Mobile Applications
CSCI 448
Lecture 10

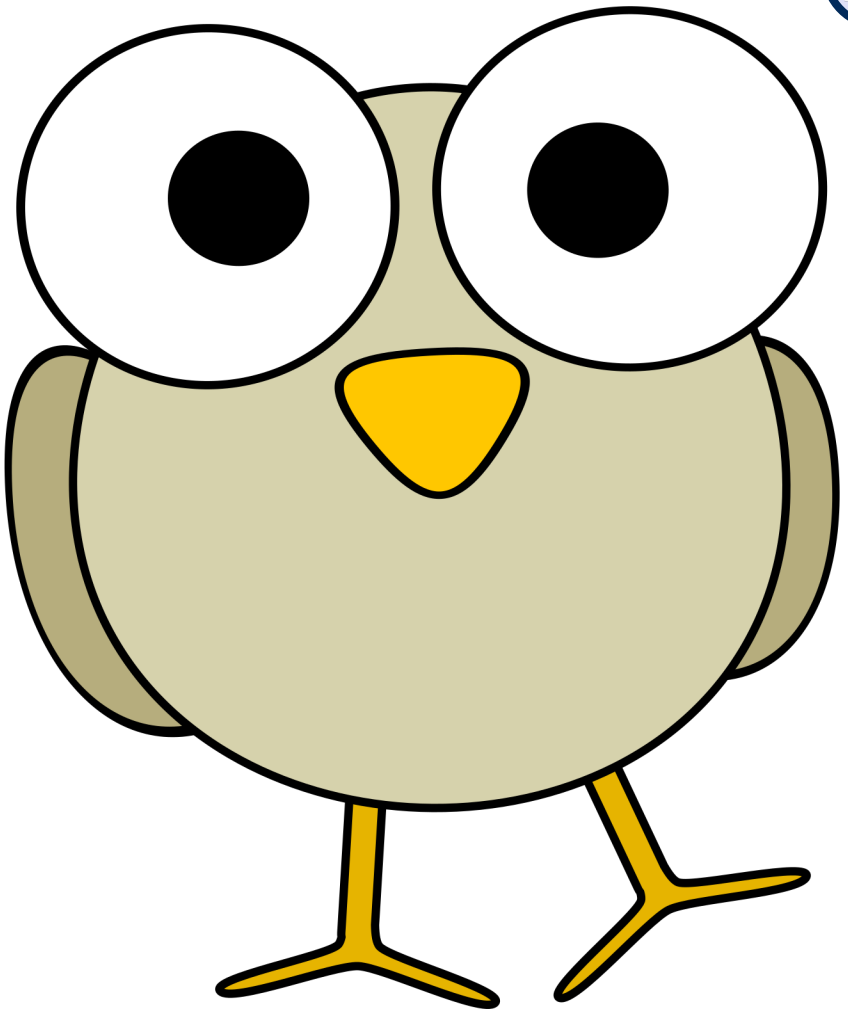
Processes, Tasks,

And the Bundle

Previously in CSCI 448

- Save UI State
  - Configuration Changes: ViewModel
    - Lifecycle Aware – persists beyond the life of an Activity
- Decorator, Singleton, & Factory Method Patterns

Questions?



Learning Outcomes For Today

- Explain the difference between a Process and a Task

- Create an app that can save state across destruction and recreation via a ViewModel and a Saved Instance State (Bundle)

On Tap For Today

- Tasks & Processes
- The Bundle

On Tap For Today

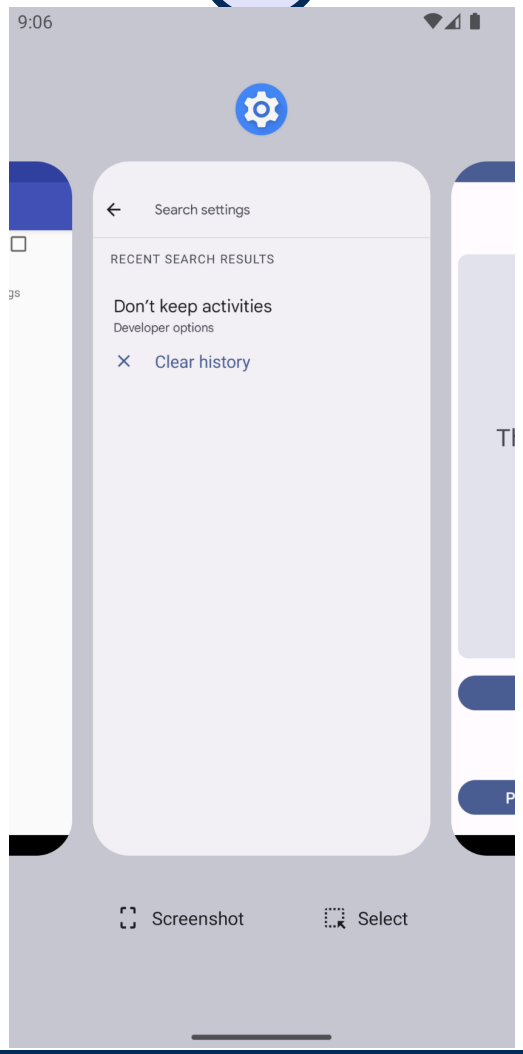
- Tasks & Processes

- The Bundle

Application Process

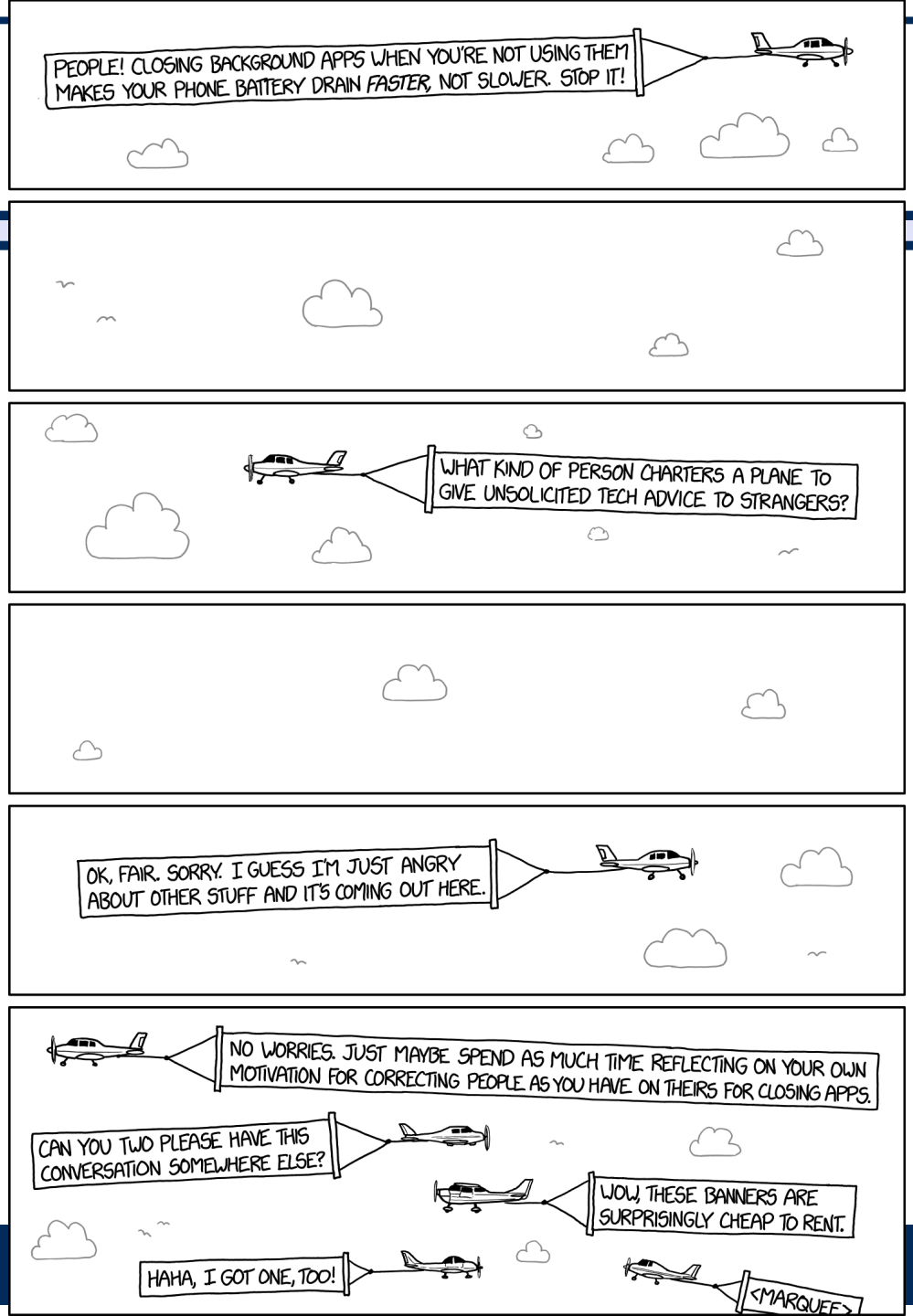
- Application's footprint
  - Contains a single thread to execute UI tasks and a piece of memory to store objects
  - Contains all Activity instances
    - Stores state data at runtime needed for application

Recents View



Processes VS Tasks

- Tasks contain only references to activities, usually across different processes
- Processes contain all running code and objects for a single application
- How does data pass between Processes then?
  - Use an `Intent`. Stay tuned! (Next month)


PEOPLE! CLOSING BACKGROUND APPS WHEN YOU'RE NOT USING THEM MAKES YOUR PHONE BATTERY DRAIN FASTER, NOT SLOWER. STOP IT!
WHAT KIND OF PERSON CHARTERS A PLANE TO GIVE UNSOLICITED TECH ADVICE TO STRANGERS?
OK, FAIR. SORRY. I GUESS I'M JUST ANGRY ABOUT OTHER STUFF AND IT'S COMING OUT HERE.
NO WORRIES. JUST MAYBE SPEND AS MUCH TIME REFLECTING ON YOUR OWN MOTIVATION FOR CORRECTING PEOPLE AS YOU HAVE ON THEIRS FOR CLOSING APPS.
CAN YOU TWO PLEASE HAVE THIS CONVERSATION SOMEWHERE ELSE?
WOW, THESE BANNERS ARE SURPRISINGLY CHEAP TO RENT.
HAHA, I GOT ONE, TOO!
<MARQUEE>

Task Killer Apps

- Android OS already contains a “Task Killer”
- Actually a Process Killer

Activity Prioritization

- Activities in a Running or Paused state have higher priority
- Activities in a Stopped state are lower priority and killable to free up resources
  - Activities and ViewModels get removed from memory

Process Death

- What happens on process death?
  - We are at the top of the Task stack in Process B
  - Process A gets killed
  - We return to the activity in Process A
  - How is state preserved?

On Tap For Today

- ViewModel
- Bundle
- Practice

Process Death

- Activity that is killed can utilize temporary short-term storage beyond Activity instances lifespan
  - Store data in a saved instance state
    - This is the `Bundle`
  - Map of key-value pairs
    - Values need to be `Parceable`

Saving Data Across Process Death

- Save state data out to Bundle when shutting down

- Can extract that data from the associated Bundle (if it exists) during creation

Using the Bundle

"onSaveInstanceState()" is called after onStop()

```
override fun onSaveInstanceState(outState: Bundle) {
  outState.putInt(KEY_INDEX, currentIndex)
  super.onSaveInstanceState(outState)
}
```

The "key" is any string

This value is our question number

Retrieve the data from the bundle when needed in onCreate(Bundle?)

```
currentIndex = savedInstanceState?.getInt(KEY_INDEX, 0) ?: 0
```

may be null

Alternate Lifecycle Flow

"onSaveInstanceState()" is called after onStop()

```
override fun onSaveInstanceState(outState: Bundle) {
  outState.putInt(KEY_INDEX, currentIndex)
  super.onSaveInstanceState(outState)
}
```

"onRestoreInstanceState()" is called after onCreate() when initially created

not null!

```
override fun onRestoreInstanceState(savedInstanceState: Bundle) {
  super.onRestoreInstanceState(savedInstanceState)
  currentIndex = savedInstanceState.getInt(KEY_INDEX, 0)
}
```

Persisting UI State

- State is lost from two scenarios when activity is destroyed & recreated

1. Configuration changes
   - Use `ViewModel`

2. Process death
   - Use `Bundle`

To Do For Next Time

- A1 due ~~tomorrow~~ Wednesday
- Final Project Storyboards due Friday
- Be working on Lab03
  - Lab03A is biggest step
  - Due Tue Feb 14