

# CSc 317

# The Stack

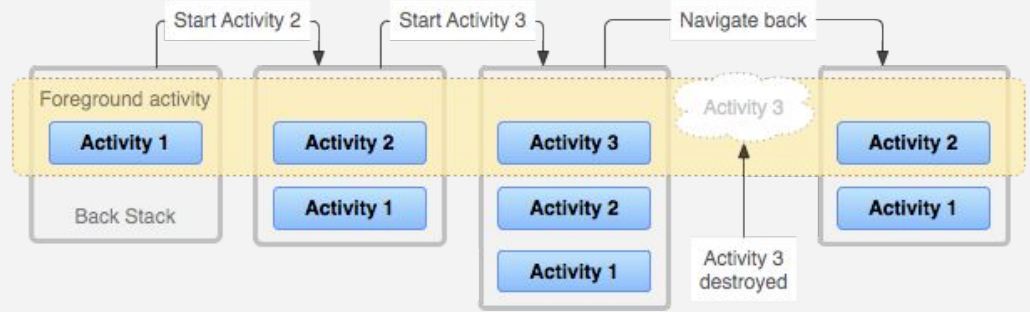
Benjamin Dicken



# Announcements

- Exam 1
  - Variety of question types
    - MC, code/resource writing, etc
  - Around 7 questions some multiple parts
  - Resources for studying:
    - Slides, example code, quizzes, readings, PA code

# The back-stack



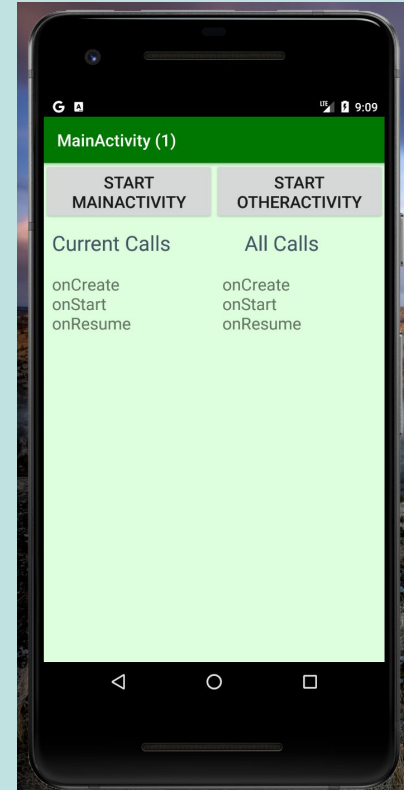
- A **task** is a stack of Activities (or Fragments)
- As activities are started, they are arranged into a stack (the back-stack) within the task
- When activities are ended (think, the back-button), they are popped off of the stack

# The back-stack

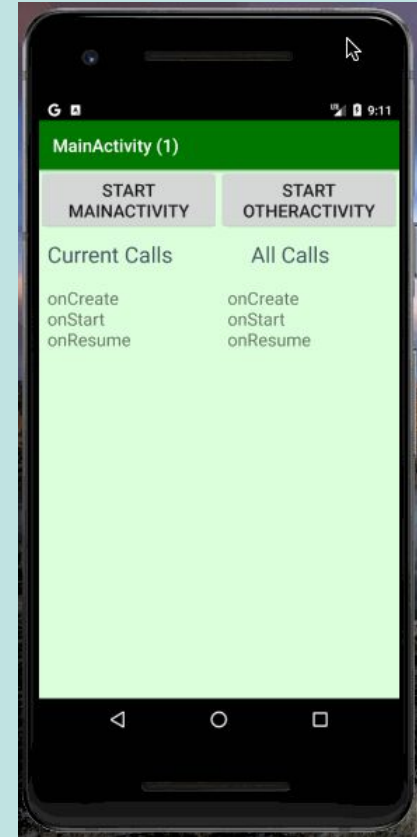
- Android can support running multiple tasks simultaneously
- Can bring tasks into the foreground, put into the background



# Download BackStackTest app



# Click through the stack

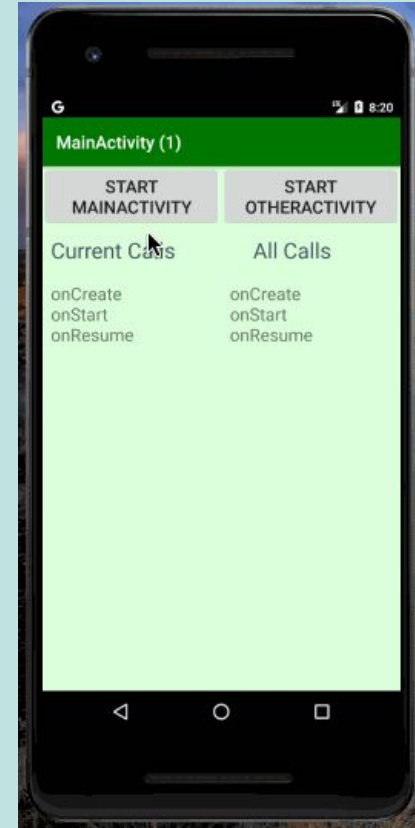


# Change default stack behavior

- `intent.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);`
  - If task is already on BackStack, does not create new activity, but brings existing one to foreground. `onNewIntent` called.
- `intent.addFlags(Intent.FLAG_ACTIVITY_NEW_(TASK/DOCUMENT));`
  - Starts the activity in a NEW task, unless it is already running in another task. `onNewIntent` called.
- `intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);`
  - If task is already on BackStack, destroys other activities between top of stack and it. `onNewIntent` called.

# Change the code to mimic this behavior

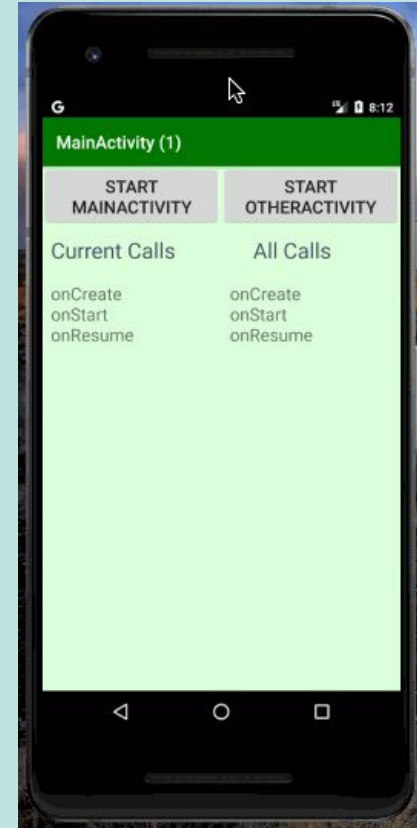
- When opening **MainActivity**, if it is on the top of the stack, should not create a new instance





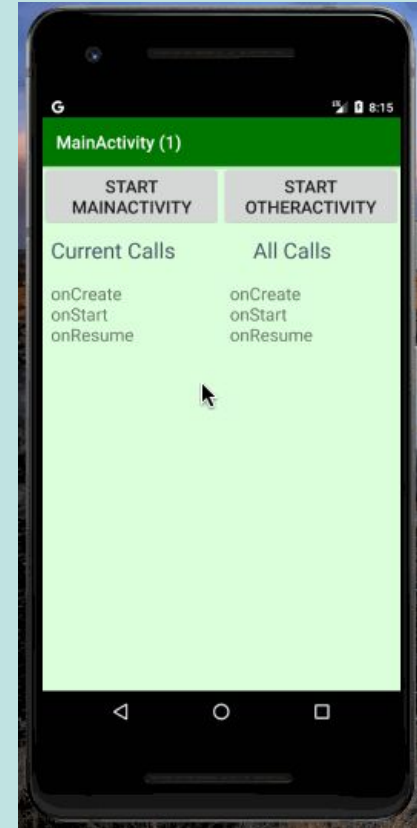
# Change the code to mimic this behavior

- When opening **OtherActivity** for the first time, should start up a new task, not just a new activity



# Change the code to mimic this behavior

- When re-opening **MainActivity**, should clear all activities between the top and the previous active **MainActivity**



# Track the onNewIntent Calls

- Change the app to that the count of calls to **onNewIntent** are tracked, per activity



# Think of a practical use-case for:

- `intent.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);`
  - If task is already on BackStack, does not create new activity, but brings existing one to foreground. `onNewIntent` called.

<https://developer.android.com/guide/components/activities/tasks-and-back-stack>

# Think of a practical use-case for:

- `intent.addFlags(Intent.FLAG_ACTIVITY_NEW_(TASK/DOCUMENT));`
  - Starts the activity in a NEW task, unless it is already running in another task. `onNewIntent` called.

<https://developer.android.com/guide/components/activities/tasks-and-back-stack>

# Think of a practical use-case for:

- `intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);`
  - If task is already on BackStack, destroys other activities between top of stack and it. `onNewIntent` called.

<https://developer.android.com/guide/components/activities/tasks-and-back-stack>

# Exam Review

If we have time, a few exam review questions for you!

For these, might want to put away your laptop to simulate exam environment

# Practice Question 1

Describe the difference between an implicit intent and an explicit intent. In this description, provide a clear example of each intent type. Your answer should be one paragraph.



# Practice Question 2

Write a snippet of an XML layout file that would display the following:

- One button that has a blue background, yellow text
- When clicked, calls a function named, **calculateResult**
- Is 100dp tall and fills the entire width of the parent

## Practice Question 3

Say that we have an activity that has a resource-intensive 3d animation on it. When the user navigates away from this activity, we want to pause / remove the animation from the view to save resources, and reload it when the user returns.

What activity lifecycle function should we load / start the animation in?

What function should we pause / stop the animation in?

Why?

# Practice Question 4

Write a complete java method named **displayData** that does the following:

- Creates an ArrayAdapter for adapting the data in the class variable **groceryList** which you can expect is an ArrayList of Strings
- Finds the ListView with id **groceries\_list\_view**
- Displays the strings on this listview

# Practice Question 5

Why does the shared preferences API exist? What not just save everything into text files or a more feature-rich database?