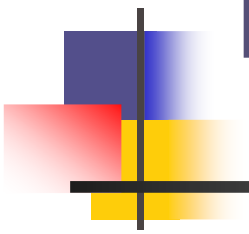


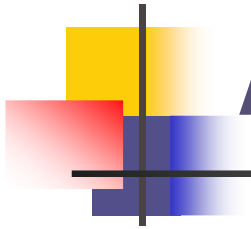
Programação Gráfica – Parte 4

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

Paulo V. W. Radtke

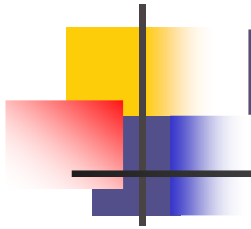
pvwradtke@gmail.com

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



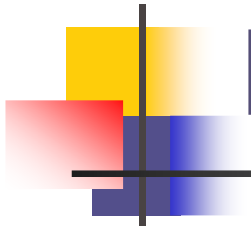
AVISO IMPORTANTE!!

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



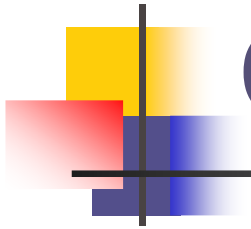
Entrega da 2ª Parcial

- 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.



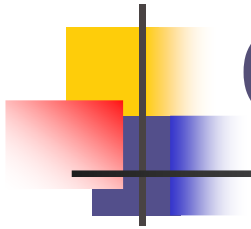
Entrega da 2ª Parcial

- 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.



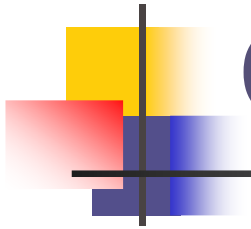
Conteúdo

- Colisão de Sprites
- Desenhando um Tilemap



Colisão de Sprites

- Em geral, os sprites no jogo são utilizados como elementos independentes.
- Assim, cada sprite mostrado na tela é um objeto em memória.

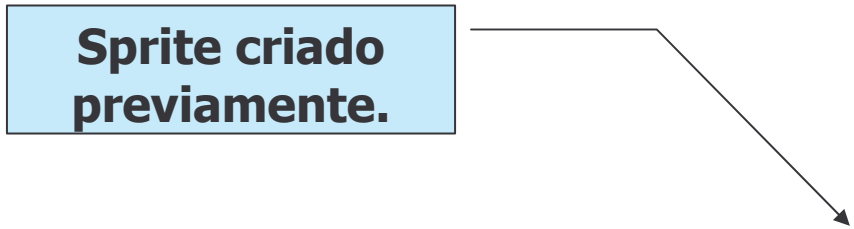


Colisão de Sprites

- Na aula passada, utilizamos um construtor da classe Sprite, que cria um sprite a partir de uma imagem no JAR.
- Existe outro construtor, que cria um Sprite a partir de outro objeto do tipo Sprite em memória.

Colisão de Sprites

- Assim, mantemos em memória um sprite que serve de modelo para criarmos os sprites utilizados pelo jogo.
- Exemplo:



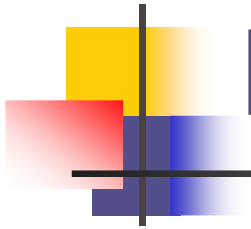
Sprite spr2=new Sprite(spr);

Colisão de Sprites

- Finalmente, para testarmos se dois sprites colidiram, utilizamos o método **collidesWith**, que retorna **true** se colidiu ou **false** caso contrário.
- Exemplo:

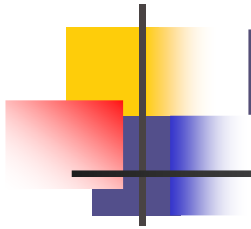
Indica se a colisão é testada ponto a ponto, ou por retângulo.

spr.collidesWith(spr2, true)



Exercício 1

- Utilizando o arquivo ***j2me-exemplo05-TestaSprite.zip***, crie o projeto **TestaSprite** no KToolBar.
- Modifique o código fonte para testar a colisão dos sprites **spr** e **spr2**.
- Quando ocorrer a colisão, indique através de um texto na tela.

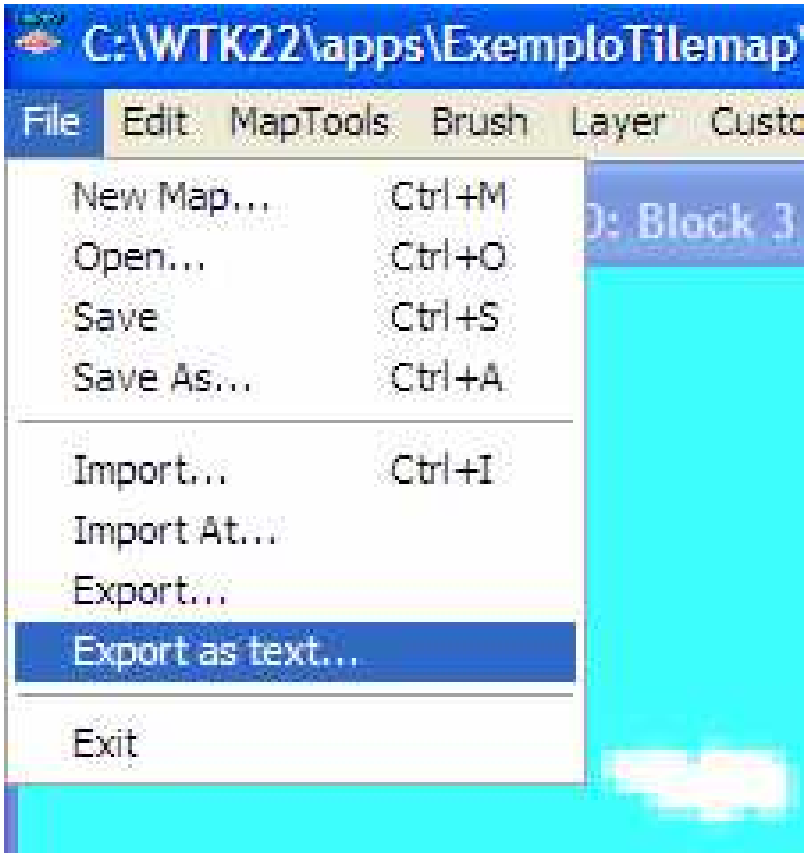


Exportando um Tilemap

- Em J2ME, o Tilemap é representado pela classe **TiledLayer**.
- Para que possamos mostrar um tilemap corretamente, precisamos converter o arquivo do Mappy.
- Um formato aceito pelo **TiledLayer** é a declaração do tilemap como um vetor de inteiros.

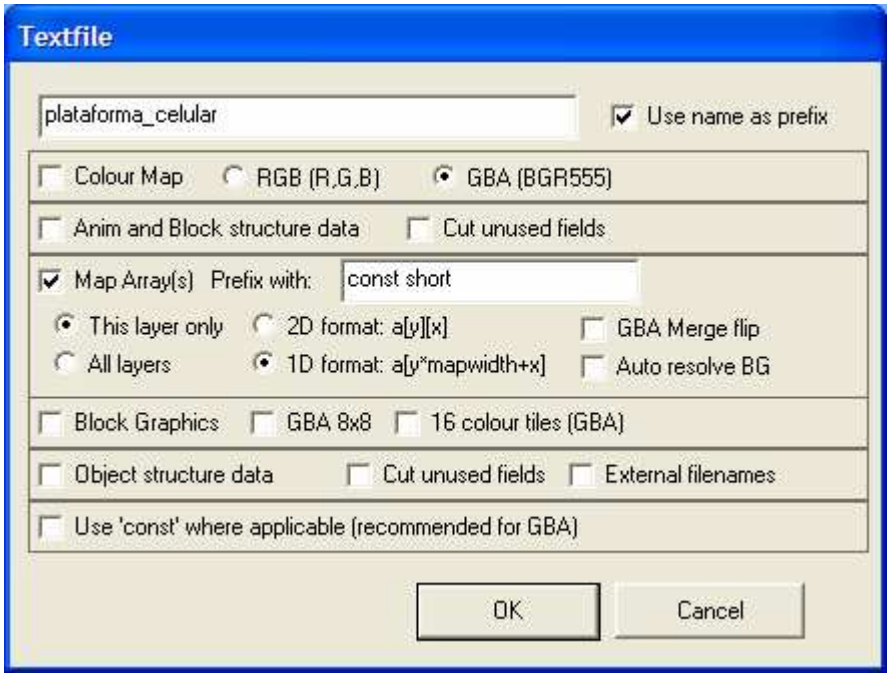
Exportando um Tilemap

- Selecione o menu **File**.
- Neste menu, selecione a opção **Export as Text**.



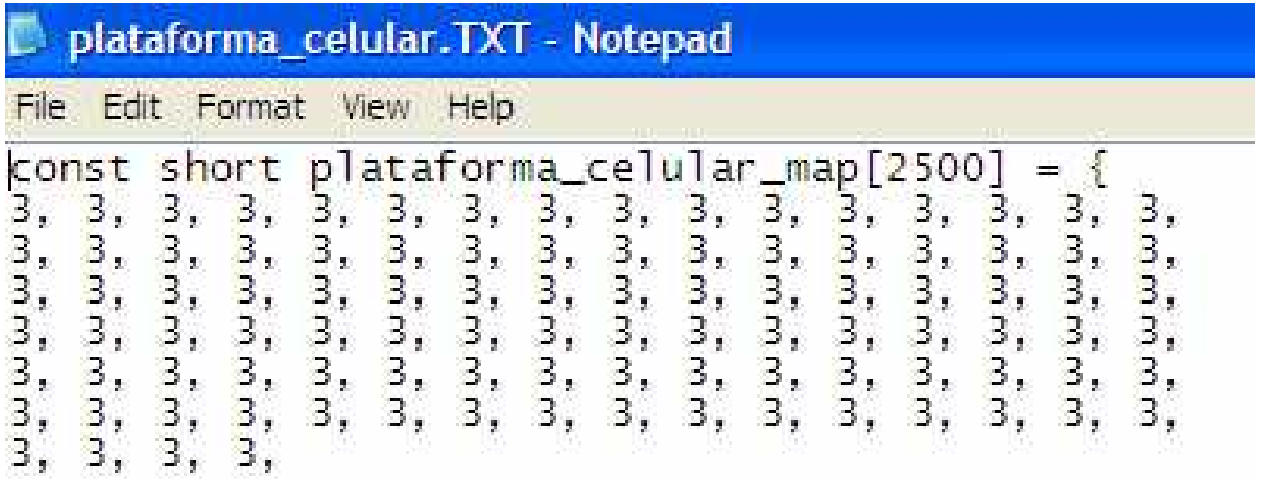
Exportando um Tilemap

- Selecione as opções como na exemplo ao lado.
- Isto irá gerar um arquivo txt com o nome do tilemap.
- O arquivo é gerado na pasta do arquivo FMP original.



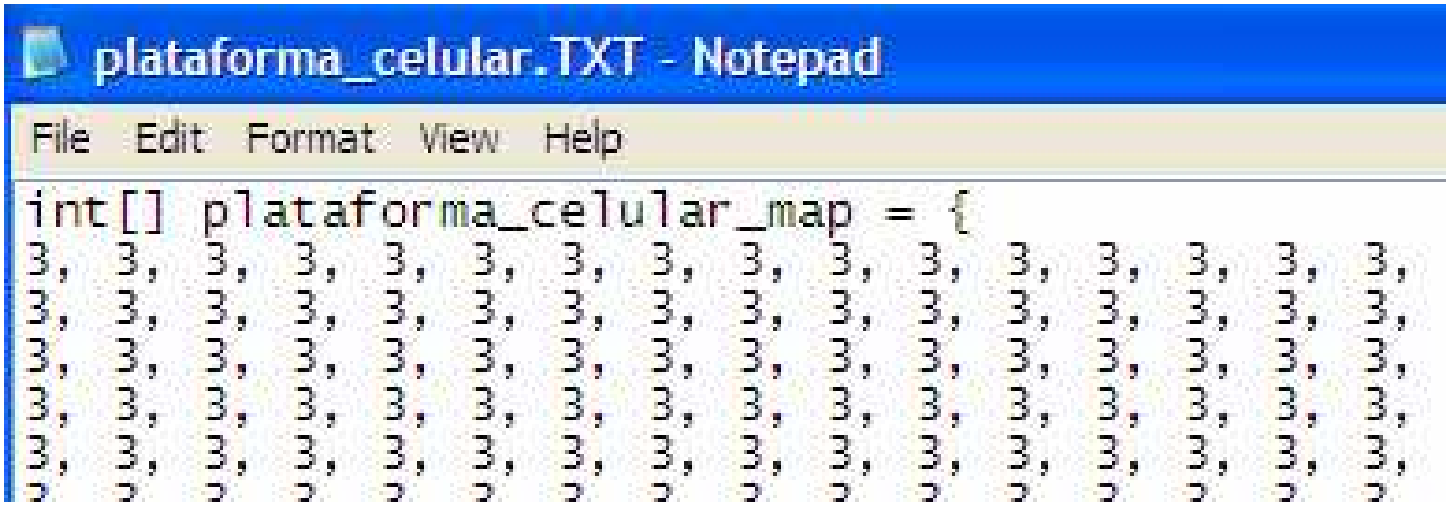
Exportando um Tilemap

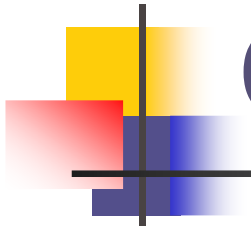
- A declaração do vetor é adequada para programas em C/C++.



Exportando um Tilemap

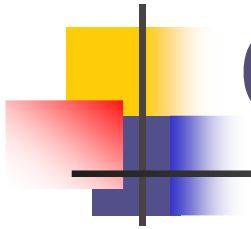
- Assim, basta mudar a declaração para um vetor de inteiros Java e copiar a declaração para o seu código Java:





O LayerManager

- Para desenharmos um tilemap e sprites, usaremos daqui em diante a classe **LayerManager**.
- Esta classe permite controlar automaticamente o redesenho de elementos como sprites e tilemaps.

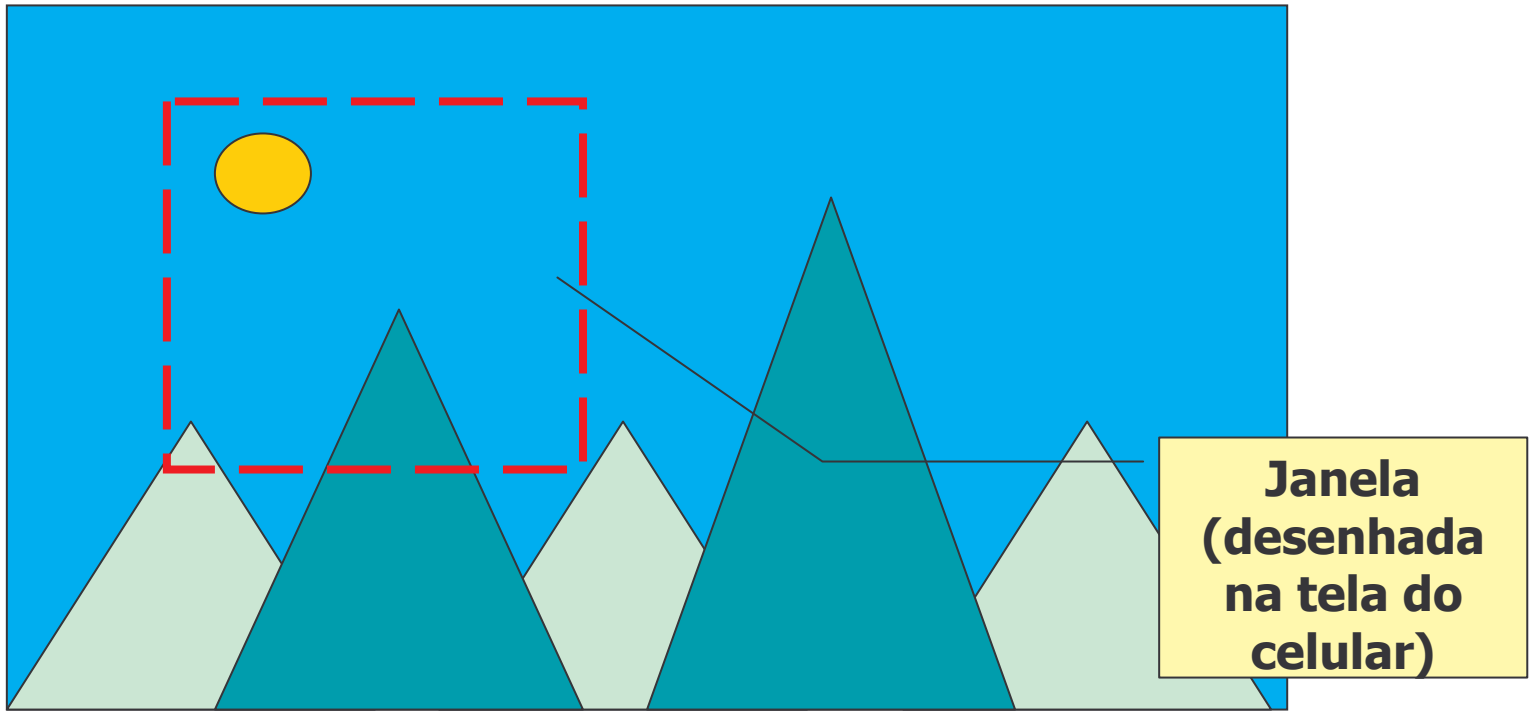


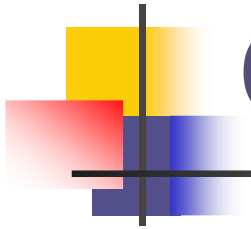
O LayerManager

- A grande vantagem de um **LayerManager** é gerenciar automaticamente uma tela virtual.
- Assim, todos os elementos são desenhados em coordenadas absolutas DENTRO da tela virtual.

O LayerManager

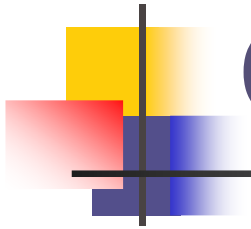
- Exemplo de imagem virtual:





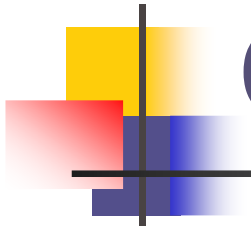
O LayerManager

- Na hora de mostrar a imagem, o **LayerManager** converte as coordenadas virtuais para coordenadas de tela.
- Ganha-se assim a flexibilidade na hora de manipular elementos no mundo, sem se preocupar com conversões do *tilemap* para a tela.



O LayerManager

- (a partir de agora, referenciaremos o **arquivo j2me-exemplo05-ExemploTilemap.zip**)
- Para criar um LayerManager, devemos primeiramente declarar um atributo no GameCanvas.



O LayerManager

- Criando o LayerManager:

gerente = new LayerManager();

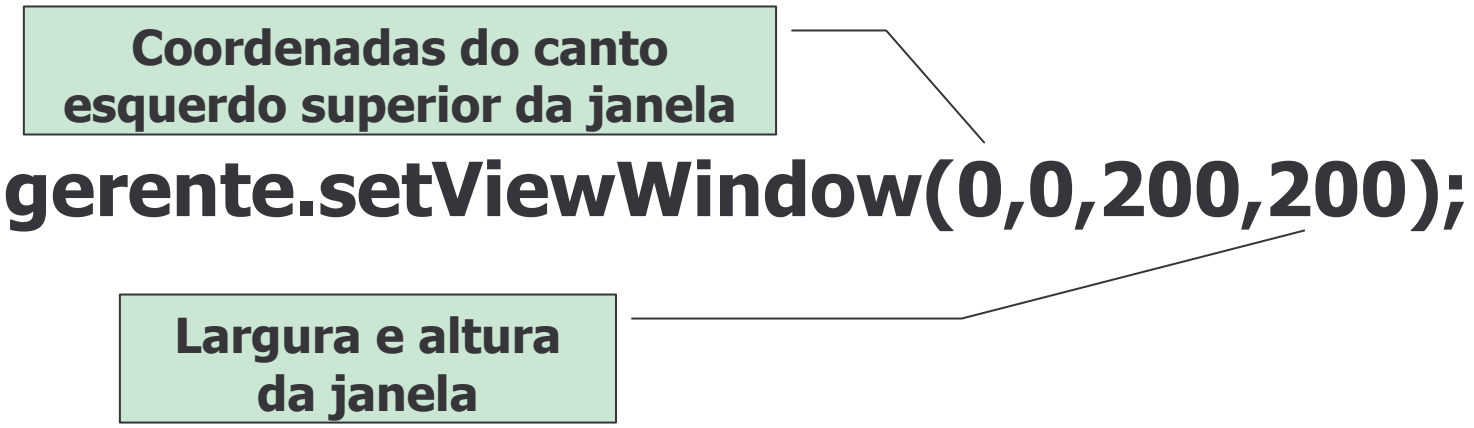
- Para adicionar um elemento ao LayerManager:

gerente.append(spr);

- *Obs: o último elemento inserido é o PRIMEIRO a ser desenhado.*

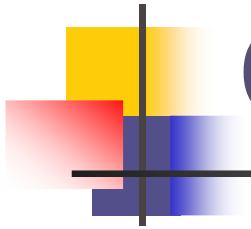
O LayerManager

- Durante o redesenho, podemos definir a coordenada do canto esquerdo superior da tela desenhada no celular dentro da tela virtual:



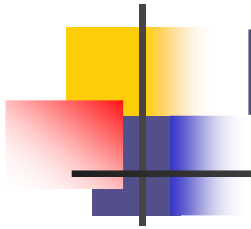
O LayerManager

- Para desenhar os elementos associados ao **LayerManager**, basta indicar qual a coordenada dentro da tela do celular o mesmo deve ser desenhado.



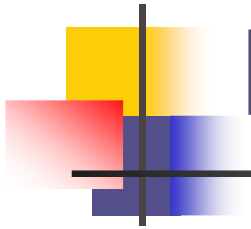
O LayerManager

- Esta técnica é utilizada principalmente para fazer o layout de tela para resoluções diferentes.
- O jogo em si é desenhado através do **LayerManager**, enquanto que os elementos decorativos, como placar, vidas, etc, são desenhados à parte.



Desenhando um Tilemap

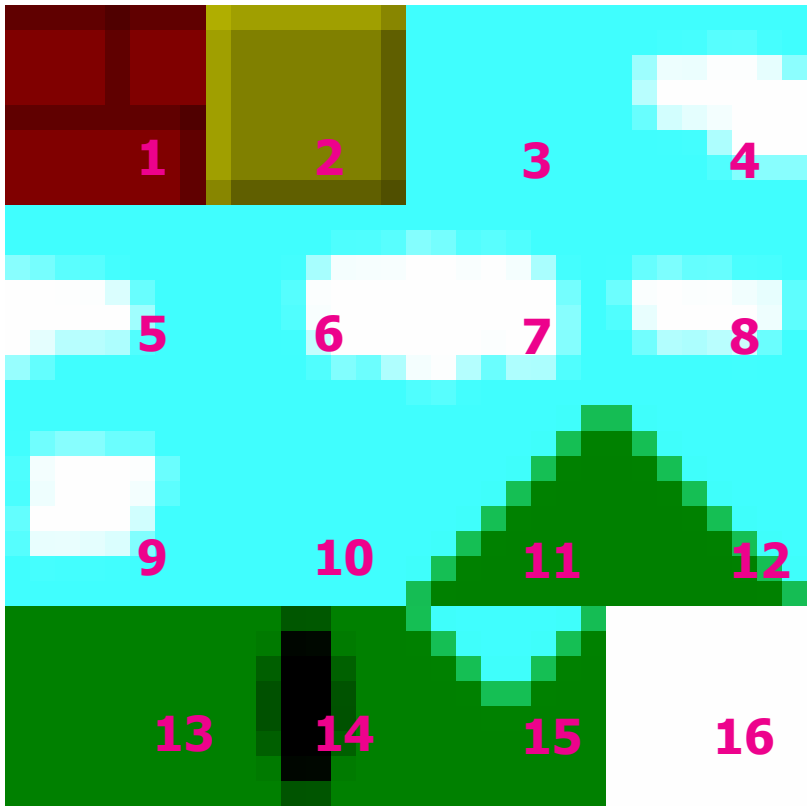
- Para criar o tilemap, precisamos de duas coisas:
  1. O tileset (gráfico em blocos)
  2. O tilemap para inicializar o tiled layer.

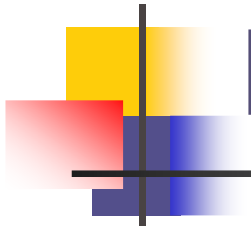


Desenhando um Tilemap

- Os blocos são numerados a partir de 1 dentro da imagem.
- O bloco zero é o bloco transparente.
- Assim, mapas gerados pelo Mappy são 100% compatíveis com J2ME.

Desenhando um Tilemap

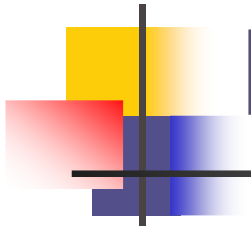




Desenhando um Tilemap

- Para criar o TiledLayer, precisamos de um dado deste tipo, que será adicionado ao **LayerManager**.
- O construtor recebe as dimensões (em blocos) do mapa, a imagem dos blocos e o tamanho destes.

mapa = new TiledLayer(100,25, temp, 8,8);



Desenhando um Tilemap

- Para iniciar o mapa com 100x25 blocos, utilizamos o seguinte código:

```
int linha, coluna;
for(int i=0;i<2500;i++)
{
   coluna = i%100;          // Cada linha tem 100 blocos
   linha = i/100;           // Idem
   mapa.setCell(coluna, linha,
   plataforma_celular_map[i]+1);
}
```

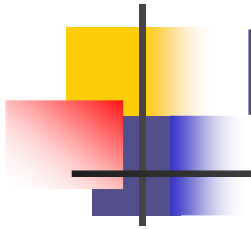
Exemplo do Cálculo Anterior

Linhas: 5, Colunas: 10

25%10=5
Coluna: 5

25/10=2
Linha: 2

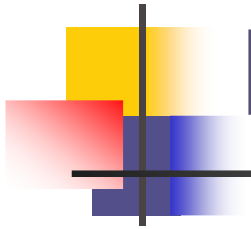
	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	10	11	12	13	14	15	16	17	18	19
2	20	21	22	23	24	25	26	27	28	29
3	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49



Desenhando um Tilemap

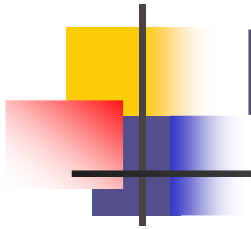
- Em seguida, colocamos o tiled layer no no LayerManager via um append:

gerente.append(mapa);



Desenhando um Tilemap

- Durante o redesenho (no método run), o gerente desenha automaticamente o tilemap.



Próxima Aula

- Desenvolvimento do protótipo.