



Programação Gráfica – Parte 3

Versão em C – 2006 – PUCPR – Tutoria de Jogos – 1º Ano

Paulo V. W. Radtke

pvwradtke@gmail.com

<http://www.ppgia.pucpr.br/~radtke/jogos/>



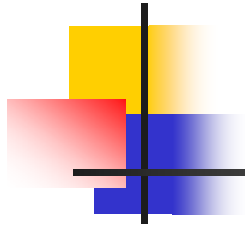
AVISO IMPORTANTE!!

- Esta versão é dedicada exclusivamente para os cursos de **Ciência da Computação** e **Engenharia da Computação**.
- Para a versão de **Sistemas de Informação**, utilizando *Java*, pegue o arquivo correspondente e participe da aula no horário adequado.



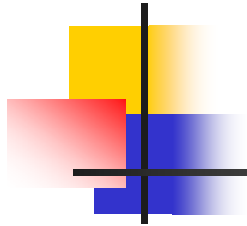
Entrega da 2ª Parcial

- Datas importantes:
 - 12 a 17 de Junho.
 - Relatório impresso contendo:
 - Código fonte do protótipo da interface.
 - Impressão no relatório dos recursos gráficos da fase/jogo (*tilemaps*, sprites, cenários de fundos, protótipo, etc).
 - Discussão do uso dos recursos com a lógica do jogo no terceiro bimestre.
 - Defesa em laboratório do protótipo e entrega do relatório com a equipe completa.



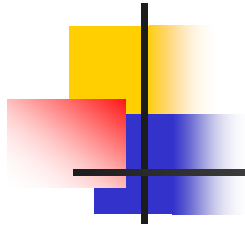
Entrega da 2ª Parcial

- Por recursos gráficos, entende-se que neste bimestre já teremos:
 1. TODOS os sprites necessários para o demo.
 2. TODAS as telas de fundo/tilemaps.
 3. TODAS as fontes.
- Logo, espera-se que o relatório inclua TODOS estes elementos.



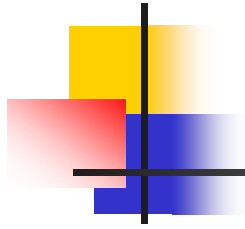
Conteúdo

- Colisão de Sprites
- Desenhando um Tilemap
- Movimentando um Tilemap



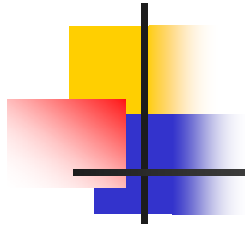
Colisão de Sprites

- A colisão de sprites ocorre quando dois sprites sobrepõe a sua área desenhada.
- Por área desenhada, entende-se os pontos opacos do sprite.
- A detecção trabalha em conjunto com a lógica do jogo, não devendo ser tratada como algo automático.



Colisão de Sprites

- Logo, o seu jogo irá decidir se ele testa a colisão e de quais sprites se testará a colisão.
- A Chien2DLite possui um método específico para testar se dois sprites colidiram.
- Este método é o **colidiuAzulejos**.



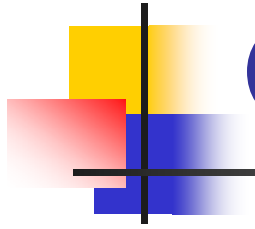
Colisão de Sprites

- O método testa a colisão ponto-a-ponto dos sprites.
- Esta detecção é acelerada por uma máscara de bits.
- Além disso, a colisão é testada apenas com *sprites* 32x32. Outros tamanhos não são suportados.



Colisão de Sprites

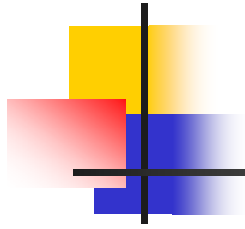
- `bool colidiuAzulejos(int identificador1, int indice1, int x1, int y1, int identificador2, int indice2, int x2, int y2):`
 - **identificador**: identificador dos sprites no sistema de vídeo.
 - **indice**: qual bloco dentro do azulejo será testado.
 - **x** e **y**: coordenadas dos sprites na tela.



Colisão de Sprites

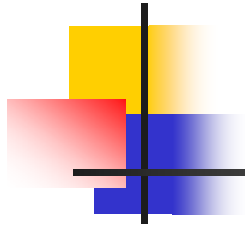
- Uso típico:

```
if(video->colidiuAzulejos(sprite1, 0, x1,  
    y1, sprite2, 3, 200,300))  
{  
    // Código para tratar a colisão  
    ...
```



Colisão de Sprites

- O mais importante é lembrar:
 - Este método testa a colisão de elementos 32x32. Logo, um jogo real testaria se a arma do personagem tocou um inimigo, usando este método.
 - Nem todos os jogos precisam testar a colisão de sprites.



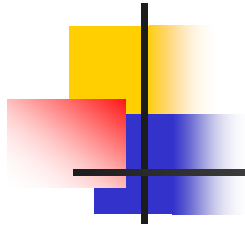
Exercício 01

- NO exercício anterior, tínhamos desenhado na tela um sprite fixo, além de um sprite controlado pelo teclado.
- Teste a colisão nestes dois sprites.
- Caso ocorra a colisão na tela, desenhe um texto indicando isto.



Desenhando um Tilemap

- Para desenharmos um *tilemap* utilizamos a biblioteca **Chien2DMappy**.
- Esta biblioteca utiliza a **Chien2DLite** para desenhar um mapa gerado pelo *Mappy* na tela.
- Por tela, entende-se aqui a tela TODA.



Desenhando um Tilemap

- Para isto, precisamos de 2 coisas:
 1. Arquivo com os blocos da imagem.
 2. Arquivo com o mapa gerado pelo Mappy, em um dos formatos:
 - Formato FMP.
 - Formato FMA.



Desenhando um Tilemap

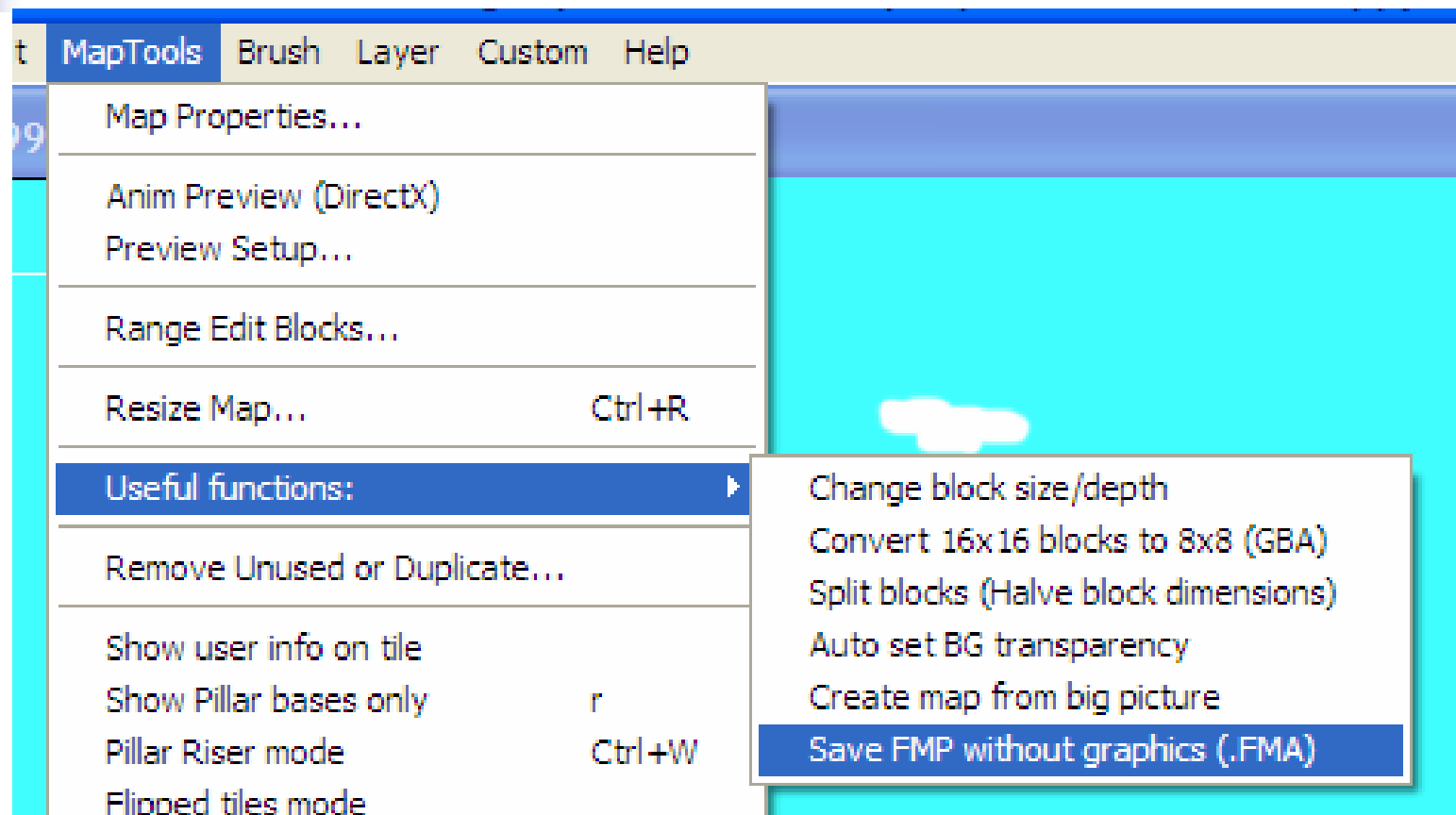
- O formato FMP é mais completo e inclui os gráficos. Porém, por falta de flexibilidade, este formato é muito recomendado para a Chien2DMappy.
- O formato FMA possui apenas os dados do tilemap, sem a imagem, gerando um arquivo menor.

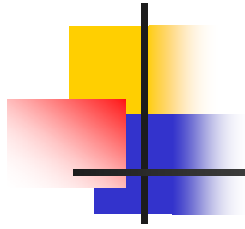


Desenhando um Tilemap

- Para gerar o arquivo FMA, basta abrir o mapa e escolher a opção “MapTools”, “Useful functions”, “Save FMP without graphics (FMA)”.
- A vantagem de separar o gráfico do tilemap no FMA é de reutilizar o mesmo tileset em diversos tilemaps.

Desenhando um Tilemap





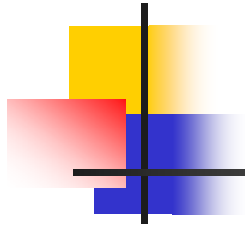
Desenhando um Tilemap

- **Dica:** durante o desenvolvimento, utilize o arquivo FMP, já que será comum você fazer mudanças frequentes no mapa.
- Apenas para a distribuição você gera o FMA a partir do FMP.
- **Nota:** você DEVE guardar o FMP original para referência.



Desenhando um Tilemap

- Para este exemplo, utilizaremos o arquivo **c-aula07-plataforma.zip**.
- O arquivo contém o tilemap (plataforma.FMP e plataforma.FMA) e o tileset (plataforma.png).
- Extraia o arquivo no diretório do projeto.
- Sobrescreva os arquivos **chien2dmappy.h** e **chien2dmappy.cpp**, estas versões no arquivo são novas e possuem duas correções importantes.



Desenhando um Tilemap

- O primeiro passo para se carregar um tilemap é carregar o seu tileset.
- Na Chien2DLite, o tileset é carregado como sprites (um conjunto de azulejos).
- `unsigned int tileset = video->carregaAzulejo("plataforma.png", "Tileset do cenario", Chien2DLite::azulejo32);`



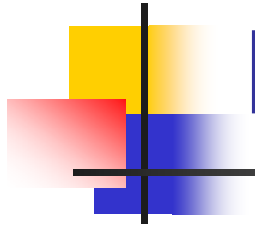
Desenhando um Tilemap

- Em seguida, criamos um objeto da classe Chien2DMappy:

```
Chien2DMappy *mapa = new Chien2DMappy();
```

- Eventualmente, quando este objeto não for mais necessário, devemos eliminá-lo explicitamente:

```
delete mapa;
```

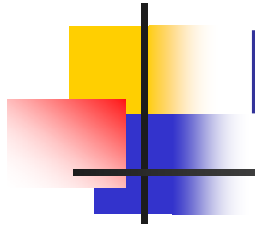


Desenhando um Tilemap

- Para carregar o mapa, utilizamos o método:

bool carregaMapa(string arquivo);

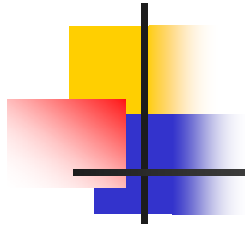
- O método recebe o nome do arquivo FMP/FMA a carregar.
- Ele retorna **true** ou **false** se conseguiu ou não carregar o mapa.



Desenhando um Tilemap

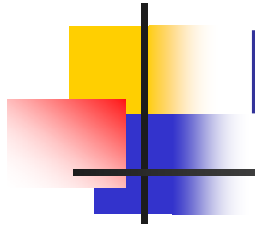
- Exemplo de uso do carregaMapa:

mapa->carregaMapa("plataforma.FMA");



Desenhando um Tilemap

- Para desenhar o tilemap, utilizamos a função:
- *void desenhaCamadaMapa(unsigned int idAzulejo, int camada, int x, int y);*
- Esta função desenha uma camada do mapa na tela, utilizando um tileset indicado por **idAzulejo** nas coordenadas **x** e **y** (internas do mapa).



Desenhando um Tilemap

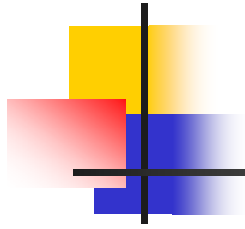
- Exemplo de uso típico:

*mapa->desenhaCamadaMapa(tileset, 0,
0, 0);*



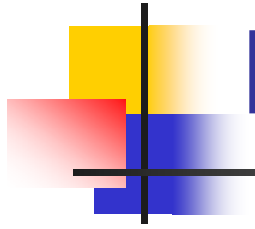
Desenhando um Tilemap

- Se um mapa possuir várias camadas, devemos desenhá-las uma a uma para gerar a imagem final.
- A Chien2DMappy não faz isso automaticamente para permitir efeitos especiais dependentes do jogo:
 - Scroll em parallax (camadas).
 - Cenários em planos superiores à ação (como em RPGs).
 - Uso de camadas do mapa para controlar os eventos do jogo (*triggers*).



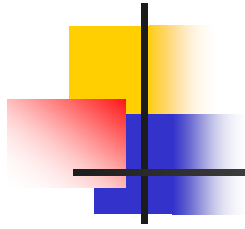
Exercício 02

- Modifique o exercício anterior para carregar o mapa "plataforma.FMA" e o seu tileset, "plataforma.png".
- Ao invés de desenhar a tela de fundo, desenhe o tilemap na coordenada **0,0**.



Movimentando um Tilemap

- O método `desenhaCamada` pode desenhar outras coordenadas dentro do mapa.
- Porém, este somente desenha coordenadas válidas, isto é, coordenadas em que o mapa não cubra a tela toda não são desenhadas.



Movimentando um Tilemap

- Para determinar as dimensões de um mapa para animar a movimentação do cenário, utilizamos o método:
- *bool dimensoesMapa(int *largura, int *altura, int *dimensaoAzulejo, int *camadas);*



Movimentando um Tilemap

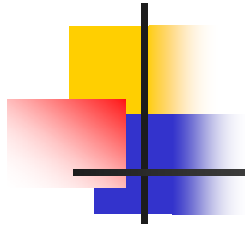
- O método retorna **true** ou **false** se ele pôde determinar as dimensões do mapa (isto é, se o mapa foi ou não carregado).
- AS dimensões do mapa são retornadas por ponteiros.
- Os valores da **altura** e **largura** indicam quantos blocos o mapa tem nessa dimensão.
- **dimensaoAzulejo** indica, em pixels, o tamanho do bloco que desenha o cenário.
- **camadas** indica o número de camadas geradas pelo Mappy.



Movimentando um Tilemap

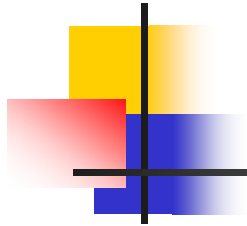
- Uso típico:

```
// Determina as dimensões do mapa  
int largura, altura;  
int tamBloco, camadas;  
mapa->dimensoesMapa(&largura,  
    &altura, &tamBloco, &camadas);
```



Movimentando um Tilemap

- O tamanho do mapa em pixels é determinado pela largura dele em blocos e o tamanho do bloco de cenário.
- Sabendo estes valores, podemos utilizá-lo para movimentar a tela dentro do mapa.



Desafio

- Modifique o exercício anterior para determinar as dimensões do mapa.
- Utilize estes valores para movimentar o mapa na horizontal automaticamente.