INTRODUCTION

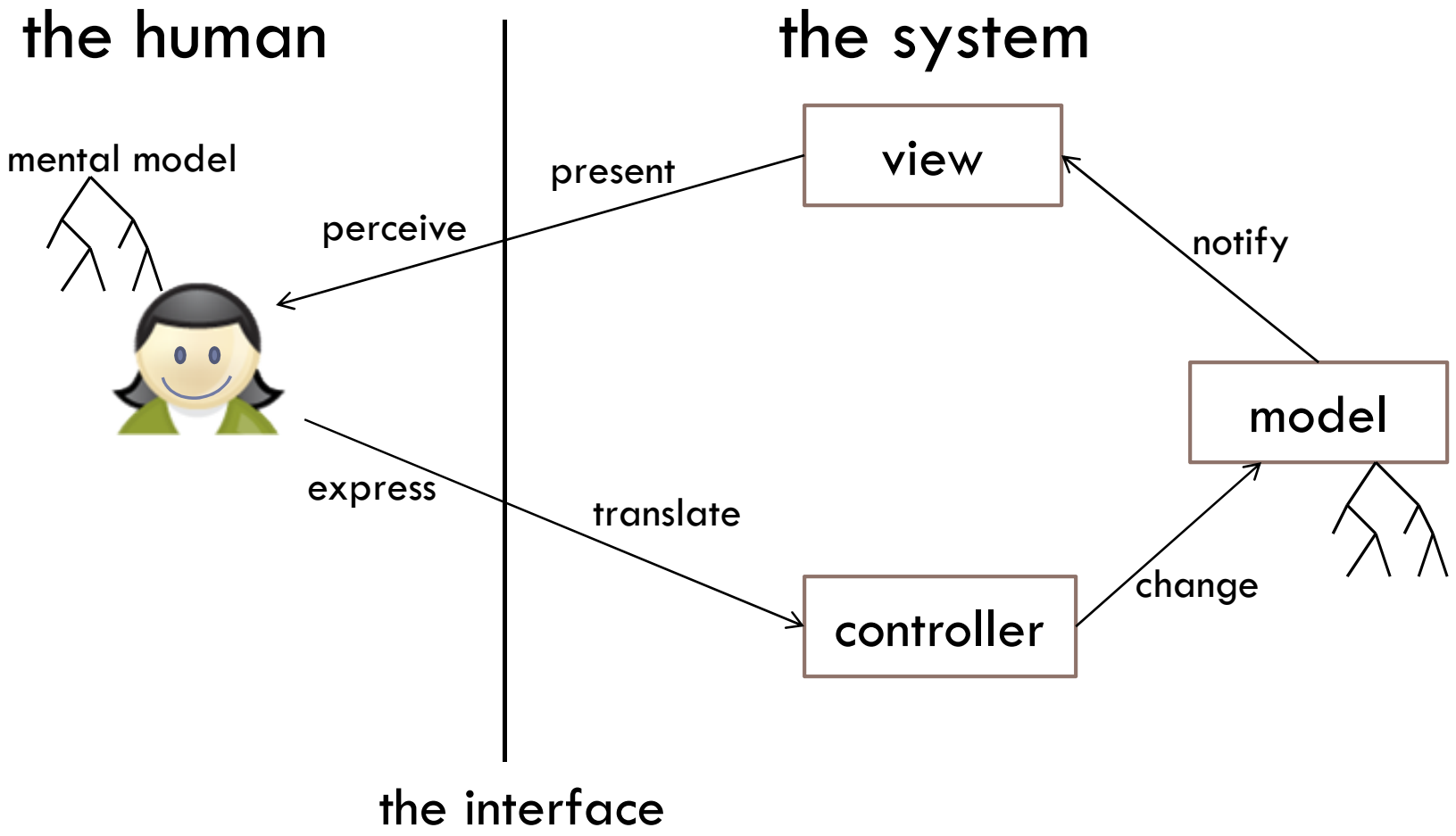
CMPT 381

Outline

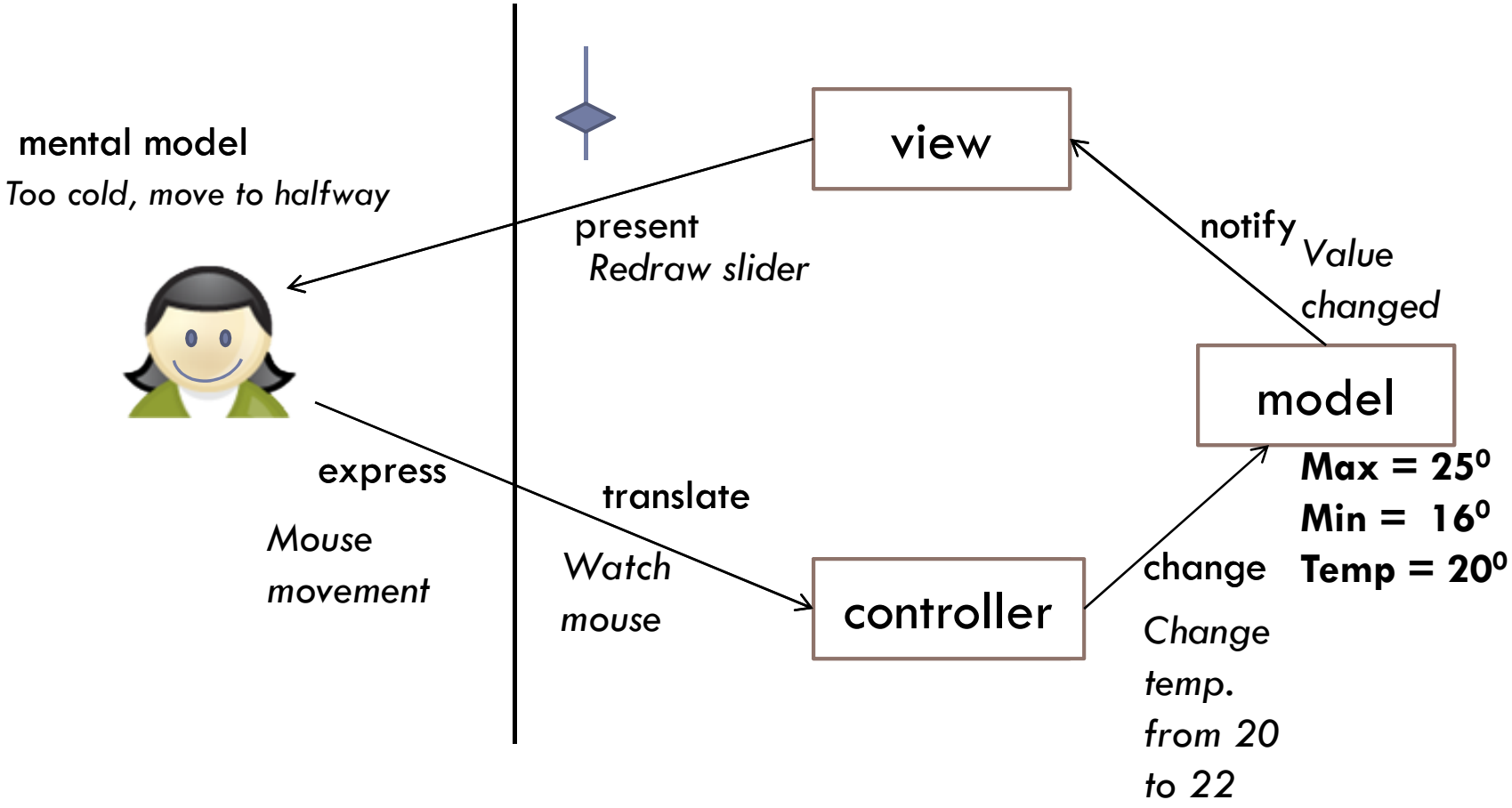
- The interaction cycle
- Parts of a Graphical User Interface
- A layered approach to supporting interaction
- Development basics

The Interaction Cycle

Architecture of interactive systems



Example: software thermostat



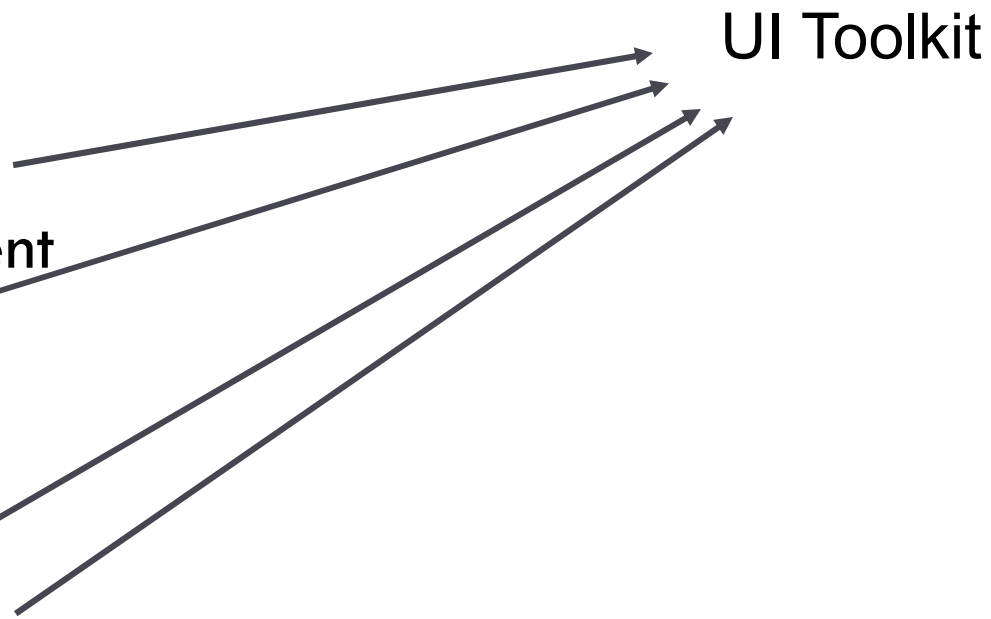
The GUI

Parts of a GUI

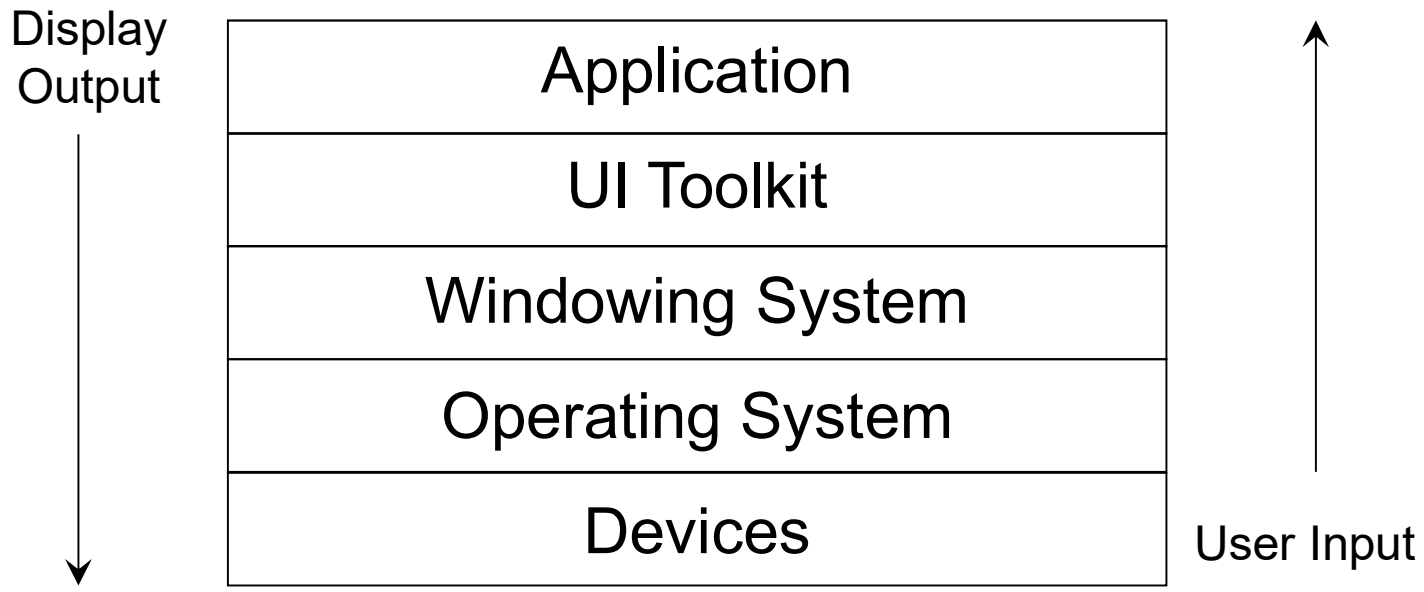
- Input side:
  - Input devices
  - Device drivers
  - Device interfaces
  - Window management
  - Input events
- Output side:
  - Interface widgets
  - Graphics & sound
  - Display/output devices

Parts of a GUI

- Input side:
  - Input devices
  - Device drivers
  - Device interfaces
  - Window management
  - Input events
- Output side:
  - Interface widgets
  - Graphics & sound
  - Display/output devices



Layered Approach



Layered Approach

- Device layer
  - Hardware level for input and output devices
  - Device drivers communicate with the OS
- OS layer
  - Turns input signals into events and sends to applications
  - Provides graphics/sound API
    - low-level machine-optimized drawing routines
    - e.g. Windows API, Macintosh Toolbox, X Intrinsics

Layered Approach

- Windowing system layer
  - puts applications in separate windows
  - handles resize, close, and iconify events
  - enforces a presentation style
- UI Toolkit layer
  - provides a set of high-level interface components
    - in Java, AWT (old) or FX (new)
    - in Android, no specific name (part of Android SDK)
  - handles input events
  - provides support for widget layout and graphics

Layered Approach

- Application layer
  - decides what to do with user input
  - manipulates the UI widgets
  - controls custom repainting of the screen
- Programmer only has to deal with the application and toolkit layers
  - (you can also talk to the window manager)

Layered Approach

- Layers are similar for all platforms (desktop, mobile, wearable, wall display)
  - The specifics capabilities of the UI toolkit will be different, but the basic ideas are similar

Window System

- Controls mouse, keyboard, monitor settings
- Provides windows for applications
  - Keeps a map of what application is where
- Dispatches input to different windows
  - Uses the map to decide who to inform

Window Manager

- Controls how windows look and act
  - Window decoration, Title bar
  - Window switching, virtual desktops
  - (Note that a 'Desktop' is not the same as the WM; it is an application)
- Microsoft Windows:
  - Combines OS, Window System, Window Manager, and Desktop

Examples

- 3D Window Managers
  - Metisse
  - http://insitu.lri.fr/~roussel/videos/metisse/metisse/metisse.mov
  - https://www.youtube.com/watch?v=P09WLS2a1H0
- Material Design UI toolkit
  - http://developer.android.com/design/material/index.html

Programming the UI

Development Environments

- JavaFX
  - Java 8 SDK, NetBeans IDE
    - www.oracle.com/technetwork/java/javase/downloads/
- Android
  - Java 7 SDK, Android Studio IDE
    - developer.android.com/sdk/

Hello World in JavaFX

```
import ...

public class HelloWorld extends Application {

 public void start(Stage primaryStage) {
        Button btn = new Button("Say Hello ");
        btn.setOnAction((ActionEvent event) -> {
            System.out.println("Hello!");
        };

        StackPane root = new StackPane();
        root.getChildren().add(btn);

        Scene scene = new Scene(root, 300, 250);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

 public static void main(String[] args) {
        launch(args);
    }
}
```

Hello World in Android (1)

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
    android:layout_height="match_parent"
   android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
   tools:context=".HelloWorldActivity">

    <TextView android:text="@string/hello_world"
   android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

Hello World in Android (2)

```
import ...

public class HelloWorldActivity extends Activity {

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hello_world);
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_hello_world, menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

Hello World in Android (3)

Plus many more files:

app/src/main/res/layout/activity_my.xml
app/src/res/AndroidManifest.xml
app/build.gradle
/res/drawable-hdpi/
/res/layout/
/res/values/

Hello World in Tcl/Tk

```
pack [button .b -text "Say Hello" -command "puts Hello"]
```

Homework

- Work with your environments
  - Hello-world assignment due this Friday
- Online tutorials, e-books from U of S library
- Textbook:
  - Chapter 1 (overview)
  - Chapter 2 (graphics)
  - Chapter 4 (widgets)