

# Computação II – Orientação a Objetos

---

Fabio Mascarenhas - 2014.1

<http://www.dcc.ufrj.br/~fabiom/java>

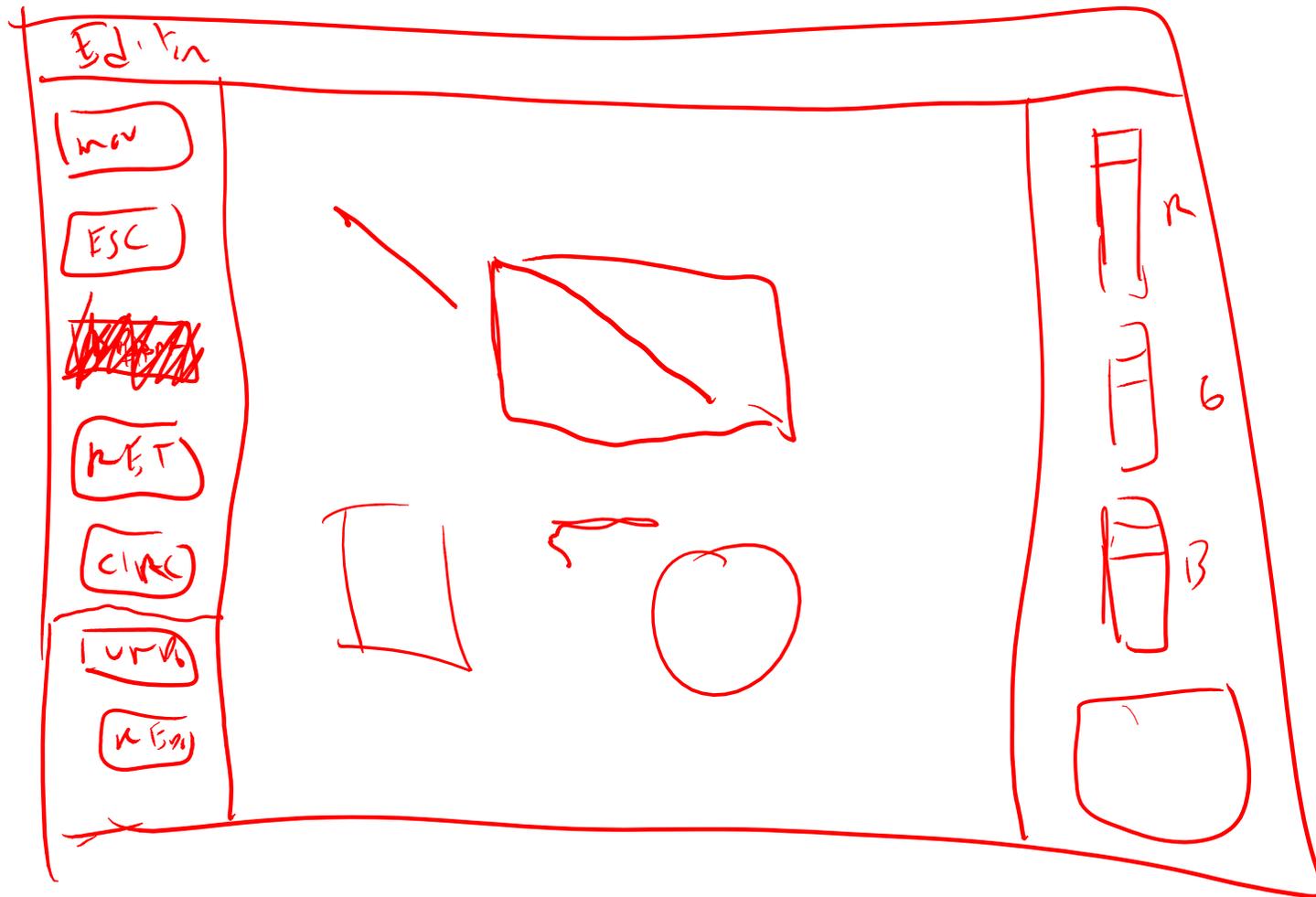
# Editor Gráfico

---

- Vamos usar nosso framework do Motor, com pequenas mudanças (para permitir interação com o mouse) para implementar não um jogo, mas uma aplicação com uma pequena interface gráfica
- Vamos fazer um programa simples para desenho e manipulação de figuras geométricas: um “nano-sketchpad” (lembrem da primeira aula!)
- Nosso editor vai ter botões de comando, as figuras vão poder ser desenhadas e manipuladas usando o mouse, e vamos ter undo e redo (desfazer e refazer) de vários níveis!

# Editor gráfico – esboço da interface

---



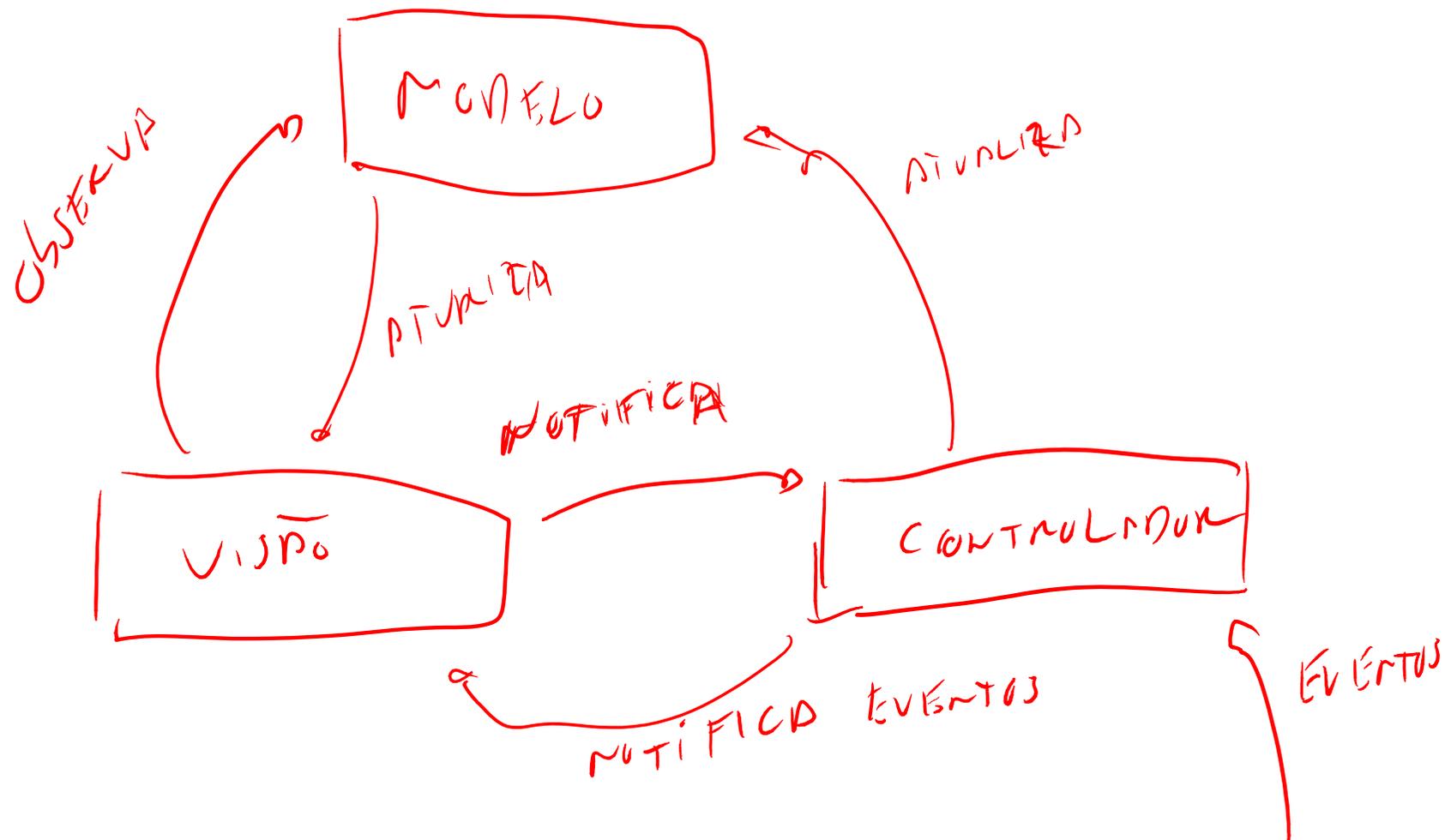
# Model-View-Controller (MVC)

---

- O MVC é o principal padrão para estruturação de aplicações com interfaces gráficas
- Ele separa a aplicação em três grandes partes:
  - O *modelo* representa os dados da aplicação, e implementa a sua lógica interna de uma maneira o mais independente dos detalhes da interface quanto possível
  - A *visão* é a parte visível da interface, e é como o usuário enxerga os dados do modelo
  - O *controlador* faz a mediação entre o usuário, o modelo e a ~~visão~~ visão

# MVC em um diagrama

---



# MVC no Editor Gráfico

---

- Em nosso editor gráfico, as visões serão os botões, os sliders, a paleta de cor e a área de desenho
- Todas as visões são instâncias da interface Componente, que é como elas interagem com o controlador, [mas algumas implementam outras interfaces para receber eventos do modelo] (opcional)
- Essas outras interfaces são observadores do modelo
- O controlador são a classe Editor e as instâncias de Acao ligadas a cada botão

Observador Canvas

# O modelo do Editor

---

- O modelo do Editor Gráfico são as instâncias das interfaces Estado, Figura e Comando, além de uma classe para coordenar os estados e guardar os dados
- As instâncias de Estado implementam as operações do editor: mudar o modo de edição, selecionar uma figura para iniciar uma operação, fazer uma operação parcial na figura selecionada, terminar a operação corrente, desfazer e refazer
- Os dados são o modo corrente, a figura atualmente selecionada, a lista de figuras, a cor atual, e as pilhas de comandos

# Aplicações sem modelo

---

- Para uma aplicação simples como essa, poderíamos ter dispensado o modelo, como fizemos com o Breakout
- Ainda valeria a pena organizar a aplicação em componentes gráficos, figuras, comandos, estados e ações, mas o “modelo” estaria fundido ao “controlador”
- Usamos o MVC quando queremos trocar complexidade por flexibilidade: quando bem arquitetado, o modelo pode ser reaproveitado
- Vamos ver isso na prática usando nossos modelos em uma aplicação com uma camada visão-controlador radicalmente diferente