

Introdução à Programação C

Fabio Mascarenhas - 2014.2

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

Chopper Drop

- Como mais um exemplo do uso de laços e vetores, vamos adaptar outro jogo simples: *Chopper Drop*
- <http://www.donpixel.com/play/060612191956/>
- Um helicóptero atravessa a tela da esquerda pra direita
- A cada passada ele pode soltar uma bomba, essa bomba destrói os prédios
- Na próxima passada ele está mais baixo, se ele colidir com um prédio o jogo acaba

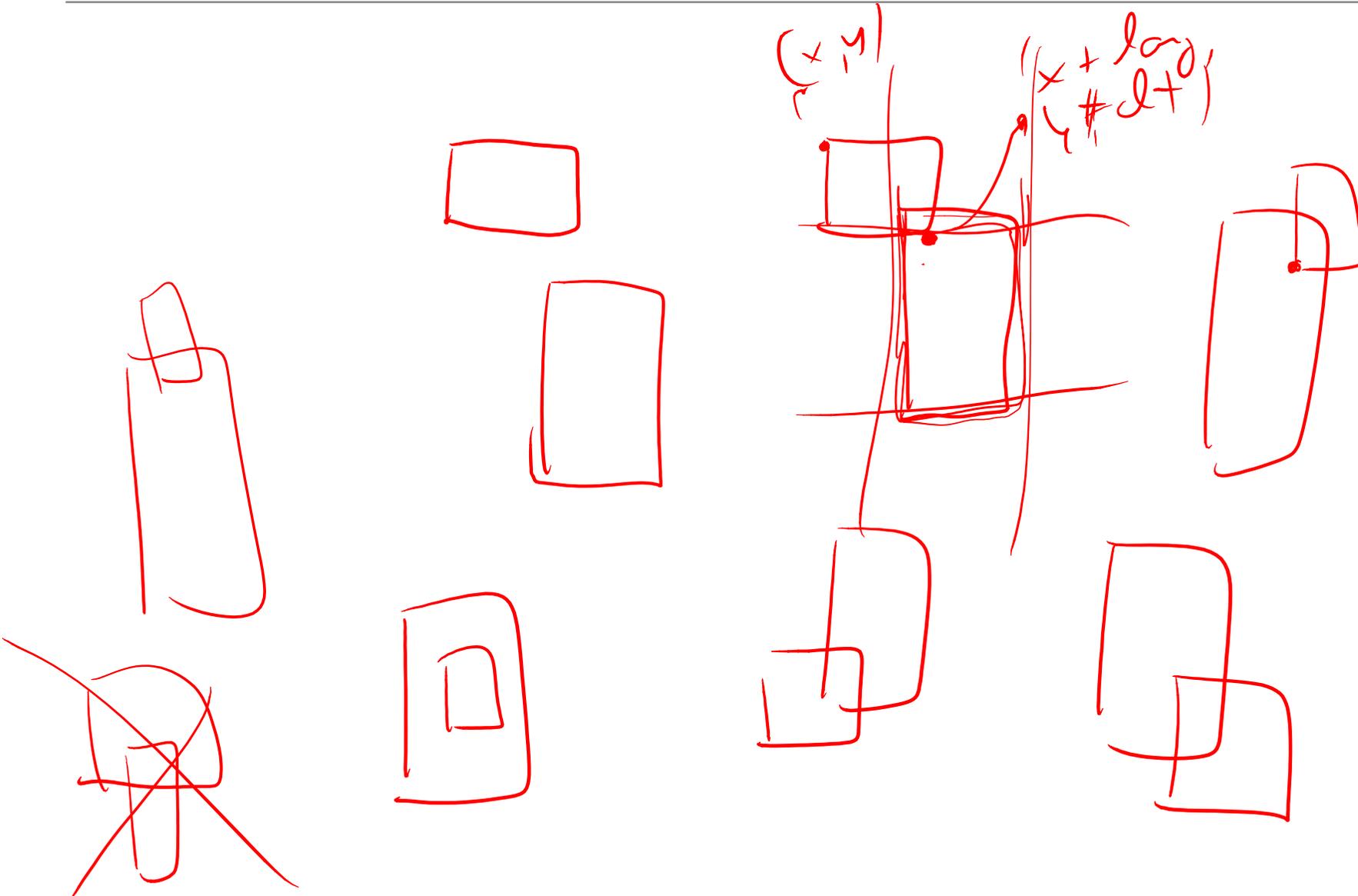
Partes do jogo

- Helicóptero – posição (horizontal e vertical)
- Bomba – posição
- Score – valor atual
- Prédios – formados por blocos, vamos fazer cada prédio ter uma altura em blocos e uma cor
- Mecânica do jogo: mover helicóptero, mover bomba (se já foi solta), verificar colisão da bomba com prédios, ou do helicóptero com prédios

Prédios

- Vamos usar vários vetores para representar os prédios, já que cada prédio tem vários atributos que temos que controlar
 - Cores – um vetor para cada componente
 - Altura – um vetor, quando a bomba destrói um prédio essa altura cai, se ela é zero é que o prédio foi totalmente destruído
 - Posição – a posição horizontal poderia ser calculada a cada vez que formos desenhar, mas é mais simples calcular ela de uma vez e armazenar em um vetor
 - Telhado – se o prédio tem um telhado triangular ou retangular

Colisão de retângulos



Várias bombas

- Uma extensão ao jogo (que torna ele mais fácil) é permitir várias bombas a cada passada
- Podemos implementar isso usando vetores para cada atributo das nossas bombas
- Para lançar uma bomba vemos se ainda tem alguma que não foi lançada
- Um novo atributo dá o “poder destrutivo” da bomba: a primeira bomba é a mais poderosa, as outras destroem um número menor de blocos

Laço for

- O laço for em C é uma forma mais compacta de escrever um laço while:

```
for(<inicial>; <condição>; <atualização>) {  
    <comandos>  
}
```

- O comando <inicial> é executado antes da primeira iteração; a cada iteração o valor da <condição> é calculado, se for diferente de zero os <comandos> são executados, depois o comando <atualização> é executado, e começa uma nova iteração

- Isso corresponde ao seguinte comando while:

```
<inicial>;  
while(<condição>) {  
    <comandos>  
    <atualização>;  
}
```