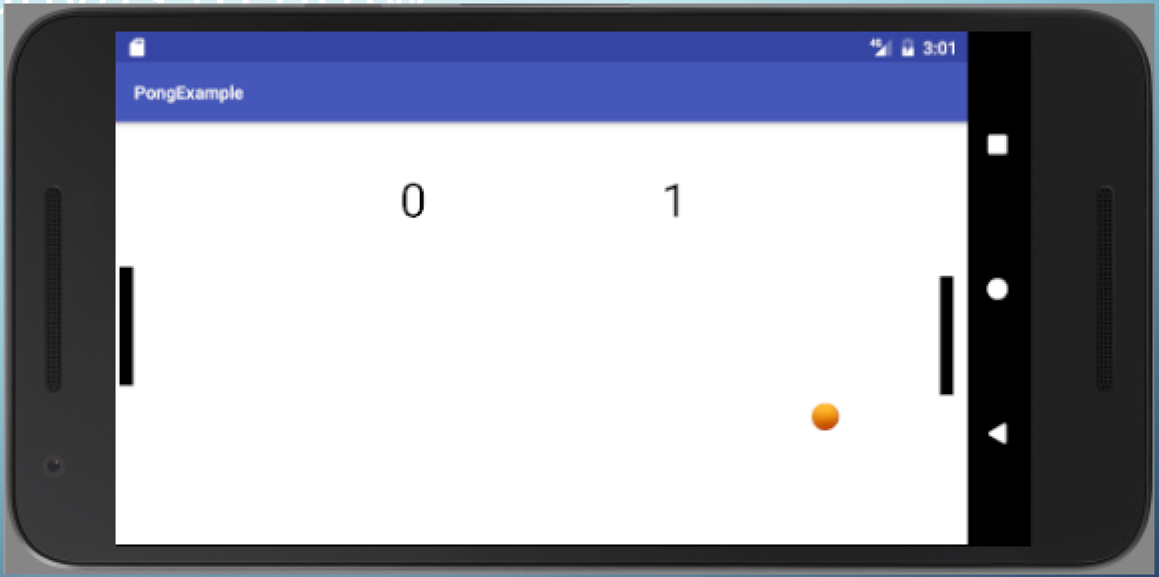
SIMPLE PONG GAME

RICHARD FROLIA & BRIAN KONDOR

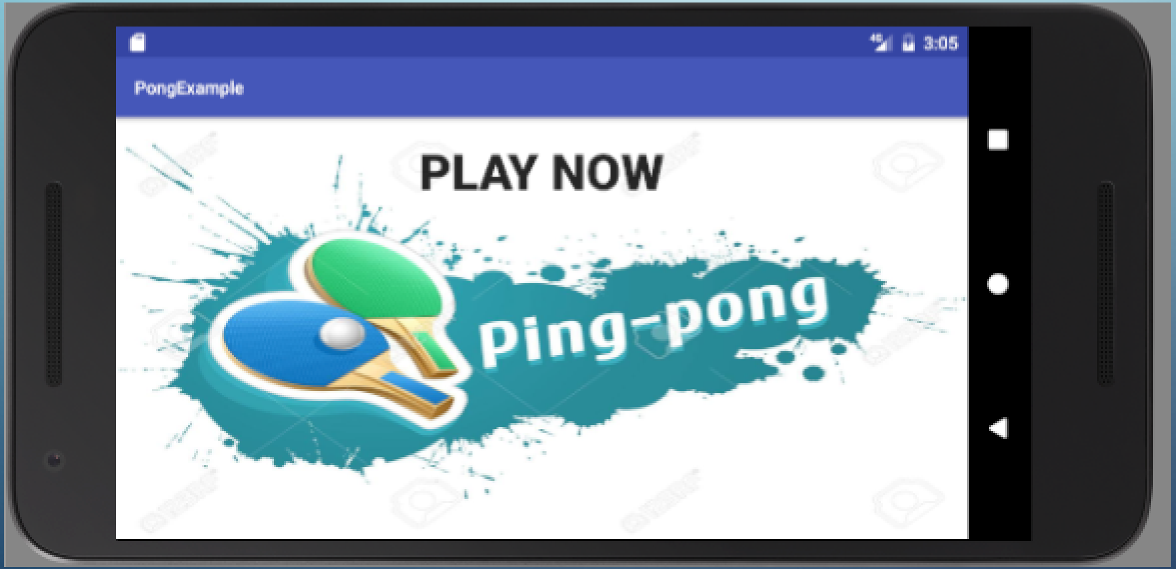
WHAT WE'RE MAKING TODAY

- We're going to demo how to make a simple pong game using a game loop and a canvas to draw



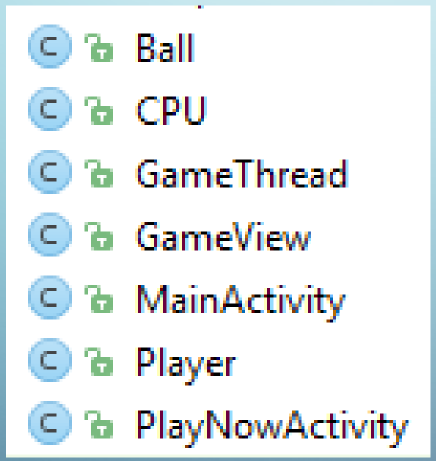
PLAY NOW MENU

- First we'll start off by making a pretty generic play now menu
- Simple Vertical Linear layout with a transparent button
- Set the screen orientation to landscape in the manifest



SETTING UP SOME CLASSES

- Before we go any further lets set up classes for what we need.



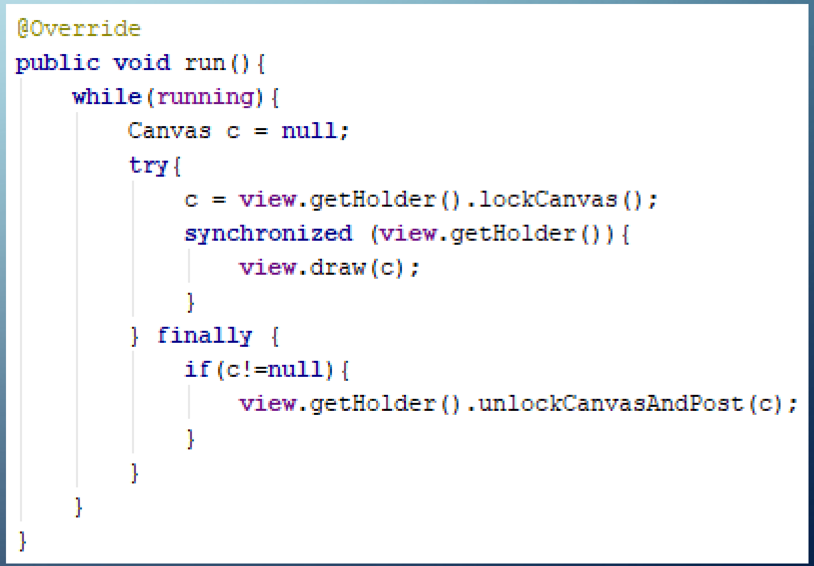
Ball
CPU
GameThread
GameView
MainActivity
Player
PlayNowActivity

GAMETHREAD AND GAMEVIEW SETUP

- The GameThread class extends the Thread class
  - Going to be responsible for continuously drawing the game view while the running Boolean is true
- The GameView class extends surface view
  - Draws everything on the canvas and sets the running Boolean of the game thread

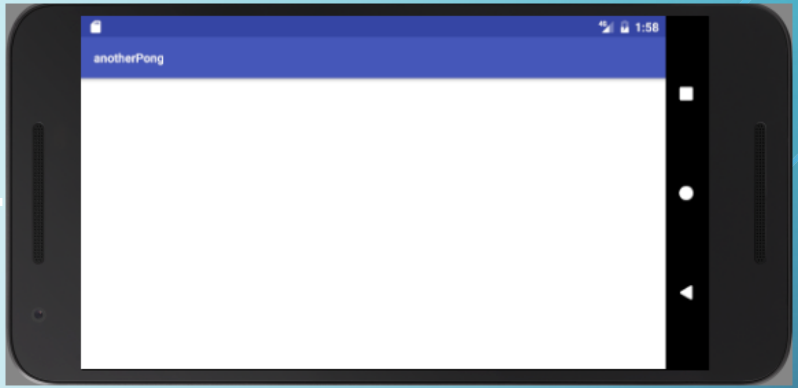
GAMEVIEW DRAW AND GAMETHREAD RUN

- The draw method predictably draws stuff onto the canvas
- For now we will just draw a blank white canvas
- The run method continuously draws the canvas while runnable is set to true



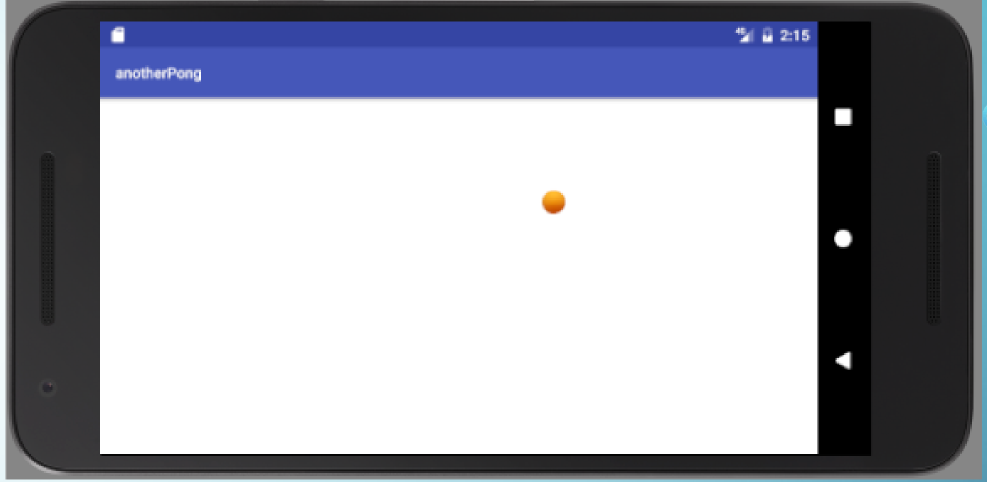
```
@Override
public void run(){
    while(running){
        Canvas c = null;
        try{
            c = view.getHolder().lockCanvas();
            synchronized (view.getHolder()){
                view.draw(c);
            }
        } finally {
            if(c!=null){
                view.getHolder().unlockCanvasAndPost(c);
            }
        }
    }
}
```

BACK TO THE PLAY NOW BUTT



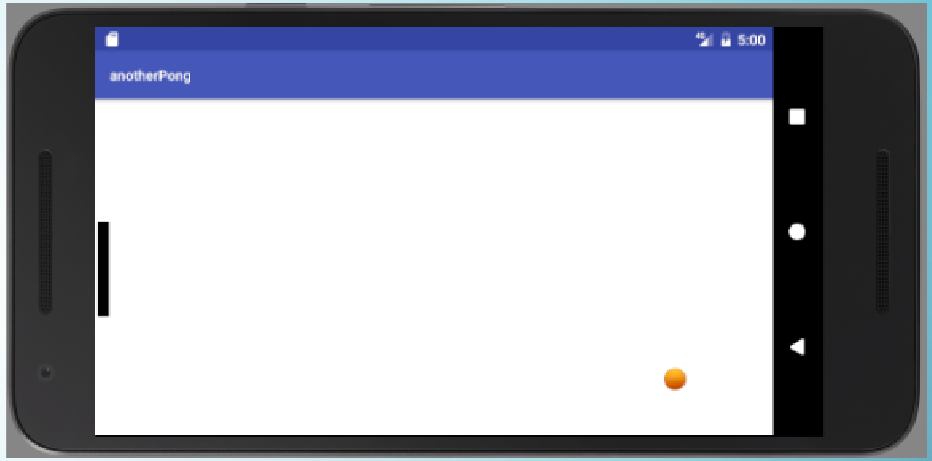
- Launch an intent in the onClick for the play now button that calls the PlayNowActivity class
- PlayNowActivity extends AppCompatActivity. OnCreate sets the content view to a new GameView
- Make sure to declare the PlayNowActivity in the manifest!
- Now when we click play we have a blank white canvas showing

BALL



- Field variables for location, vertical/horizontal direction, score for/against, and a bitmap
- Main part of the class is the move method. It determines where the ball will be located next time the game view is drawen.
- Declare a ball in the GameView and draw a bitmap to the canvas with the balls current location
- Now we have a moving ball on the screen

CPU



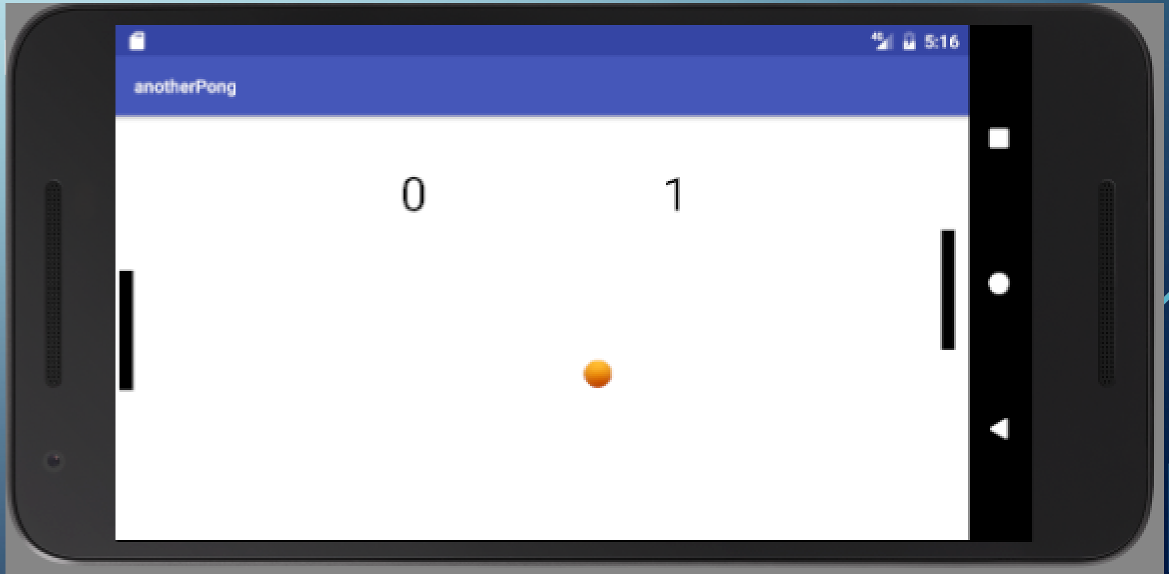
- Field variables for the left, right, top, and bottom locations of the rectangle, the upper and lower bounds of the movement path, and the direction it’s traveling.
- Move method that just goes up and down
- Declare a CPU in Gameview, call the mainmethod, and draw a rectangle on the canvas based on the CPU location.
- Now we have a CPU rectangle that goes up and down on the screen

PLAYER AND ONTOUCHEVENT

- Only field variables for the four sides of the rectangle and setters for the upper and lower portions
- Draw the same way as CPU
- Set the vertical location based on a onTouchEvent in gameview

COLLISIONS AND SCORES

- Check if the balls location is within a certain distance of either rectangle. If yes call the switchDirections method we made in the ball class.
- We're already keeping track of the score in the ball class, so now we just have to draw it o



QUESTIONS?