



Programação Gráfica – Parte 1

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

Paulo V. W. Radtke

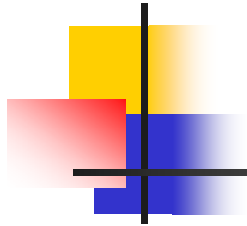
pvwradtke@gmail.com

<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.



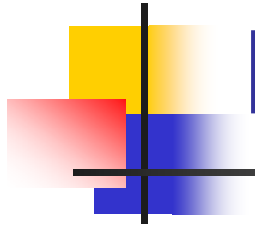
Conteúdo

- Introdução ao J2ME
- Instalação Básica e o WTK
- Criando um Projeto no WTK
- O Midlet
- Exemplo: "Hello J2ME"



Livro *Free* de J2ME e Jogos

- **J2ME & Gaming**
- Por Jason Lam
- <http://www.jasonlam604.com/books.php>
- Apesar de ser em inglês, é *free*.



Livros Comerciais

- **Wireless Game Development in Java with MIDP2.0**
- Autor: Ralph Barbagallo
- Wordware Publishing
- Disponibilidade: tem que importar.
- Aonde: Amazon (nota: livro NÃO paga imposto)



Livros Comerciais

- **J2ME Game Programming**
- Autor: Martin J. Wells
- Course Technology PTR
- Disponibilidade: importado, mas tem na biblioteca.
- Problema: trabalha com MIDP1.0, enquanto vamos trabalhar com MIDP2.0.



Introdução ao J2ME

- Versão do Java feita para rodar em dispositivos limitados.
- Quem usa J2ME:
 - Celulares
 - Palmtops
 - *Players* BluRay (Sony, vídeo de alta definição)



Introdução ao J2ME

- Terminologia:
 - MIDP: Mobile Information Device Profile
 - MIDP 1.0 (suporte total)
 - MIDP 2.0 (mais recente, suporte não é 100%)
 - Midlet: MIDP Application
 - CLDC: Connected Limited Device Configuration
 - JAD: Java Application Descriptor



Introdução ao J2ME

- Terminologia:
 - JAR: Java Archive (o programa distribuído)
 - OTA: over the air (distribuição e acesso)
- O JAR é separado do JAD por um motivo, qual?
- Para o aparelho determinar apenas com um arquivo pequeno se o JAR serve para ele!!



Introdução ao J2ME

- Limitações:
 - Cada aparelho tem um tamanho máximo para o JAR.
 - Menos memória que um sistema desktop.
 - Não existe suporte a números reais (ponto flutuante).



Introdução ao J2ME

- Desempenho:
 - Use StringBuffer ao invés de String.
 - Use atributos públicos em classes e acesse-os diretamente.
 - Variáveis locais são mais rápidas que atributos de classe.
 - Reutilize objetos.
 - Use métodos da biblioteca Java (rapidez).



Instalação Básica e o WTK

- Para desenvolver aplicações J2ME precisamos de duas coisas:
 - **J2SE SDK** ou J2EE SDK.
 - Wireless Toolkit (WTK).
- Ambos podem ser obtidos em java.sun.com.
- Para o WTK de aparelhos específicos, visite o site de seu fabricante.



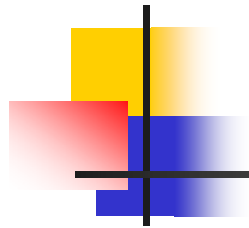
Instalação Básica e o WTK

- Nokia: <http://www.forum.nokia.com/>
- Samsung:
http://developer.samsungmobile.com/eng/front_zone/bbs/bbs_main.jsp?p_menu_id=1500
- SonyEricson:
http://developer.sonyericsson.com/site/global/home/p_home.jsp
- Motorola: <http://www.motocoder.com>



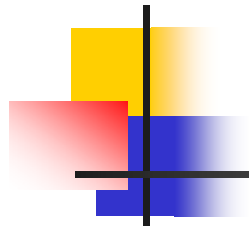
Instalação Básica e o WTK

- O WTK é o programa responsável pela compilação e execução dos Midlets.
- Ele NÃO faz a edição dos programas.
- Para isto, utilizaremos uma de duas ferramentas:
 - Windows Notepad
 - JCreator



Instalação Básica e o WTK

- Os projetos criados ficam dentro do diretório
<instalação>\WTK22\apps.
- Instalação é o diretório aonde se encontra instalado o WTK.
- O "22" no nome refere-se à versão, no caso, a 2.2.
- Versões diferentes terão outro nome.

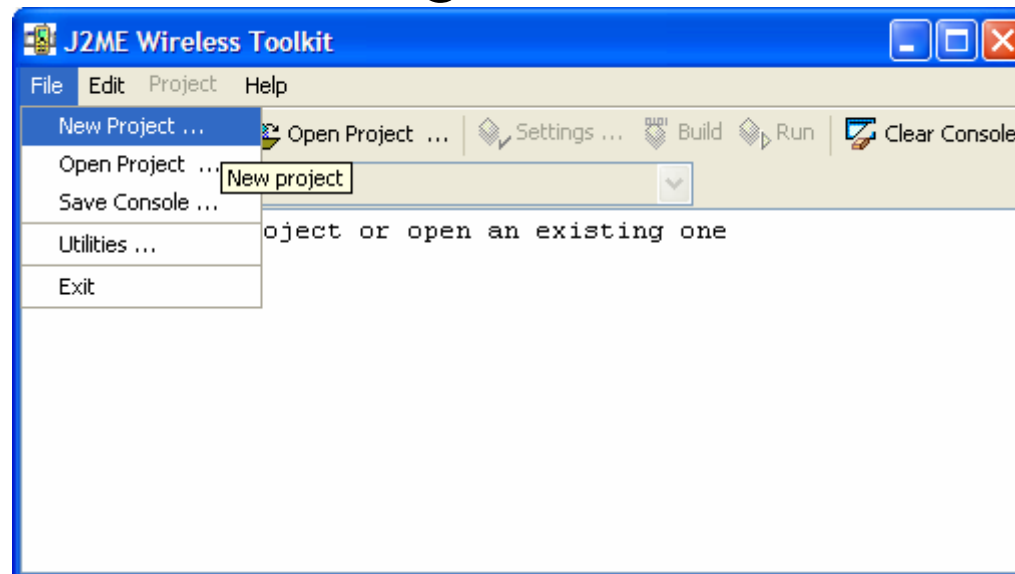


Instalação Básica e o WTK

- Diretórios importantes em um projeto do WTK:
 - **src**: aonde ficam os fontes .java, que podem ser organizados em pacotes.
 - **classes**: aonde são gerados os binários.
 - **bin**: aonde são gerados os arquivos JAR e JAD.
 - **res**: recursos do projeto, como imagens.
 - **lib**: diretório para bibliotecas extras.

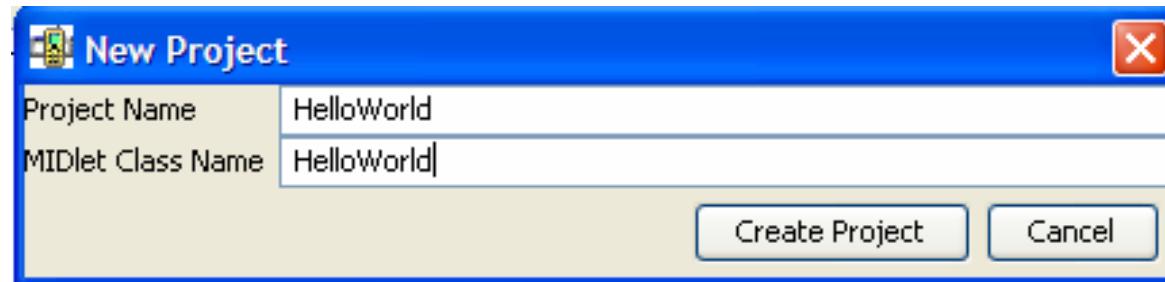
Criando um Projeto no WTK

- O programa principal é o KToolbar.
- Para criar um projeto, selecionamos a opção **New Project** em **File**.



Criando um Projeto no WTK

- Uma vez selecionada a opção, o Ktollbar irá pedir o nome do projeto e o nome da Classe.
- Use para ambos **HelloWorld**.

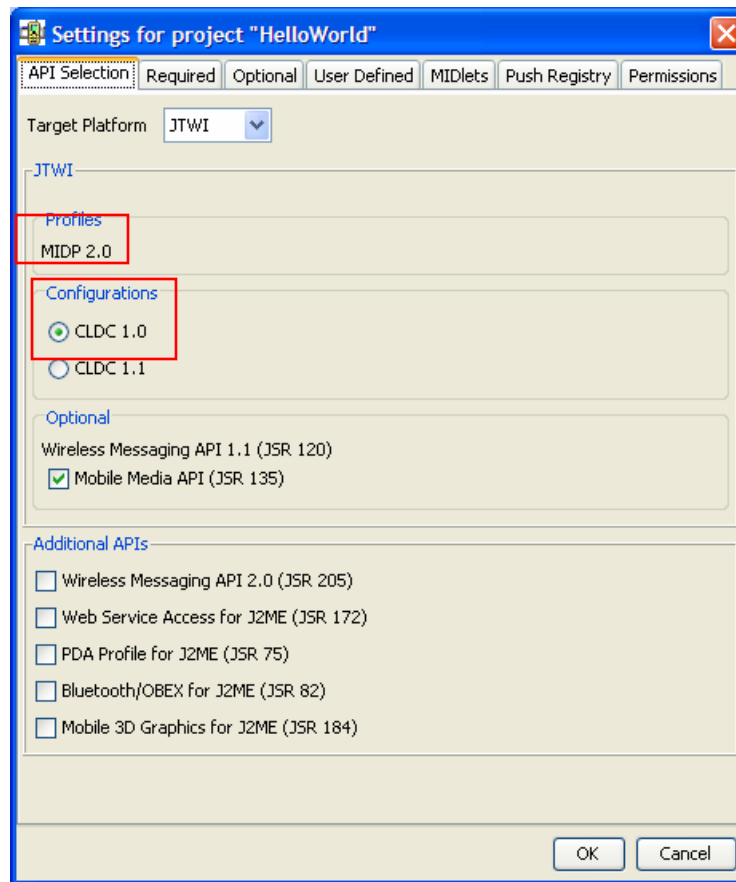




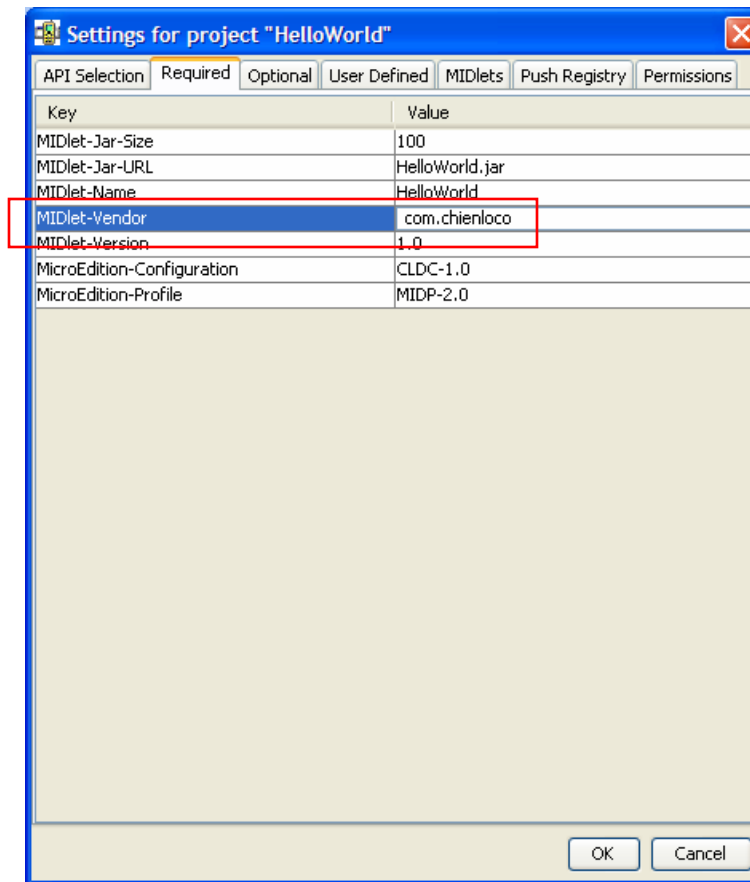
Criando um Projeto no WTK

- Após isto serão perguntadas informações do projeto.
- Na janela de opções, três abas são importantes.
- A **API Selection** define o que a sua aplicação terá disponível (APIs).
- **Required** define várias informações extra, como o nome do desenvolvedor.
- **User Defined** permite definir valores que podem ser lidos pelo Midlet.

Criando um Projeto no WTK

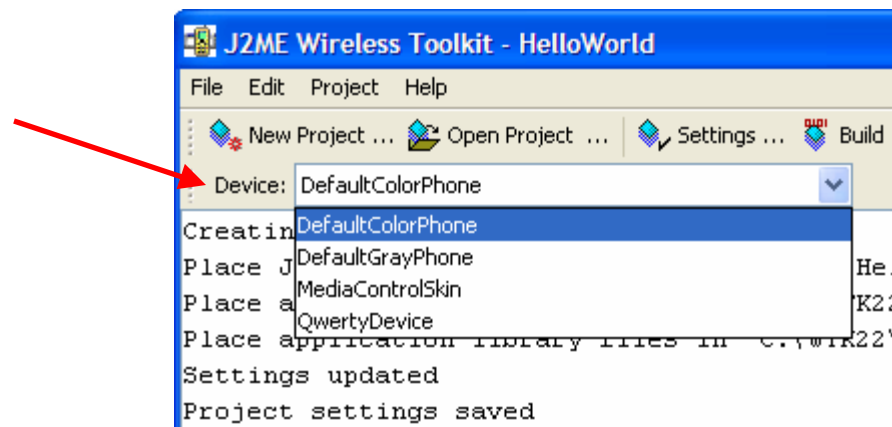


Criando um Projeto no WTK



Criando um Projeto no WTK

- Além destas opções, o WTK permite escolher um aparelho no simulador.
- Caso você tenha pego o WTK do fabricante do seu aparelho, ele estará disponível aqui:



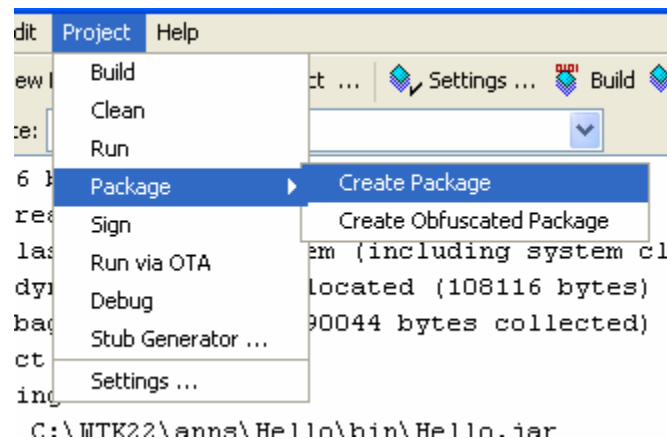


Criando um Projeto no WTK

- Agora, basta criar arquivos .java no diretório **src**, colocar recursos em **res** e gerar a aplicação.
- Mas atenção, o projeto é gravado no diretório de instalação do WTK, logo ...
- **É de sua responsabilidade fazer cópias de segurança dos arquivos. Não há garantias que os mesmos ficarão no disco ou que terceiros não irão copiá-los.**

Gerando Pacotes no KToolbar

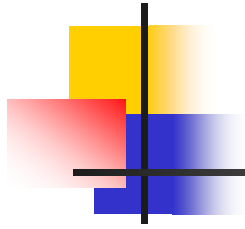
- Para gerar um pacote que possa ser utilizado em um celular, selecionamos "Project, Package, Create Package."
- Isto gera os arquivos JAR e JAD.





O Midlet

- Para trabalharmos com J2ME utilizaremos classes e pacotes:
 - **javax.microedition.midlet.MIDlet**
 - **javax.microedition.lcdui.game.GameCanvas**
 - **javax.microedition.lcdui.Display**
 - Pacote **javax.microedition.lcdui**
 - **javax.microedition.lcdui.game.Sprite**
 - **javax.microedition.lcdui.game.Layer**
 - **javax.microedition.lcdui.game.LayerManager**
 - **javax.microedition.lcdui.game.TiledLayer**



Trabalho Para Casa

- Sabia que o WTK tem no diretório docs/api/midp a especificação de TODAS as classes MIDP?
- Tarefa para seu conhecimento: ler a documentação das classes do slide anterior.



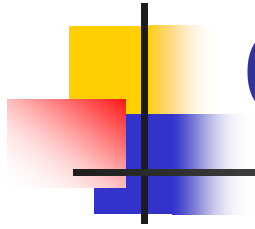
O Midlet

- O Midlet é a classe base de TODA aplicação J2ME.
- O TODA está enfatizado pelo simples fato de que não existe aplicação J2ME sem uma classe Midlet.
- Logo, escrever um jogo para celular com Java exige que conheçamos o Midlet.



O Midlet

- Ao contrário dos programas escritos até agora em Programação I (tudo estático), o Midlet segue uma série de convenções.
- Fatos “*assustadores*” para iniciantes de Java:
 - O Midlet não tem método **main** .
 - O fluxo do programa é controlado por eventos, não necessariamente pela sequência do programa.
 - O sistema pode encerrar a aplicação a qualquer momento e devemos responder a isso.

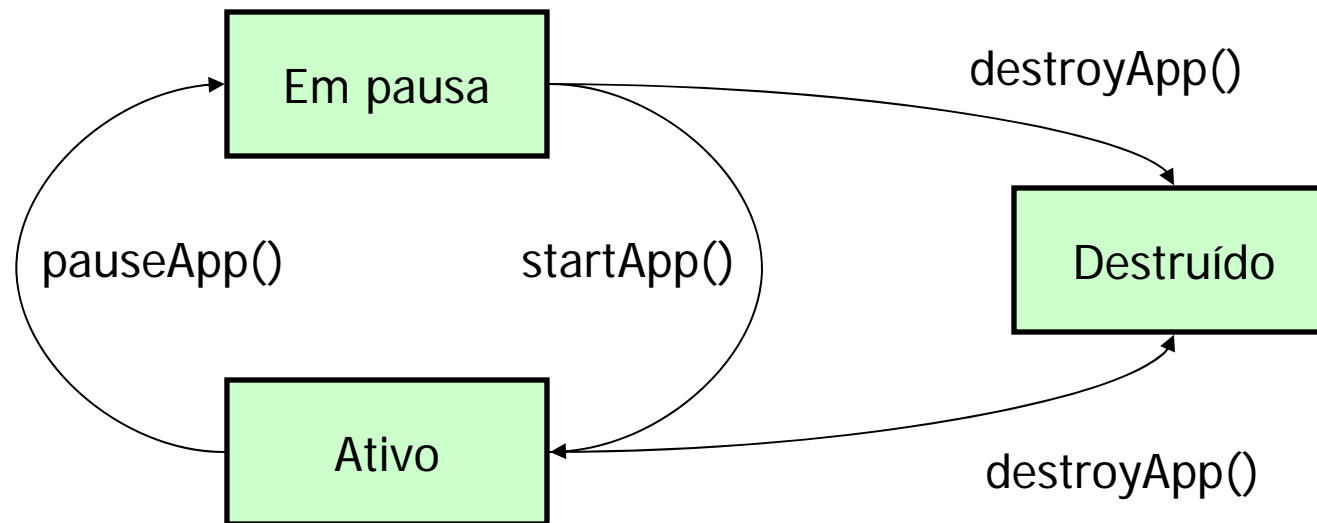


O Midlet

- Porém, esta será apenas uma impressão inicial.
- A programação de um Midlet é fácil uma vez que entendamos os seus conceitos.
- O fundamental é o ciclo de vida do Midlet.

O Midlet

■ Ciclo de vida do Midlet:





O Midlet - Exemplo

- Para o restante da aula, será utilizado a aplicação **Hello World**:
 1. Crie o projeto como descrito anteriormente.
 2. Faça o download do exemplo da aula de hoje, *j2me-exemplo01.zip*.
 3. Extraia o arquivo na pasta src do projeto **HelloWorld** do WTK.
 4. Verifique se foram criados corretamente os arquivos *HelloWorld.java* e *HelloCanvas.java*.



O Midlet

- O Midlet tem AO MENOS quatro métodos distintos:
 - Construtor (recebe o mesmo nome que a classe e não recebe/retorna valores).
 - startApp: chamado quando a aplicação sai do estado de pausa.
 - pauseApp: chamado quando a aplicação vai para o estado de pausa.
 - destroyApp(boolean): chamado quando encerra a aplicação.



O Midlet

- O usuário pode definir outros métodos para o Midlet, implementar *interfaces*, etc. Porém, isto é **OPCIONAL** e depende da aplicação.



O Midlet

- Passos de execução:

1. A JVM carrega o Midlet na memória.
2. A classe principal do Midlet, descrita no JAD é criada e seu construtor é chamado.
3. O Midlet neste ponto está pausado.
4. Em seguida a JVM chama o método `startApp()` do Midlet.
5. O programa está rodando e pode ser pausado/iniciado de acordo com o ambiente.
6. Ao encerrar a aplicação o método `destroyApp()` é chamado.



O Midlet – HelloWorld.java

```
import javax.microedition.midlet.*;  
import javax.microedition.lcdui.*;
```

```
public class HelloWorld extends MIDlet {  
    private Display display;  
    private HelloCanvas canvas;
```

```
    public HelloWorld()  
    {  
        display = Display.getDisplay(this);  
        canvas = new HelloCanvas();  
    }
```



O Midlet – HelloWorld.java

```
public void startApp()
{

    display.setCurrent(canvas);
}
public void pauseApp()
{
}
public void destroyApp(boolean unconditional)
{
}
}
```

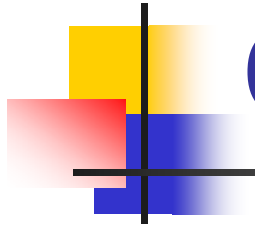


O Midlet – HelloCanvas.java

```
import javax.microedition.lcdui.*;
import javax.microedition.lcdui.game.*;

public class HelloCanvas extends GameCanvas
{
    private int maxX; // Largura da tela
    private int maxY; // Altura da tela

    public HelloCanvas()
    {
        super(true);
        maxX = getWidth();
        maxY = getHeight();
    }
}
```



O Midlet – HelloCanvas.java

```
public void paint(Graphics g)
{
    g.setColor(255, 255, 255);
    g.fillRect(0, 0, maxX, maxY);
    g.setColor(0, 0, 0);
    g.drawString("Hello World! APlicação J2ME em
ação.", maxX/2,maxY/2,Graphics.HCENTER |
Graphics.BASELINE);
}
}
```



O Midlet

- Resumo da operação:
 - Ao ser construído, o Midlet HelloWorld obtém o *display* (tela) do celular e armazena-a em um atributo.
 - Em seguida, ele cria o canvas (tela lógica) da aplicação HelloCanvas, também armazenando-o em atributo da classe.
 - A escolha é feita para que isto seja feito uma vez e fique disponível durante toda a execução.



O Midlet

- Resumo da operação (continuação):
 - No próximo passo, o método startApp é chamado.
 - Neste ponto a aplicação deve ganhar controle dos recursos do sistema, ou recuperá-los.
 - Aqui devemos associar à tela do celular o nosso canvas (HelloCanvas), armazenado no atributo.
 - Obs: poderíamos recuperar aqui o canvas do display, porém usamos o atributo por questões de desempenho.



O Midlet

- Resumo da operação (continuação):
 - Ao fecharmos a aplicação (pelo celular), o método `destroyApp` é invocado.



O Midlet

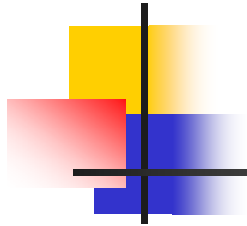
- O GameCanvas é mais simples.
- O construtor simplesmente recupera a largura e altura do display.
- O método paint é chamado toda vez que for preciso redesenhar a tela.
- Este método simplesmente escreve o texto na tela.

Exemplo: "Hello J2ME"



Exemplo: "Hello J2ME"





Próxima Aula

- O GameCanvas em detalhes.
- Desenhando um sprite na tela.
- Movendo o sprite com eventos do teclado.