GUI アプリ開発

鎌田十三郎

趣旨

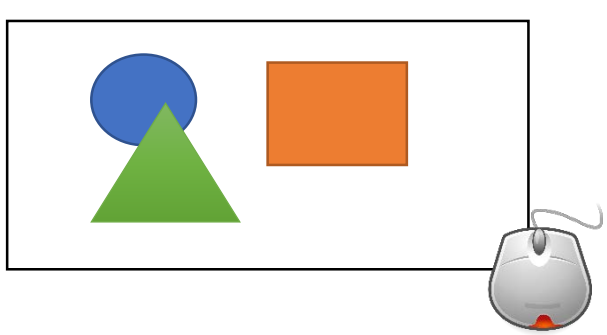
- GUI アプリ開発の基本を学ぶ
  - MVC モデル
  - View の構成
  - イベント処理
  - Draw 系コンポーネントの例

- まずは、slide をつかって簡単な構図の説明
- そのあと、実際のプログラムを見ながら、
  納得してもらう予定。

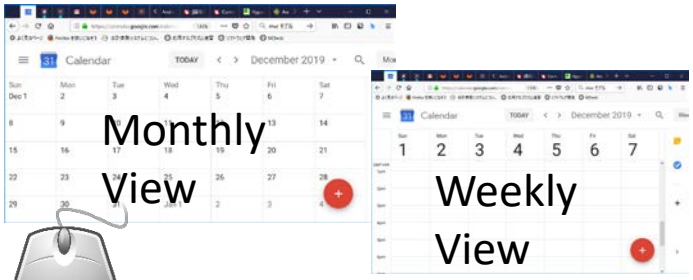
MVC モデル

以下の3つから構成

- Model: データ構造及びロジック
- View: Model を画面表示したもの
- Controller: ユーザ入力に基づいて Model 操作



View



Controller

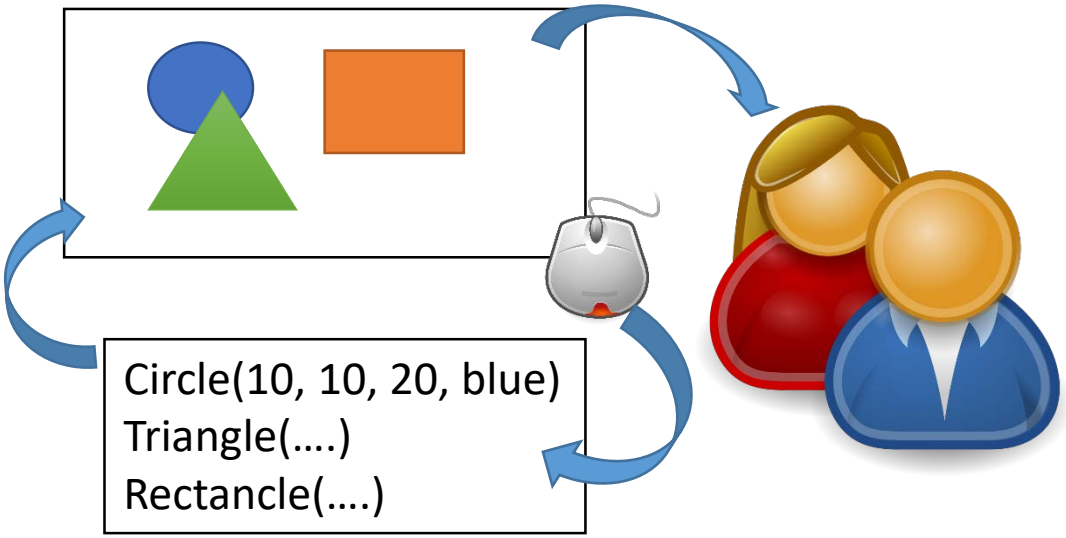
Circle(10, 10, 20, blue)
Triangle(….)
Rectancle(….)

Model

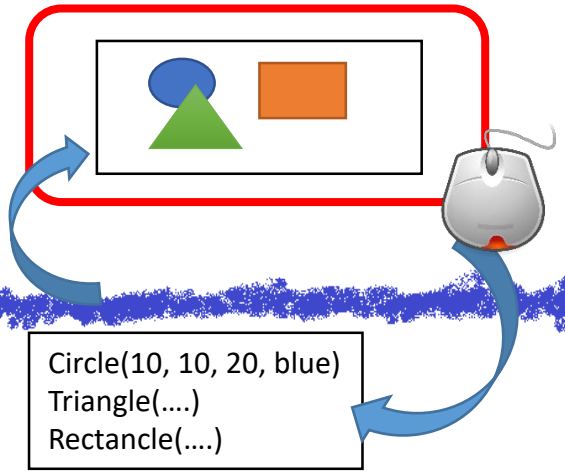
ソフトウェア開発, 10/19, 12:20-13:50
応用アルゴリズム演習, 10/22 10:40-12:10
…..

MVC モデル

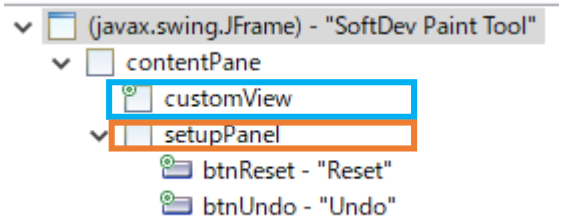
- GUI アプリの場合
  - ユーザが Controller 経由で操作
  - Controller が Model を変更
  - Model の変更をView が反映
  - View をユーザが確認

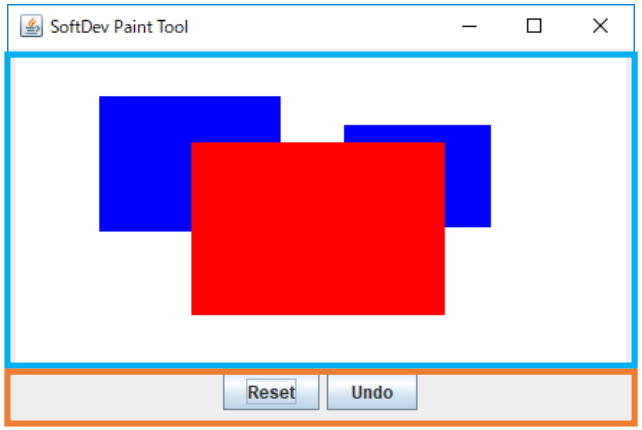


View の構成



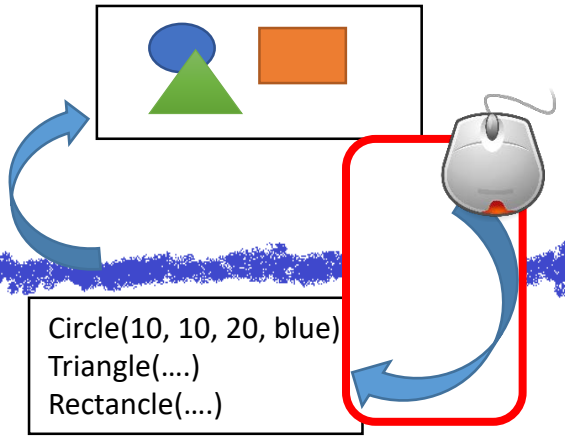
- 基本階層構造



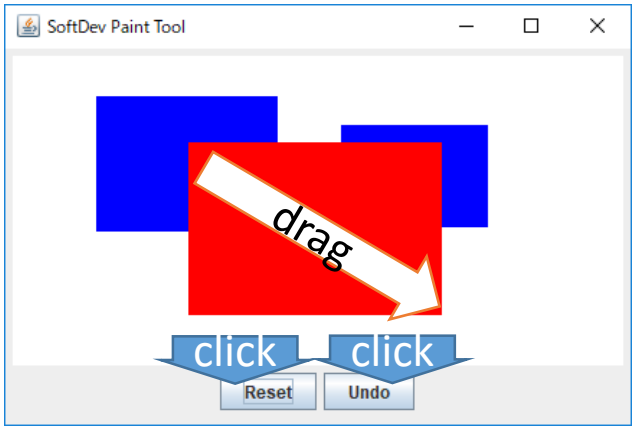


- 開発環境サポート:
  GUI ツールを用いて部品配置
  (Eclipse Window Builder,
  Ansroid Studio)

Event 処理

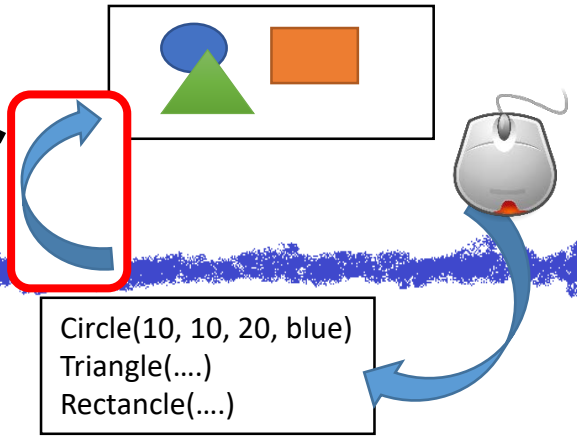


- 各部品で起きたイベントを
  システムが把握
- 開発者が定義・登録した
  ハンドラをシステムが起動
- 当然、インタフェイスが
  決まってないと



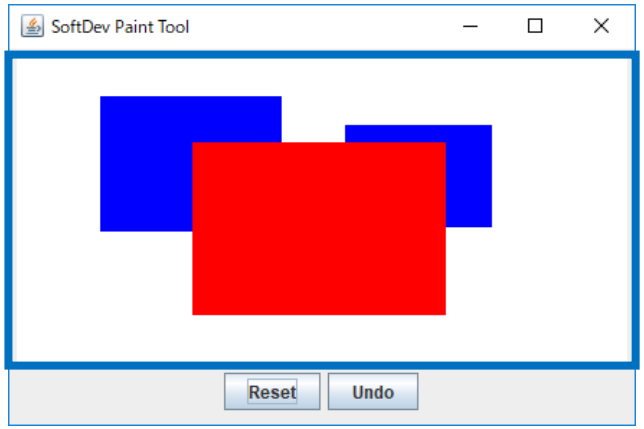
- Event 処理で Model を更新、
  必要に応じてView の再描画

Draw 系コンポーネント



- 通常のボタンなど:
  データの表示法は固定
  - Model に変化
    システムが View 更新

- Model に応じた描画をしたい場合
  - 自分で描画ルーチンを定義
    - Swing: paintComponent(Graphics)
    - Android: View#onDraw(Canvas)
    - システムが必要に応じて実行



  - Model が View 再描画を促す場合は下記メソッドを呼ぶ
    - Swing: repaint(), Android: invalidate()
    - システムはそのうち再描画実行 （1秒に60回もやれば十分)

コード解説は Web テキストで

- 利用事例は二つ