

Advanced Graphics

Graphics2D Quick Review

```
// Entire Screen Display
```

```
@Override
```

```
public void onCreate (Bundle savedInstanceState)
{
    super.onCreate (savedInstanceState);
    requestWindowFeature (Window.FEATURE_NO_TITLE);
    this.getWindow ().setFlags
        (WindowManager.LayoutParams.FLAG_FULLSCREEN,
         WindowManager.LayoutParams.FLAG_FULLSCREEN);

    WindowManager window = getWindowManager ();
    mDisplay = window.getDefaultDisplay ();

    setContentView (new MySurfaceView (this));
}
```

Graphics with View

```
public class MySurfaceView extends View
{
    private Bitmap mBitmap;
    public Panel(Context context)
    {
        super(context);
        mBitmap = BitmapFactory.decodeResource(getResources(),
            R.drawable.ball_blue);
    }
    @Override
    public void onDraw(Canvas canvas)
    {
        Log.d("Panel", "width:" + getWidth () + "height:" + getHeight ());
        canvas.drawColor(Color.BLACK);
        canvas.drawBitmap(mBitmap, getWidth() / 2, getHeight () / 2, null);
    }
}
```

Graphics with a SurfaceView

- SurfaceView
 - Provides a dedicated drawing surface embedded inside a view hierarchy
 - Allows surface formatting such as size change to occur
 - Allows surface access via the SurfaceHolder interface retrieved using `getHolder ()`
 - Best way to do custom animation and games

Why SurfaceView?

- Provides a surface for a secondary thread to render to the screen
- IMPORTANT Thread issues
 - All SurfaceView and callback methods are called from the thread running the SurfaceView's window
 - Thread synchronization is necessary
 - Drawing thread can only touch Surface while the Surface exists

View versus SurfaceView?

- If we change the previous example to extend SurfaceView, what happens?

Answer: Nothing as `onDraw ()` is never called

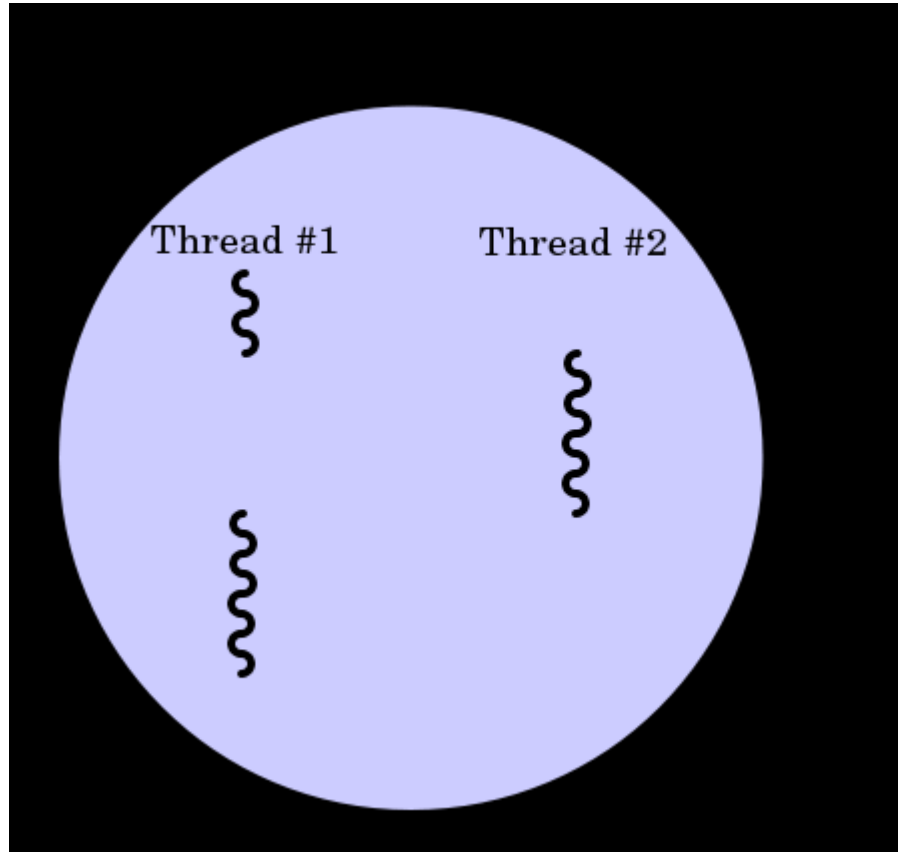
- What to do?

Answer: Implement a view thread that will handle the drawing. Why? A thread gives more control over when and where you draw

Threads

- A thread
 - is the smallest unit of processing that an OS can schedule
 - is a concurrent unit of execution
 - has its own call stack for invoked methods
- Each virtual machine has at least one main thread of execution
- Each virtual machine typically has other threads for housekeeping
- Applications can start their own threads for specific purposes

Threads



Back to our SurfaceView story

SurfaceHolder.Callback Interface

- `surfaceCreated` (`SurfaceHolder holder`) - called when surface is first created and used to start up the rendering code
- `surfaceDestroyed` (`SurfaceHolder holder`) - called right before the surface is destroyed
- `surfaceChanged` (`SurfaceHolder`, `int format`, `int width`, `int height`) - called immediately after any structural changes are made

Using a Drawing Thread

- Grab BallAnimation in CS260-01 Public

Using a Drawing Thread

- Grab BallAnimation in CS260-01 Public

Problem

- Figure out a way to determine the Frame Rate in the frame rate class.
- Check the frame rate for GraphicsView
- Check the frame rate for GraphicsSurfaceView
- Modify the GraphicsSurfaceView to add multiple animating balls as before
- Check the frame rate as more balls are added