

# Projeto de Jogos

## Parte II – Gráficos

Paulo V. W. Radtke  
pvwradtke@gmail.com  
<http://www.ppgia.pucpr.br/~radtke/jogos>



## Conteúdo

- Introdução
- Vídeo – Considerações (PC e celular)
- O Mundo em Blocos de Imagem
- Sprites e Animação
- Formatos de Arquivo



## Introdução



- Os recursos de áudio e vídeo são o componente fundamental da interface do jogo com o usuário.
- Através da imagem o usuário tem o retorno visual do mundo representado em memória do jogo.
- Através do áudio podem ser fornecidas pistas da ação, aumentando o grau de imersão.

PUCPR - CCET

3

## Introdução



- Ambos os recursos podem ser utilizados para tornar a experiência mais interessante ou agradável.
- Efeitos visuais podem tornar o jogo mais bonito e interessante de se ver.
- Música pode tornar a atmosfera do jogo mais completa.

PUCPR - CCET

4

## Introdução

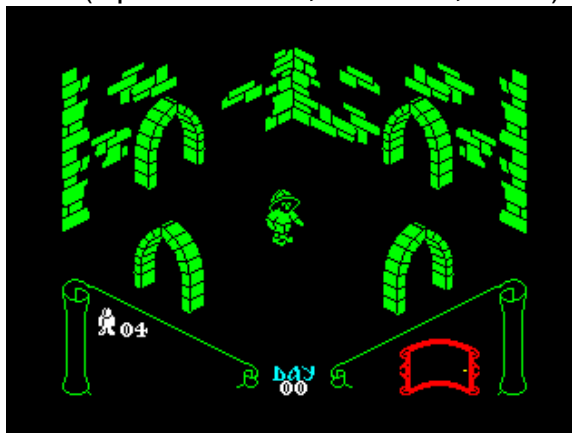
- Problemática:
  - Limite de hardware.
  - Escopo do jogo.
  - Custo de produção dos recursos.
- A combinação destes fatores influencia diretamente a apresentação áudio-visual do jogo.

PUCPR - CCET

5

## Introdução

- Knightlore (Spectrum 48k, Ultimate, 1983):



PUCPR - CCET

6

## Introdução

- Half-Life 2 (Windows PC, Valve, 2004)



PUCPR - CCET

7

## Introdução

- Tanto o Knightlore como o Half-Life 2 foram jogos que redefiniram os padrões de áudio-visual no seu lançamento.
- O hardware da época ditou o limite atingido por cada um dos jogos.

PUCPR - CCET

8

## Introdução



- Regras de ouro:
  - Bons gráficos não fazem um bom jogo.
  - Música e efeitos sonoros devem fazer parte da experiência, mas nunca se tornam a experiência em si.
- **Objetivo: encontrar o balanceamento adequado entre investimento em visual e áudio frente ao investimento no conceito e implementação.**

## Introdução

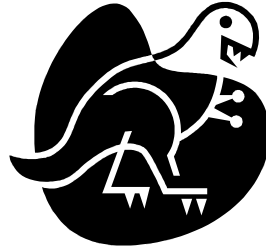


- Foco desta tutoria: jogos 2D.
- Técnicas analisadas:
  - Geração de imagens a partir de tilemaps.
  - Animação de objetos em sprites.

## Vídeo – Considerações (PC)



- Modos de vídeo em *palette* (256 ou 16 cores):



- Caminho: modo *true color* ou *high color*.

## Vídeo – Considerações (PC)

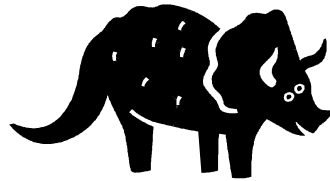


- Motivação da escolha do modo de cores:
  - Aceleração por hardware disponível em diversas situações.
  - Processadores e memórias suficientemente rápidas para trabalhar com grandes quantidades de dados.
  - A maioria das placas aceleradas 3D trabalha com os modos de vídeo com *palette* em modo simulado, logo não é mais rápido utilizá-los.

## Recomendações - Desktop



- Resoluções VGA e VGA ModoX:



- Resolução mínima do Windows XP: 800x600 pixels.
- Mínimo multi-plataforma: 640x480 pixels.

## Recomendações - Desktop



- Motivação da escolha a resolução:
  - Processadores e memórias suficientemente rápidas para trabalhar com grandes quantidades de dados.
  - Suporte de sistemas operacionais para padronizar o acesso a estas resoluções de vídeo.

## Recomendações - Desktop



- Em geral, jogos que não façam uso extensivo de efeitos de imagens translúcidas e rotação não são problema para os sistemas atuais.
- Quanto menor a resolução, menor a complexidade dos gráficos a serem produzidos.
- O hardware atual possibilita atingir taxas de desenho de tela na ordem dos 75 quadros por segundo.

PUCPR - CCET

15

## Recomendações - Celulares



- Celulares são dispositivos de baixa resolução por natureza.
- Os modelos atuais variam de resolução, como 128x160, 160x240, etc.
- Uma resolução “comum” segura é 128x128.
- Outras resoluções dependem de uma família de aparelhos.
- O ideal é escolher uma família de aparelhos e produzir nestas condições.

PUCPR - CCET

16



## Recomendações - Celulares



- Além da baixa resolução, celulares também possuem uma baixa gama de cores no display.
- Em geral, a profundidade de cores varia de 10 a 16 bits (1024 a 65536).
- Logo, nem todas as cores são reproduzidas, sendo necessário escolher as faixas de contraste.

## Recomendações - Celulares



- Celulares tem desempenho reduzido de vídeo, logo a simplicidade dos gráficos torna-se mandatória.
- A taxa de atualização da tela é pequena se comparada com sistemas desktop, logo o jogo deve considerar este aspecto.
- Memória limitada requer uso inteligente de gráficos, e jogos mais complexos devem obrigatoriamente usar *tilemaps*.

## O Mundo em Blocos



- Jogos 2D fazem uso extensivo de uma técnica conhecida como *tile mapping*.
- Nesta técnica, os gráficos são construídos a partir de blocos menores que se encaixam.
- Assim, uma pequena quantidade de gráficos é capaz de gerar uma área extensa de cenário.

PUCPR - CCET

19

## O Mundo em Blocos



King's Valley II ©Konami, 1988

PUCPR - CCET

20

## O Mundo em Blocos

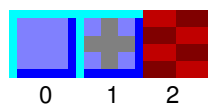


- Cada tela é representada por um mapa que contém uma referência cruzada para um bloco do tile set (conjunto de blocos).
- Ao desenhar a tela, o jogo determina a posição de cada bloco (tile) na tela e usa um índice para referenciar o bloco a ser desenhado.

PUCPR - CCET

21

## O Mundo em Blocos - Exemplo



Tile set (blocos)

2	2	2	2	2	2
2	1	0	0	1	2
2	0	0	0	0	2
2	0	0	0	0	2
2	1	0	0	1	2
2	2	2	2	2	2

Tile map (mapa)

Referência cruzada  
no tile set

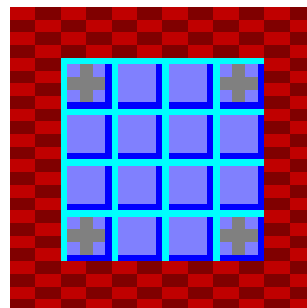


Imagem gerada

PUCPR - CCET

22

## O Mundo em Blocos



- No caso de um jogo com movimentação de tela, um mapa é parcialmente visualizado na tela.
- A ilusão de movimento é dada pelo deslocamento dentro do mapa de um indexador que indica o bloco do canto esquerdo superior da tela (coordenadas padrão).

PUCPR - CCET

23

## O Mundo em Blocos



- Os blocos do tile map podem ser utilizados também para controlar o posicionamento dos elementos.
- Por exemplo, o chão aonde o jogador anda é indicado por alguns índices (referências) de blocos.
- O mesmo vale para escadas, paredes e outros elementos.

PUCPR - CCET

24

## O Mundo em Blocos



- Ao movimentar, o jogo testa o tipo de bloco que o jogador está “pisando” para determinar se o jogador:
  - Anda;
  - Cai.
  - Pára (por andar contra uma parede).
  - Morre (pisar em fogo, espinho ou cair na água).

## O Mundo em Blocos



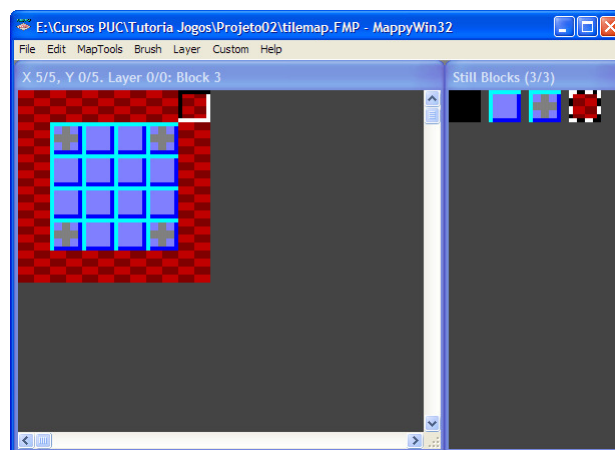
- O tamanho do bloco depende da resolução do sistema alvo e da granularidade desejada.
- Para celulares, blocos 8x8 são um bom compromisso.
- Já para sistemas desktop, 16x16 ou 32x32 são mais adequados pela resolução mais alta.

## O Mundo em Blocos



- Existe um editor de mapas gratuito para Windows, o Mappy.
- O editor importa arquivos BMP e PNG para gerar mapas.
- Além de trabalhar com diversos tamanhos de blocos, o editor possui suporte a mapas isométricos (ex: Knightlore).

## O Mundo em Blocos



## O Mundo em Blocos



- A primeira opção é usar uma biblioteca de playback do Mappy, disponível para Allegro.
- Além desta opção, também podemos:
  - Exportar o mapa em formato texto.
  - Exportar o mapa como declaração de matriz em C/JAVA.
  - Exportar o mapa em um formato binário para ser lido dentro de uma matriz C/JAVA.
- Obs: a Game API do MIDP2.0 suporta explicitamente os tilemaps.

PUCPR - CCET

29

## O Mundo em Blocos



- Considerações:
  - Alguns jogos não necessitam obrigatoriamente de tilemaps, como o caso de jogos de trivia, jogo da velha, jogos de aventura, etc.
  - Nestes casos, imagens integrais são utilizadas e a lógica do jogo ou um descritor determina o posicionamento dos elementos em cada tela.

PUCPR - CCET

30

## Animação e Sprites



- Considera-se um *sprite* todo elemento gráfico que é desenhado contra um cenário de fundo no jogo.
- Possui as seguintes características:
  - Animação (opcional).
  - Possui uma área transparente, referente a área sem desenho útil.
- Os testes de colisão entre elementos e cenário são feitos sobre estes elementos.

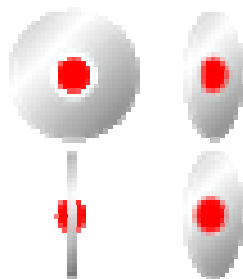
PUCPR - CCET

31

## Animação e Sprites



- A animação é atingida principalmente através da troca temporizada de quadros animados:



PUCPR - CCET

32

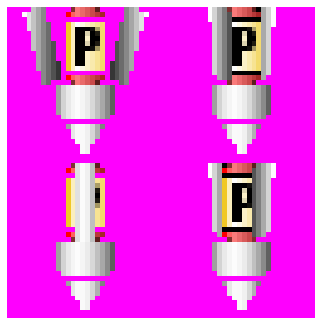


## Animação e Sprites

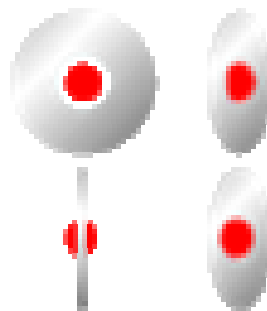


- A transparência dos pontos não desenhados pode ser atingida de duas maneiras:
  1. Através de colorkey: uma cor é escolhida para exclusão (normalmente rosa em *true color* ou cor 0 em sistema de palette).
  2. Através de canal de alpha em imagens 32 bits.

## Animação e Sprites



Colorkey



Canal de alpha

## Animação e Sprites



- Allegro suporta tanto canal de alpha como colorkey.
- J2ME suporta na maioria dos aparelhos canal de alpha simples – ou pontos opacos, ou pontos completamente ignorados.

## Animação e Sprites



- Canal de alpha tem a vantagem de permitir transparência real, inclusive pontos translúcidos.
- Além disso, é possível fazer operações de *anti-aliasing* para suavizar bordas de imagens com precisão.
- Desvantagem: modo de 32 bits não é suportado em TODOS os displays para sistemas desktop.

## Animação e Sprites



- Colorkey tem como vantagem a maior velocidade, requerendo apenas cópia de pontos.
- Como desvantagem, não é possível utilizar *anti-aliasing* para melhorar a qualidade de imagem.

## Animação e Sprites



- Normalmente, como nos blocos de cenário, associa-se aos sprites um tamanho fixo.
- Esta escolha normalmente é feita para padronizar e acelerar o mecanismo de colisão através de máscaras de bits.
- Para criar elementos maiores, utilizam-se vários sprites e uma espécie de tilemap para representar cada quadro do elemento.

## Formatos de Arquivo



- A maioria dos editores do mercado trabalha com um grande conjunto de formatos de imagem.
- Destes formatos, nos interessam os seguintes:
  - BMP (Allegro)
  - TGA (Allegro)
  - PNG (J2ME)

## Formatos de Arquivo



- No caso do J2ME, é importante que o editor suporte a edição de imagens contra um fundo transparente.
- Exemplos:
  - Paint Shop Pro
  - Gimp
  - Photoshop

## Formatos de Arquivo



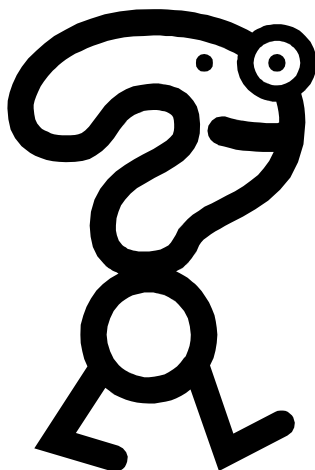
- No caso do Allegro, a escolha do formato é praticamente uma escolha pessoal.
- A única ressalva é quanto a modos 32 bits.
- Neste caso, o adequado são imagens TGA.
- Motivo: por mais que alguns editores gerem arquivos BMP 32 bits e a Allegro suporte, estes são fora do padrão oficial e a vasta maioria dos editores sequer os gera.

## Próximo Seminário



- Projeto III: Áudio
- Exemplo de projeto: jogo da velha

## Perguntas?



PUCPR - CCET

43