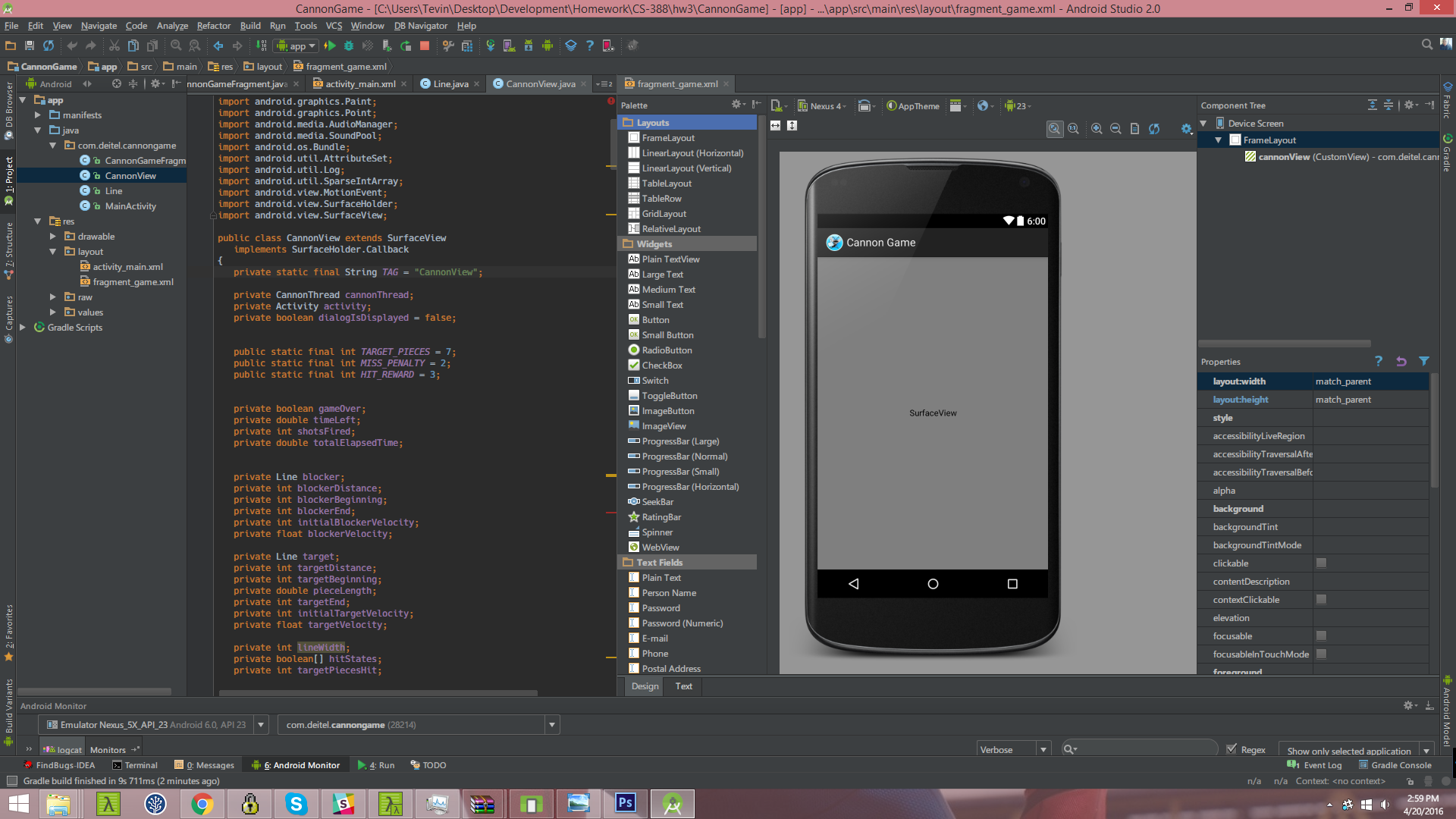
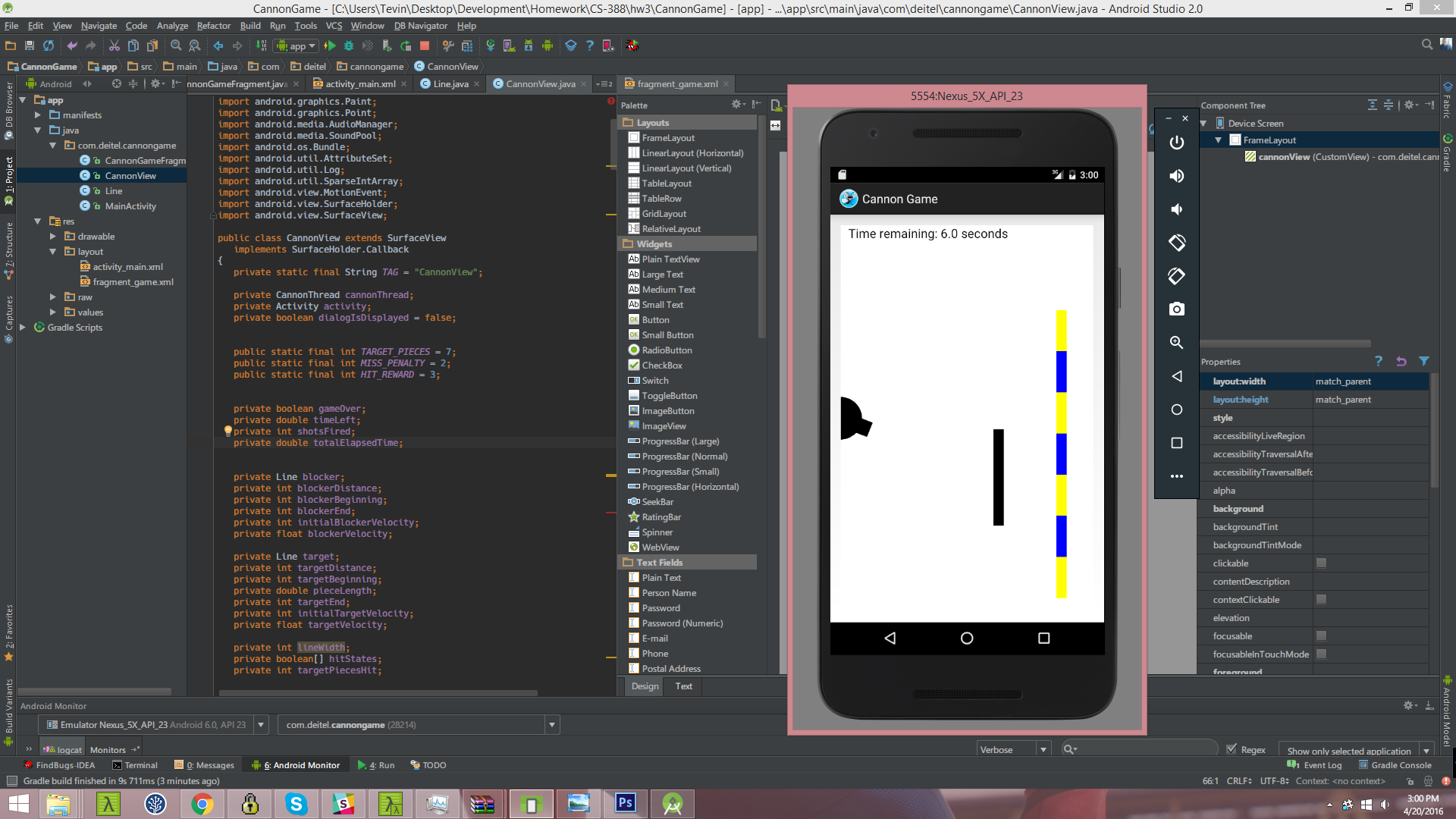
Tevin Jeffrey

1. How Chapter 6 Cannon Game works, pps 183-210





**MainActivity** is called when the app first launches with a layout that is tied to the **CannonGameFragment**. This fragment obtains a reference to a CanonView in onCreate View. In onActivityCreated we set up volume control once Activity is created. When MainActivity is paused, CannonGameFragment terminates the game. When MainActivity is destroyed, CannonGameFragment releases resources.

**CannonView** which extends Surface view, contains many variables for the game loop and tracking statistics, variables for the blocker and target, variables for the cannon and cannonball, constants and variables for managing sounds, Paint variables used when drawing each item on the screen.

When the view is initialized, we register *SurfaceHolder.Callback* listeners, initialize Lines and Point representing game items, initialize hitStates as a boolean array, initialize SoundPool to play the app's three sound effects, create Map of sounds and pre-load sounds, construct Paints for drawing text, cannonball, cannon, blocker and target; these are configured in method *onSizeChanged*. *onSizeChanged* is called by surfaceChanged when the size of the SurfaceView changes, such as when it's first added to the View hierarchy. *updatePositions* is called repeatedly by the CannonThread to update game elements. It does function like updating cannonball position, check for collision with blocker and play blocker sounds. *fireCannonball* fires a cannonball when the user touches the screen. *showGameOverDialog* display an AlertDialog when the game ends. It must obtain the UI thread to draw as you cannot up date the UI thread from the CannonThread. drawGameElements draws the game to the given Canvas.