Aula 03 – Model-View-Controller

Programação em Java para a Plataforma Android

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Agenda

- O que é modelo, visão e controle
- Como implementar a camada modelo
- O princípio da responsabilidade única
- Objetos imutáveis
- Quais eventos pode ser tratados?
- O que são cadeias de responsabilidades?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Um padrão bem conhecido

- O que é a visão?
- O que é o modelo?
- O que é o controlador?
- Por que esse é um bom padrão de projeto?

Visão
(View)

Controlador
(Controller)

Modelo
(Model)

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Exemplo: fazedor de pontos

Especificação do programa:

- Crie uma aplicação que desenhe pontos aleatórios na tela.
  - Esta aplicação deverá ter dois botões, **RED** e **GREEN**.
  - Ela terá também duas caixas de texto: x e y.
  - E uma área de tela onde pontos devem ser desenhados.
  - Ao clicar em RED, um ponto vermelho deverá ser desenhado em uma posição aleatória dentro da área de pontos.
  - Ao clicar em GREEN um ponto verde deverá ser desenhado.
  - As caixas de texto marcarão a coordenada do último ponto desenhado.

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Qual a visão do programa?

Especificação do programa:

- Crie uma aplicação que desenhe pontos aleatórios na tela.
  - Esta aplicação deverá ter dois botões, **RED** e **GREEN**.
  - Ela terá também duas caixas de texto: x e y.
  - E uma área de tela onde pontos devem ser desenhados.
  - Ao clicar em RED, um ponto vermelho deverá ser desenhado em uma posição aleatória dentro da área de pontos.
  - Ao clicar em GREEN um ponto verde deverá ser desenhado.
  - As caixas de texto marcarão a coordenada do último ponto desenhado.

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Qual a visão do programa?

Protótipo:
Esquematize um protótipo dessa visão.

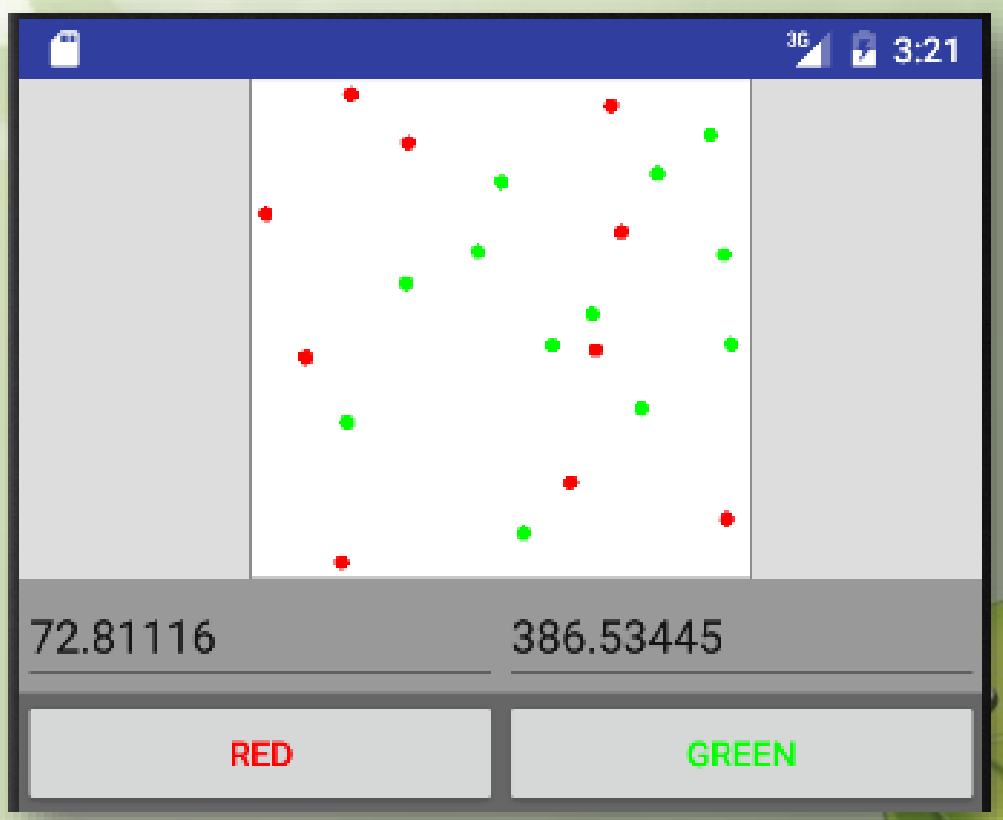
Especificação do programa:

- Crie uma aplicação que desenhe pontos aleatórios na tela.
  - Esta aplicação deverá ter dois botões, **RED** e **GREEN**.
  - Ela terá também duas caixas de texto: x e y.
  - E uma área de tela onde pontos devem ser desenhados.
  - Ao clicar em RED, um ponto vermelho deverá ser desenhado em uma posição aleatória dentro da área de pontos.
  - Ao clicar em GREEN um ponto verde deverá ser desenhado.
  - As caixas de texto marcarão a coordenada do último ponto desenhado.

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Protótipo

- Quantos componentes esse protótipo possui?
- O que é a grande área de desenho?
- Como criar esse protótipo em XML?



Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

O que já aprendemos...

- Como criar a área branca de pontos?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

A área de pontos

```
public class AulaActivity3 extends Activity {
    final Dots dotModel = new Dots();
    private DotView dotView;
    ...
    public void onCreate(Bundle state) {
        super.onCreate(state);
        dotView = new DotView(this, dotModel);
        setContentView(R.layout.main);
        ((LinearLayout)findViewById(R.id.root)).
                addView(dotView, 0);
        ...
```

Sob demanda:
Por hora, vamos apenas supor que DotView e Dots existem, e que podemos usá-los.

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

O Modelo

- O **Modelo** são os **dados** sobre os quais a aplicação atua
  - E as operações que esses dados "*sabem*" realizar.

- Mas, qual é o modelo dessa nossa aplicação em questão?

Programação orientada a Objetos:
E desde quando dados sabem fazer qualquer coisa?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Um Ponto

- O que um ponto sabe fazer?
  - Em outras palavras, qual é a interface de um ponto?

```
public float getX();

public float getY();

public int getColor();

public int getDiameter();
```

Implementação: Como podemos implementar essa interface?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

O ponto por dentro

```
public final class Dot {
    private final float x, y;
    private final int color;
    private final int diameter;
    public Dot(float x, float y, int color, int diameter) {
        this.x = x;
        this.y = y;
        this.color = color;
        this.diameter = diameter;
    }
    public float getX() { return x; }
    public float getY() { return y; }
    public int getColor() { return color; }
    public int getDiameter() { return diameter; }
}
```

Encapsulamento:
Por que alguns campos são públicos e outros privados?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Muitos Pontos

- Precisamos de uma estrutura separada para representar um conjunto de pontos?

- Qual a vantagem de reusar uma estrutura que já existe?
  - E qual poderia ser essa estrutura?

- E qual a vantagem de usarmos uma estrutura nova?
  - O que uma lista de pontos sabe fazer?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Uma lista de Pontos

```
void setDotsChangeListener(DotsChangeListener l);

Dot getLastDot();

List<Dot> getDots();

void addDot(float x, float y,

int color, int diameter);

void clearDots();
```

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Eventos de Pontos

- A lista de pontos escuta eventos.
  - Mas que eventos são esses?

```
public class Dots {
    public interface DotsChangeListener {
        void onDotsChange(Dots dots);
    }
    private final LinkedList<Dot> dots = new LinkedList<Dot>();
    private DotsChangeListener dotsChangeListener;
    public List<Dot> getDots() {
        return dots;
    }
    public void setDotsChangeListener(DotsChangeListener l) {
        dotsChangeListener = l;
    }
    public void addDot(float x, float y, int color, int dim) {
        dots.add(new Dot(x, y, color, dim));
        notifyListener();
    }
    public void clearDots() {
        dots.clear();
        notifyListener();
    }
    private void notifyListener() {
        if (null != dotsChangeListener) {
            dotsChangeListener.onDotsChange(this);
        }
    }
}
```

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Responsabilidade(s)

- A lista de pontos sabe desenhar os próprios pontos na tela do dispositivo?
- Quais os prós de dar esse conhecimento à lista de pontos?
- E quais os contras?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Invariantes

- De acordo com o princípio da Responsabilidade Única, a lista de pontos não deveria saber desenhar os pontos em uma tela de dispositivo

- Mas temos um problema:
  - Alguém tem de desenhar os pontos
  - Para desenhar os pontos, é preciso ver a lista de pontos
  - Mas não queremos que essa lista possa ser modificada fora de `Dots`. **Por que?**

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Objetos Imutáveis

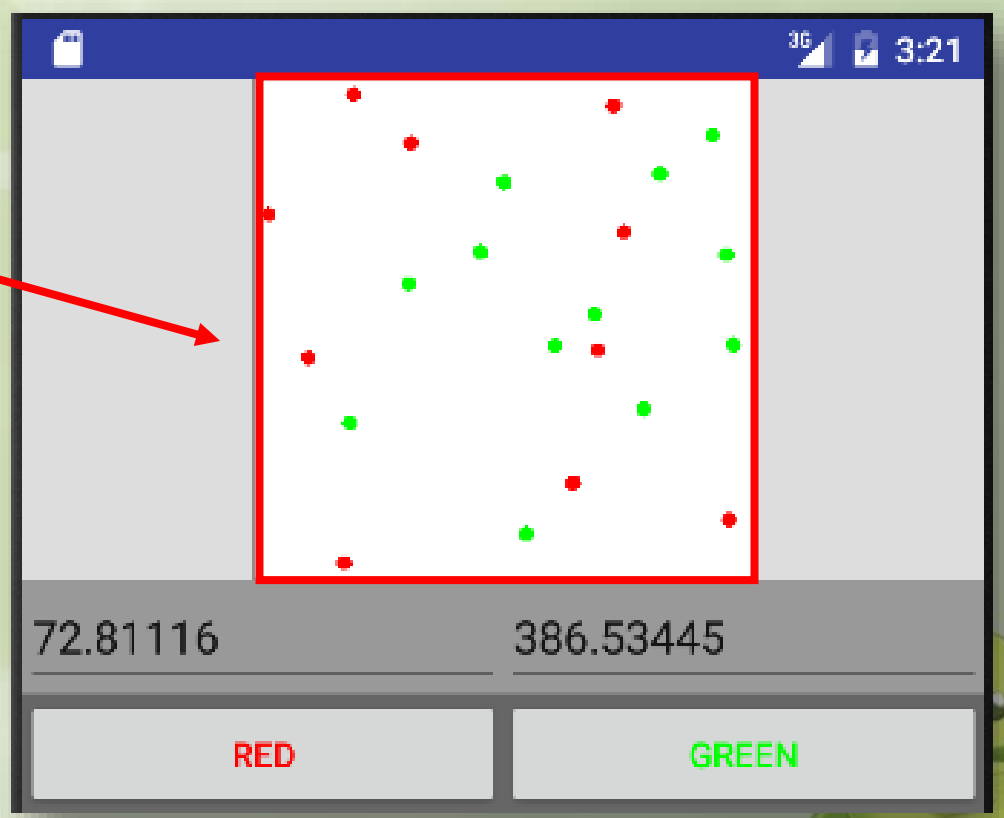
```
public class Dots {
    public interface DotsChangeListener {
        void onDotsChange(Dots dots);
    }
    private final LinkedList<Dot> dots = new LinkedList<Dot>();
    private DotsChangeListener dotsChangeListener;
    private final List<Dot> safeDots = Collections.unmodifiableList(dots);
    public List<Dot> getDots() {
        return safeDots;
    }
```

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

A Janela de Pontos

- Precisamos de uma **visão** para desenhar os pontos
  - Não existe uma visão desse tipo na biblioteca de Android
  - Precisamos criá-la a partir de primitivas gráficas

- Mas em geral é melhor reusar código que já existe. **Por que?**



Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

DotView

```
public class DotView extends View {
    private final Dots dots;
    public DotView(Context context, Dots dots) {…}
    @Override
    protected void onMeasure(int widthMeasureSpec,
                             int heightMeasureSpec) {
        setMeasuredDimension(getSuggestedMinimumWidth(),
                getSuggestedMinimumHeight());
    }
    @Override
    protected void onDraw(Canvas canvas) {…}
}
```

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

DotView

```
public class DotView extends View {
    private final Dots dots;
    public DotView(Context context, Dots dots) {…}
    @Override
    protected void onMeasure(int widthMeasureSpec,
                             int heightMeasureSpec) {
        setMeasuredDimension(getSuggestedMinimumWidth(),
                getSuggestedMinimumHeight());
    }
    @Override
    protected void onDraw(Canvas canvas) {…}
}
```

O que deveríamos fazer no construtor?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

```
public class DotView extends View {
    private final Dots dots;
    public DotView(Context context, Dots dots) {
        super(context);
        this.dots = dots;
        setMinimumWidth(400);
        setMinimumHeight(400);
        setFocusable(true);
    }
    @Override
    protected void onMeasure(int widthMeasureSpec,
                             int heightMeasureSpec) {
        setMeasuredDimension(getSuggestedMinimumWidth(),
                getSuggestedMinimumHeight());
    }
    @Override
    protected void onDraw(Canvas canvas) {…}
}
```

E como deveria ser o onDraw?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

DotView

```
@Override
protected void onDraw(Canvas canvas) {
    canvas.drawColor(Color.WHITE);

    Paint paint = new Paint();
    paint.setStyle(Paint.Style.STROKE);
    paint.setColor(hasFocus() ? Color.BLUE : Color.GRAY);
    canvas.drawRect(0, 0, getWidth() - 1, getHeight() - 1, paint);

    paint.setStyle(Paint.Style.FILL);
    for (Dot dot : dots.getDots()) {
        paint.setColor(dot.getColor());
        canvas.drawCircle(dot.getX(), dot.getY(), dot.getDiameter(), paint);
    }
}
```

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

DotView

```
@Override
protected void onDraw(Canvas canvas) {
    canvas.drawColor(Color.WHITE);

    Paint paint = new Paint();
    paint.setStyle(Paint.Style.STROKE);
    paint.setColor(hasFocus() ? Color.BLUE : Color.GRAY);
    canvas.drawRect(0, 0, getWidth() - 1, getHeight() - 1, paint);

    paint.setStyle(Paint.Style.FILL);
    for (Dot dot : dots.getDots()) {
        paint.setColor(dot.getColor());
        canvas.drawCircle(dot.getX(), dot.getY(), dot.getDiameter(), paint);
    }
}
```

O que esse código está fazendo?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

DotView

```
@Override
protected void onDraw(Canvas canvas) {
    canvas.drawColor(Color.WHITE);

    Paint paint = new Paint();
    paint.setStyle(Paint.Style.STROKE);
    paint.setColor(hasFocus() ? Color.BLUE : Color.GRAY);
    canvas.drawRect(0, 0, getWidth() - 1, getHeight() - 1, paint);

    paint.setStyle(Paint.Style.FILL);
    for (Dot dot : dots.getDots()) {
        paint.setColor(dot.getColor());
        canvas.drawCircle(dot.getX(), dot.getY(), dot.getDiameter(), paint);
    }
}
```

É possível inserir novos pontos aqui?

O controlador

- Quais eventos precisam ser controlados?
- O botão GREEN recebe um clique?
  - Um ponto verde é criado na lista de pontos
- O botão RED recebe um clique?
  - Um ponto vermelho é criado na lista de pontos
- E quando os pontos são desenhados?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Eventos de Botão

E como é a implementação de makeDot?

```
public void onCreate(Bundle state) {
    super.onCreate(state);
    ...
    ((Button) findViewById(R.id.button1))
            .setOnClickListener(new Button.OnClickListener() {
                public void onClick(View v) {
                    makeDot(dotModel, dotView, Color.RED);
                }
            });
    ((Button) findViewById(R.id.button2))
            .setOnClickListener(new Button.OnClickListener() {
                public void onClick(View v) {
                    makeDot(dotModel, dotView, Color.GREEN);
                }
            });
    ...
```

Desenhando os Pontos

```
private final Random rand = new Random();
public static final int DOT_DIAMETER = 6;
void makeDot(Dots dots, DotView view, int color) {
    int pad = (DOT_DIAMETER + 2) * 2;
    dots.addDot(
            DOT_DIAMETER + (rand.nextFloat()
               * (view.getWidth() - pad)),
            DOT_DIAMETER + (rand.nextFloat()
               * (view.getHeight() - pad)),
            color, DOT_DIAMETER);
}
```

E o que acontece quando a lista de pontos muda?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Eventos de Pontos

```
public void onCreate(Bundle state) {
    super.onCreate(state);
    ...
    final EditText tb1 = (EditText) findViewById(R.id.text1);
    final EditText tb2 = (EditText) findViewById(R.id.text2);
    dotModel.setDotsChangeListener(
            new Dots.DotsChangeListener() {
                public void onDotsChange(Dots dots) {
                    Dot d = dots.getLastDot();
                    tb1.setText((null == d)
                            ? "" : String.valueOf(d.getX()));
                    tb2.setText((null == d)
                            ? "" : String.valueOf(d.getY()));
                    dotView.invalidate();
                }
            });
    ...
```

Ops, ainda não implementamos o getLastDot()

Eventos de Pontos

```
public Dot getLastDot() {
    return (dots.size()<=0)?null:dots.getLast();
}
```

Qual é a ordem em que as coisas acontecem?

A ordem das coisas

1. Quando o botão recebe um clique, o `ClickHandler` dele é chamado.
2. A classe anônima no `ClickHandler` chama o método `makeDot`.
3. Um ponto é adicionado à `Dots`.
4. Um evento de ponto é detectado por `DotsChangeListener`.
5. O observador pede à `DotView` que redesenhe os pontos.

Ps.: Já podemos executar o app!

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Múltiplos eventos

- Como pontos estão sendo criados atualmente?
- Que outros eventos podemos usar para criar pontos?
- E que outros eventos podemos usar para enriquecer a nossa aplicação?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Eventos de toque

```
...
dotView.setOnTouchListener(new View.OnTouchListener() {
    public boolean onTouch(View v, MotionEvent event) {
        if (MotionEvent.ACTION_DOWN != event.getAction()) {
            return false;
        }
        dotModel.addDot(event.getX(),
                event.getY(), Color.CYAN, DOT_DIAMETER);
        return true;
    }
});
...
```

Qual evento está sendo detectado?

Que outros eventos de toque devem existir?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Apertar, Arrastar e Soltar

- Eventos de movimento podem sobrecarregar um dispositivo mais lento.
  - Coordenadas precisam ser atualizadas o tempo todo.

- Android permite que esses eventos sejam armazenados em um buffer.
  - E posteriormente consultados.

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Rastreando o Touch Pad

```
public class TrackingTouchListener
        implements View.OnTouchListener {
    private final Dots mDots;
    TrackingTouchListener(Dots dots) {
        mDots = dots;
    }
    private void addDot
            (Dots dots, float x, float y, float p, float s) {
        dots.addDot(x, y, Color.CYAN,
                (int) ((p * s * AulaActivity3.DOT_DIAMETER) + 1));
    }
    public boolean onTouch(View v, MotionEvent evt) {
        ...
    }
}
```

O que seria esse `p` e esse `s`?

Como deve ser a implementação de `onTouch`?

Rastreando o Touch Pad

```
public boolean onTouch(View v, MotionEvent evt) {
    switch (evt.getAction()) {
        case MotionEvent.ACTION_DOWN:
            break;
        case MotionEvent.ACTION_MOVE:
            for (int i = 0, n = evt.getHistorySize(); i < n; i++) {
                addDot(mDots, evt.getHistoricalX(i),
                        evt.getHistoricalY(i), evt.getHistoricalPressure(i),
                        evt.getHistoricalSize(i));
            }
            break;
        default:
            return false;
    }
    addDot(mDots, evt.getX(), evt.getY(), evt.getPressure(),
            evt.getSize());
    return true;
}
```

Mas como podemos registrar e esse escutador de eventos?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Rastreando o Touch Pad

```
public void onCreate(Bundle state) {
    ...
    dotView.setOnTouchListener(new TrackingTouchListener(dotModel));
    ...
}
```

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Eventos de teclas

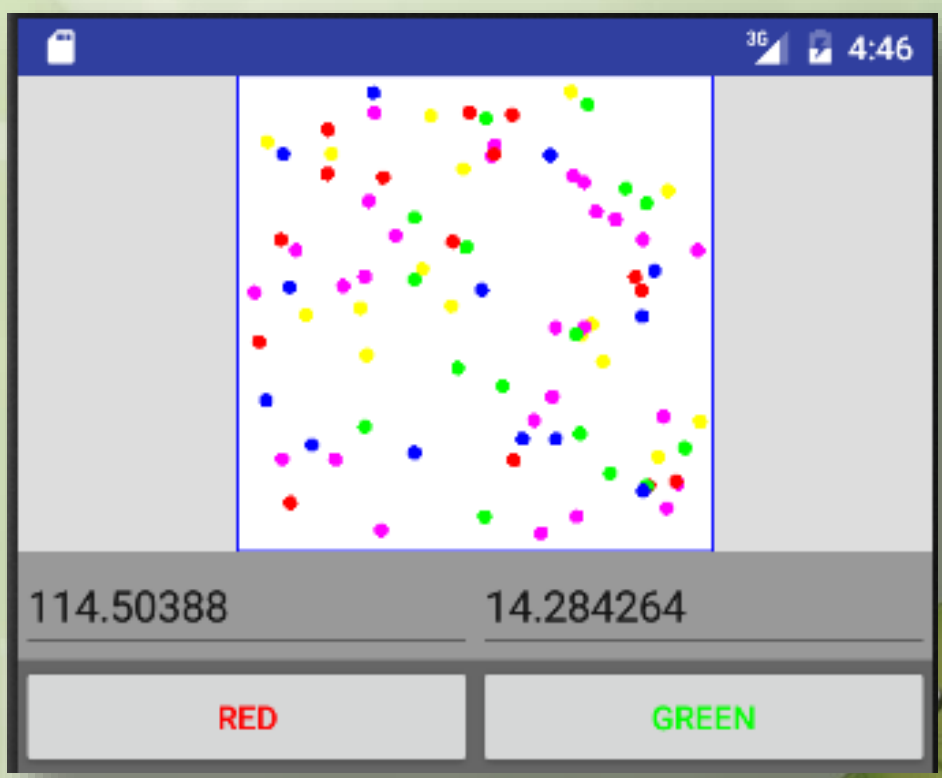
```
dotView.setFocusable(true);
dotView.setOnKeyListener(new View.OnKeyListener() {
    public boolean onKey(View v, int keyCode, KeyEvent event) {
        if (KeyEvent.ACTION_UP != event.getAction()) {
            int color = Color.BLUE;
            switch (keyCode) {
                case KeyEvent.KEYCODE_SPACE:
                    color = Color.MAGENTA;
                    break;
                case KeyEvent.KEYCODE_ENTER:
                    color = Color.YELLOW;
                    break;
                default:
                    ;
            }
            makeDot(dotModel, dotView, color);
        }
        return true;
    }
});
```

O que esse código faz?

Eventos de tecla

- Cada tecla possui um número identificador

- Teclas podem ser tratadas de forma diferenciada em um `switch`

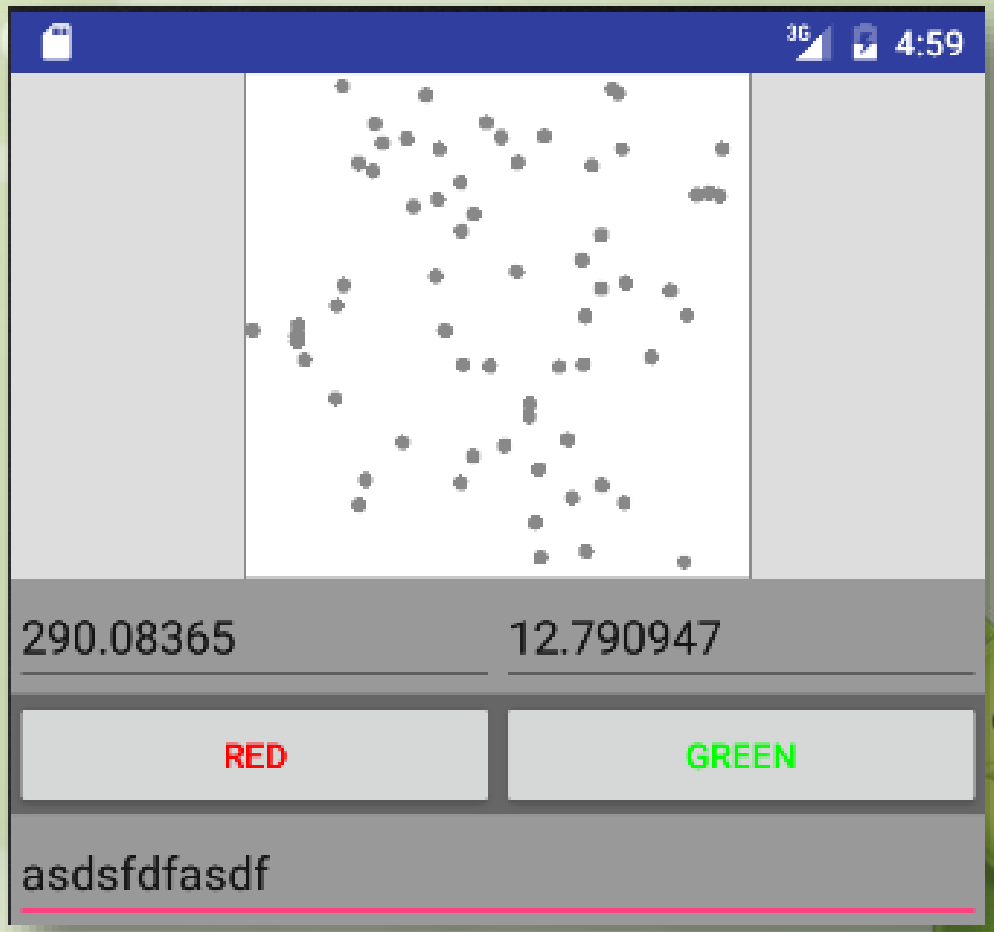


Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Caixa de texto editável

Eventos de Teclado:

- Adicione uma caixa de texto à sua aplicação.
- Esta caixa precisa escutar eventos de teclado.
- Teclas numéricas causam o aparecimento de pontos cinza na área de desenho.
- As outras teclas escrevem na caixa de texto.



Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Caixa de texto editável

```
...
<LinearLayout
    android:orientation="horizontal"
    android:background="@color/grey"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <EditText
        android:id="@+id/text3"
        android:text="@string/defaultText"
        android:focusable="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"/>
</LinearLayout>
</LinearLayout>
```

O que esse atributo faz?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Cadeia de responsabilidades

- Eventos são implementados segundo uma cadeia de responsabilidades de dois níveis.

- Elementos gráficos sabem tratar alguns eventos.
  - Uma caixa de texto escreveria em sua área as teclas pressionadas.

- O usuário pode sobreescrever este tratamento padrão.
  - Mas pode passar alguns eventos para ele também.

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Passa, não passa

```
final EditText text3 = (EditText) findViewById(R.id.text3);
text3.setOnKeyListener(new View.OnKeyListener() {
    @Override
    public boolean onKey(View v, int keyCode, KeyEvent event) {
        if (KeyEvent.ACTION_UP != event.getAction()) {
            if(Character.isDigit(event.getUnicodeChar())){
                makeDot(dotModel, dotView, Color.GRAY);
                return true;
            }
        }
        return false;
    }
});
```

Esses eventos nós tratamos aqui!

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Eventos de foco

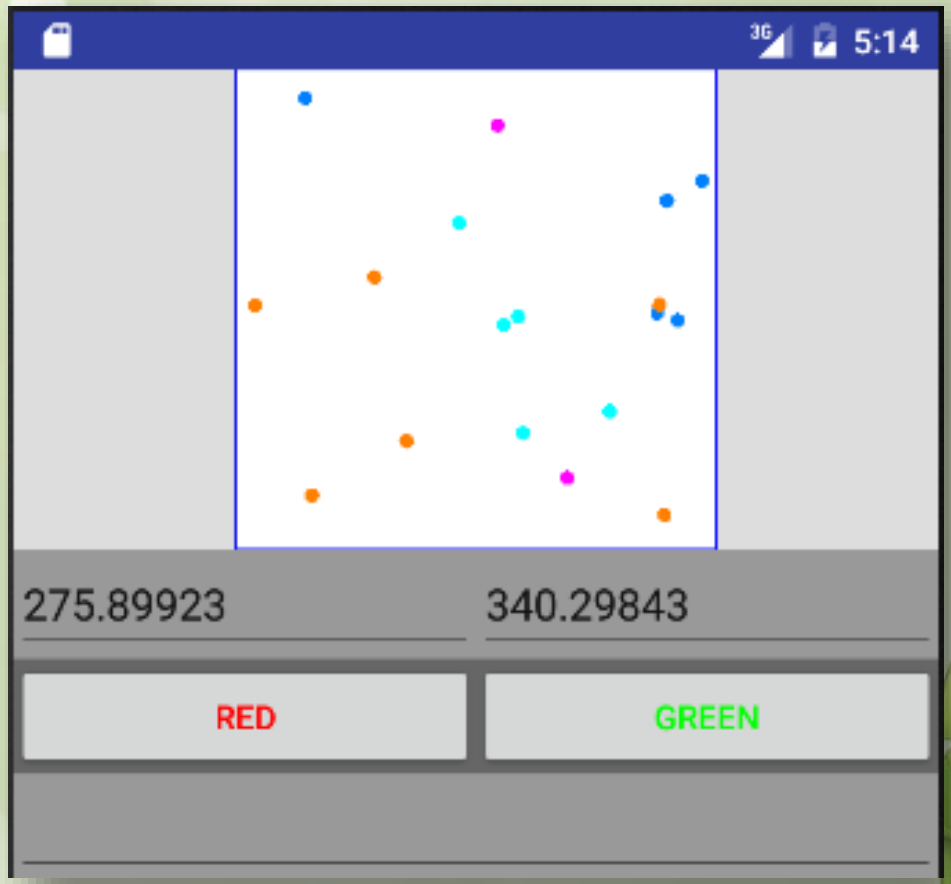
Como detectar a mudança de foco?

Como desenhas pontos dessas cores?

Mudando o foco:

Adicione um tratador de mudança de foco à `dotView`.

- Se `dotView` ganha foco, então você precisa desenhar um ponto alaranjado.
- Se `dotView` perde foco, então você precisa desenhar um ponto azul claro.



Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Eventos de foco

```
dotView.setOnFocusChangeListener(new View.OnFocusChangeListener() {
    public void onFocusChange(View v, boolean hasFocus) {
        if (!hasFocus) {
            makeDot(dotModel, dotView, Color.rgb(0, 128, 255));
        } else {
            makeDot(dotModel, dotView, Color.rgb(255, 128, 0));
        }
    }
});
```

A tela de desenho está ficando uma bagunça... Como limpar tudo?

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Menus

- Podemos adicionar menus a uma atividade sobreescrevendo dois métodos:
  - `onCreateOptionsMenus`
  - `onOptionsItemSelectet`

- O que faz cada um desses métodos faz?

Limpando tudo:
Adicione um menu a sua aplicação, com a opção “clean”, que limpa a tela quando selecionada

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Adicionando um Menu

```
private final int CLEAR_MENU_ID = 1;
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add(Menu.NONE, CLEAR_MENU_ID, Menu.NONE, getString(R.string.limpar));
    return true;
}
@Override
public boolean
onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case CLEAR_MENU_ID:
            dotModel.clearDots();
            return true;
        default:
            ;
    }
    return false;
}
```

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Adicionando um Menu

```
private final int CLEAR_MENU_ID = 1;
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add(Menu.NONE, CLEAR_MENU_ID, Menu.NONE, getString(R.string.limpar));
    return true;
}
@Override
public boolean
onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case CLEAR_MENU_ID:
            dotModel.clearDots();
            return true;
        default:
            ;
    }
    return false;
}
```

O que fazem esses parâmetros?
Como podemos descobrir?

Recapitulando

Visão
(View)

Controlador
(Controller)

Modelo
(Model)

- A visão é a interface gráfica da aplicação, e é definida por um arquivo XML, mais um conjunto de objetos gráficos (`EditText`, `Button`, etc)
- O modelo são os dados que a aplicação manipula. Nesse exemplo: ponto e lista de pontos.
- O controlador são os escutadores e tratadores de eventos. Em geral implementados como *observadores*.

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Exercício: Diâmetro

- Sempre que o foco estiver sobre a área de pontos, e o usuário clicar no d-Pad, desenhe uma linha ligando os dois pontos mais distantes da tela.

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br

Dúvidas

Agripino Gabriel M. Damasceno
agripinogabriel@pucminas.br