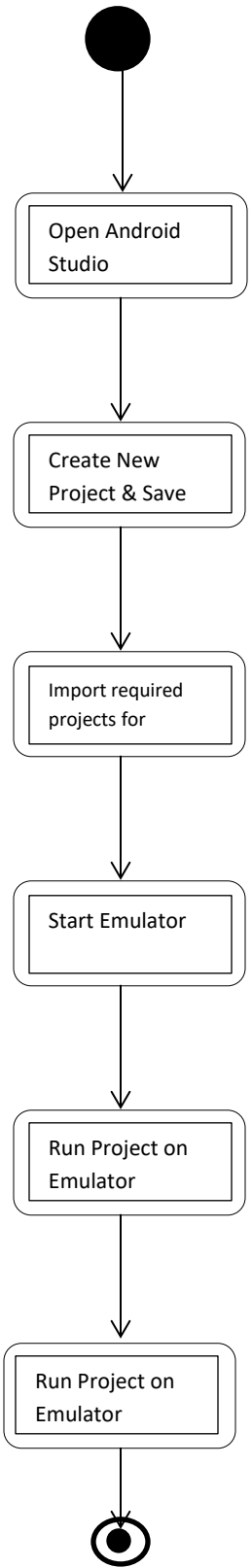
Example of how to make an UML activity diagram:

We'll consider an example of developing software in Android Studio

1. Open Android Studio

2. Create New Project & Save the project under an unique name
3. Import the new packages which are needed for the project
4. Start the emulator
5. Run the project on the emulator
6. Exit Project

UML Activity Diagram:



Conclusion:

So, in all we can say that UML Activity Diagrams are necessary during and before development of any software system. It is very useful in documenting and depicting visualizing the exact process and steps involved in the developmental process. All the complex stages can be displayed very easily. The person who knows the notations correctly can easily draw the UML activity diagrams. The most important part of using these diagrams is anyone can draw them according to the flow and at almost every step of the software system development life cycle. There are many softwares available online which are helpful in drawing the UML Activity Diagrams like SmartDraw, etc.