CSc 317
Animation

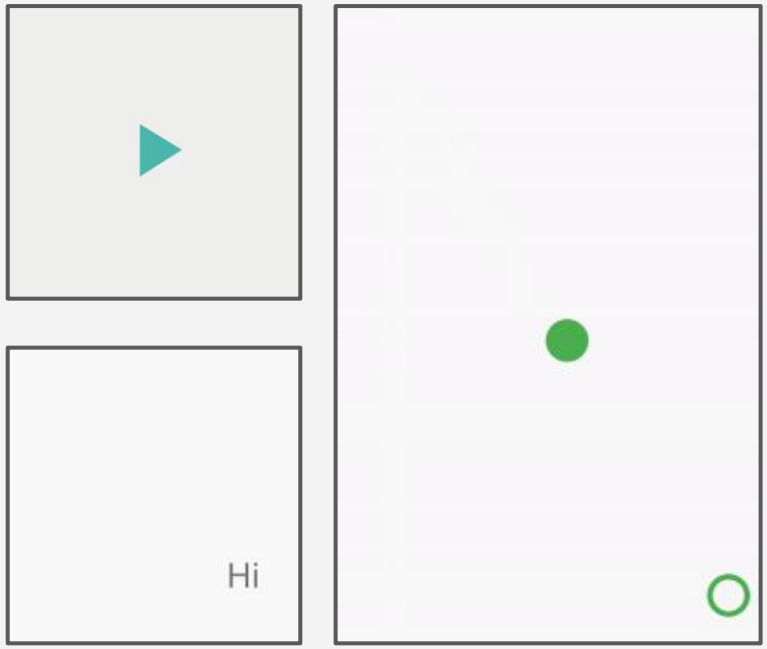
Benjamin Dicken

Announcements

- Exam 2 grades published
- Quiz 8 grades published
- The remaining quizzes
- Final Project Documents

Animation

- Animate drawable resources with **AnimateDrawable** class
- Animate UI elements with **ValueAnimator** or **ObjectAnimator**
- Physics-based animation with **Spring** and **Fling** animation
- Game engines



Animation

Other than for games, animation can be useful for

- Creating a “fancier” UI experience
- Animating views and images to their locations (slide, fade, etc), rather than just having them suddenly appear
- Loading bars
- Drag-and-drop interfaces

ObjectAnimator - one value

```
// must have setTranslationY and getTranslationY
TextView tv = findViewById(R.id.text_to_move_1);

ObjectAnimator animatorY = ObjectAnimator.ofFloat(tv, "translationY", 1000);

animatorY.setDuration(1000); // Milliseconds
animatorY.start();
```

ObjectAnimator - multiple values

```
// must have (set/get)TranslationX and (set/get)TranslationY
TextView tv = findViewById(R.id.text_to_move_1);

PropertyValuesHolder pX = PropertyValuesHolder.ofFloat("translationX", 500);
PropertyValuesHolder pY = PropertyValuesHolder.ofFloat("translationY", 0);

ObjectAnimator animatorXY = ObjectAnimator.ofPropertyValuesHolder(tv, pX, pY);

animatorXY.setDuration(1000); // Milliseconds
animatorXY.start();
```

Move a word

- Use an **ObjectAnimator** to move the word form one side of the screen to another

```
TextView tv = findViewById(R.id.???);
ObjectAnimator animatorY =
        ObjectAnimator.ofFloat(tv, "???", ???);
animatorY.setDuration(???);
animatorY.start();
```

Move a word

- Use an **ObjectAnimator** to move the word on both X and Y axis
- From top-left to bottom-right

```
TextView tv = findViewById(R.id.???);
PropertyValuesHolder pX =
    PropertyValuesHolder.ofFloat("???", ???);
PropertyValuesHolder pY = . . .
ObjectAnimator animatorXY =
    ObjectAnimator.ofPropertyValuesHolder(tv, pX, pY);
animatorXY.setDuration(???);
animatorXY.start();
```

ObjectAnimator events

- Can write code to run at various events, such as:
  - **`onAnimationStart`** - Called when the animation starts
  - **`onAnimationEnd`** - Called when the animation ends
  - **`onAnimationRepeat`** - Called when the animation repeats itself
  - **`onAnimationCancel`** - Called when the animation is canceled

ObjectAnimator events

```
PropertyValuesHolder pX = PropertyValuesHolder.ofFloat("translationX", 500);
PropertyValuesHolder pY = PropertyValuesHolder.ofFloat("translationY", 0);
ObjectAnimator animatorXY = ObjectAnimator.ofPropertyValuesHolder(?, pX, pY);

// . . .

animatorXY.addListener(new AnimatorListenerAdapter() {
    // Can create instance variables if necessary
    public void onAnimationEnd(Animator animation) {
        // Can re-use animatorXY, or create new animator
        Final ObjectAnimator other = ObjectAnimator.ofPropertyValuesHolder(tv, . . .);
        other.setValues(. . .);
        other.start();
    }
});
```

Move word

- Use an **ObjectAnimator** to move the word from one corner, to another, and then back

```
animator.addListener(new AnimatorListenerAdapter() {
    // Can create instance variables if necessary
    public void onAnimationEnd(Animator animation) {
        // Can re-use an animator, or create new animator
        Final ObjectAnimator other =
                ObjectAnimator.ofPropertyValuesHolder(tv, ???);
        other.setValues(. . .);
        other.start();
    }
});
```

Move word in a square

- Use an **ObjectAnimator** to move the word from one corner, to another, and then back

```
animator.addListener(new AnimatorListenerAdapter() {
    // Can create instance variables if necessary
    public void onAnimationEnd(Animator animation) {
        // Can re-use an animator, or create new animator
        Final ObjectAnimator other =
                ObjectAnimator.ofPropertyValuesHolder(tv, ???);
        other.setValues(. . .);
        other.start();
    }
});
```