

Computação II (MAB 225)

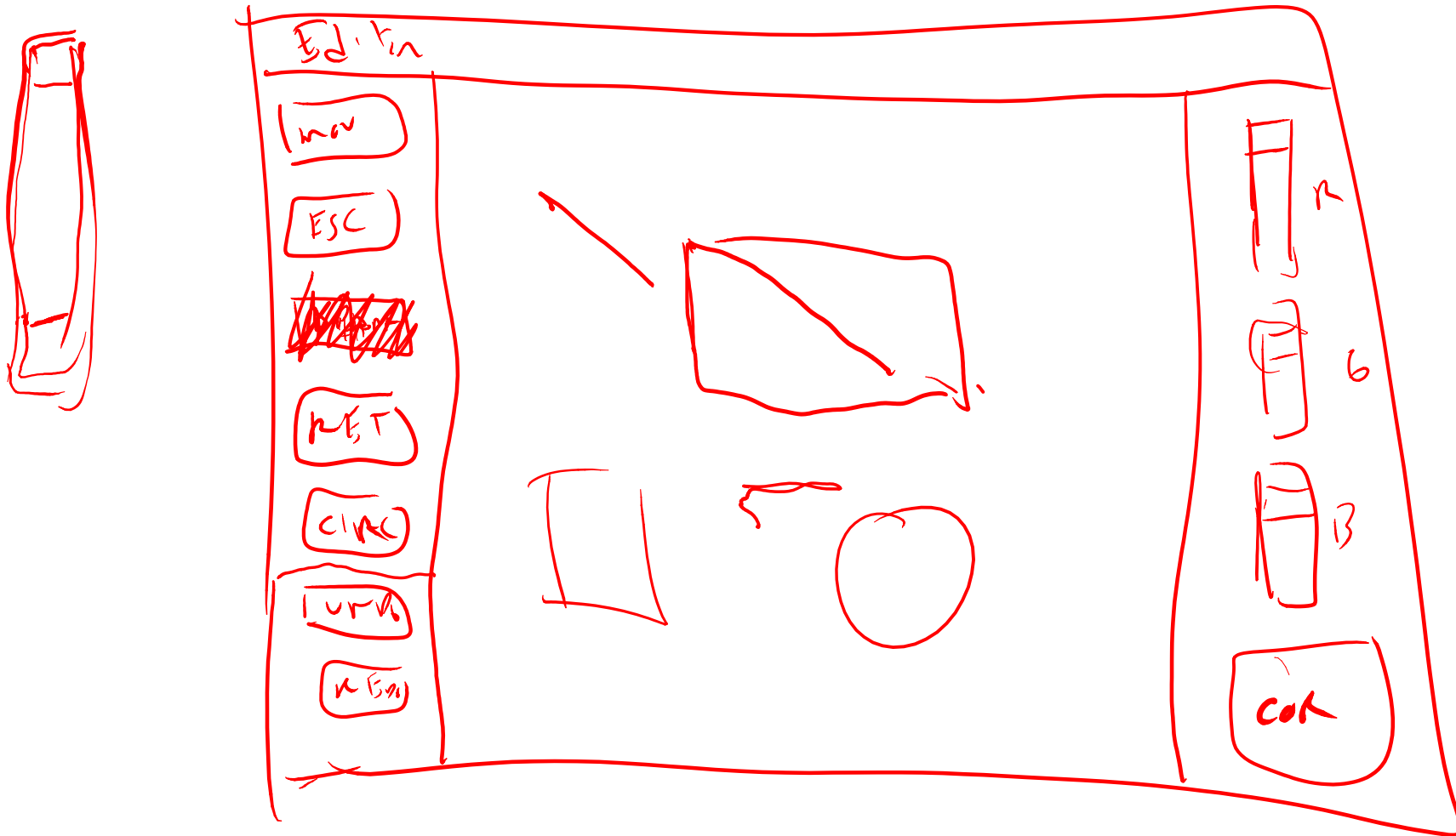
Fabio Mascarenhas - 2015.1

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

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
- 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

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 serão instâncias da interface Componente, que é como elas interagem com o controlador
- O controlador é formado pela classe principal (Editor), e é uma instância da interface App, que é a interface Jogo ampliada com eventos do mouse
- O modelo é um conjunto de classes que representa uma coleção de figuras

Componentes

- A interface `Componente` é formada por quatro campos, `x1`, `y1`, `x2` e `y2`, que dão as *bordas* componente
- Usamos essas coordenadas para saber se um clique do mouse está dentro do componente ou não
- Também temos um método `desenhar(tela)`, para o componente se desenhar
- Finalmente, temos quatro métodos, `clicou`, `apertou`, `solto` e `arrastou`, que dão os eventos do mouse
- Cada um desses métodos recebe as coordenadas `x` e `y` do evento

Botões

- Um botão é nosso componente mais simples
- Quando um botão é clicado, ele deve executar uma *ação*: essa ação é instância de uma interface bem simples, com um único método executar sem parâmetros
- O botão também monitora quando o mouse é apertado ou solto, para fornecer *feedback* visual ao usuário de que ele está sendo clicado
- O controlador despacha cliques do mouse para o botão apropriado

Canvas

- O canvas é uma área de desenho com uma borda
- Assim como botões despacham cliques para objetos ação, o canvas despacha eventos para um *observador*
- Esse observador desenha no canvas, e é avisado quando o mouse é arrastado no mesmo
- O canvas avisa o observador que ele deve desenhar nele com o método `desenhar`, e avisa sobre um arrasto do mouse com os métodos `inicio`, `meio` e `fim`

O modelo do Editor

- O modelo do Editor Gráfico é formado por uma classe principal, mais instâncias de classes que dão o *modo de edição*, e de instâncias de classes que representam figuras
- O modo de edição diz o que acontece quando o usuário começa uma ação, quando continua essa ação, e quando termina essa ação
- Vamos começar com três modos simples, para desenhar um retângulo, para mover uma figura tela, e para apagar uma figura
- As figuras são representadas por uma interface bem abstrata, com métodos para descobrir se um ponto está dentro da figura, para mover a figura e para desenhá-la em um canvas