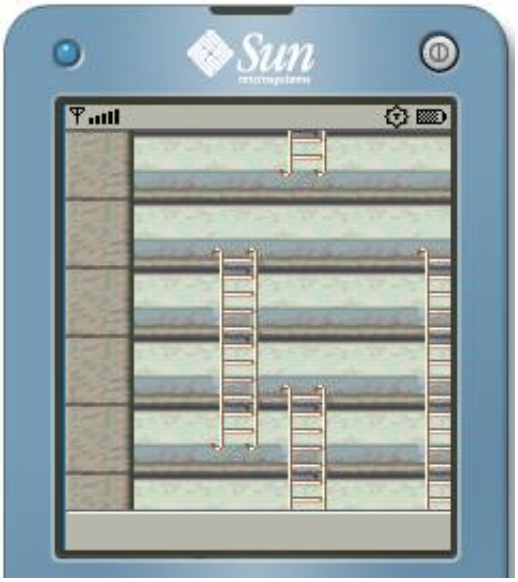
Tiled Layer

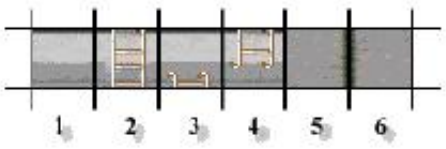
Back | Tutorial Home | Next

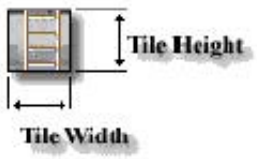
TiledLayer is the last but certainly not the least significant class that is available in the game package. Games with large backgrounds / virtual areas the TiledLayer is the perfect candidate in dealing with this. It allows you to simply to define all the unique areas of the background and reuse them as many times as need to produce a complete image. Take for example the following image:



TiledLayer Example Screen Shot

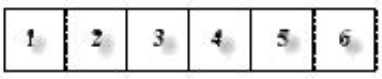
If you study the image you will notice there several areas that are similar and in fact you can break the image up into 6 distinct areas.

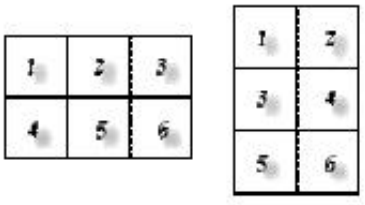




Example Tile Sheet

You will notice all tiles have the same dimensions 32 x 32 pixels as required by the TiledLayer class. Each tile is given an index numeric value starting from 1. The numbering occurs from left to right and top to bottom. In other words numbers are assigned row-by-row bases. You can lay the tiles in any configuration and assuming you kept the same tile order the assigned index values will not change.





TiledLayer Assigned Index

In the above example these tiles are considered static tiles because of the one to one fixed relationship from tile to index value. If the index contains the value zero this indicates that cell is empty, meaning any cell with the value zero nothing will drawn in that area the cell occupies.

TiledLayer Constructor

To created a TiledLayer simply instantiate the constructor

```
TiledLayer(int columns, int rows,
Image image, int tileWidth, int tileHeight)
```

The first 2 values refer to the number of columns and rows that make up the entire final image. The third parameter is the image of tiles / tile sheet used to map against the index values. The last 2 parameters is the actual width and height for one specific tile..

TiledLayer Manipulation

To map a tile image to a specific cell you need to use the setCell(int col, int row, int tileIndex) method. Column and row is the location where you want the tile image to be rendered at and the last parameter maps the tile to be render at that cell location.

To replace the entire tile set / tile sheet after you have called the constructor simply use the setStaticTileSet(Image image, int tileWidth, tileHeight) method. If the new tile set is the same or contains more tiles then the original tile set the animated tiles and cell contents will remain the same. However, if the new set contains less tiles then the previous tile set then the mapping will be reset; all indexes will contain the value zero and all animated tiles will be deleted.

To fill specified region of cells with a specific tile you can use the fillCells(int col, int row, int numCols, int numRows, int tileIndex) method. The first to values indicate the column and row for the top-left cell in the specified region. Next 2 define the number of columns and rows for the region. The last parameter is the index in the region.

Display TiledLayer

Like the other game objects all you need to do is call the paint method directly or use the LayerManager.

Retrieve Current TiledLayer Settings

There are several self-explanatory methods available to obtain information on the current TiledLayer in use:

- getCell(int col, int row)
- getCellHeight()
- getCellWidth()

- getColumns()
- getRows()

Animated Cells

There is one more feature TiledLayer has to offer which is animated tiles. Animated tiles are indicated by negative index values. Each animated cell is dynamically associated to a static tile. This allows us to easily animate a group of cells simply by changing the associated static cell. This is great for background animation like a cheering crowd, moving clouds and/or rippling water. Aside from the previously mentioned methods there are three additional methods specific to animated cells:

- createAnimatedTile(int staticTileIndex) – creates a new animated tile a the specified index and returns the next consecutive negative index for the animated tile. By default it will contain a static tile (positive number) or the value zero.
- getAnimatedTile(int animatedTileIndex) – retrieves the tile map to the animated tile index.
- setAnimatedtile(int animatedTileIndex, int staticTileIndex) – links an animated tile to a static tile

Non-Animated TiledLayer Example

GameCanvas Source Code

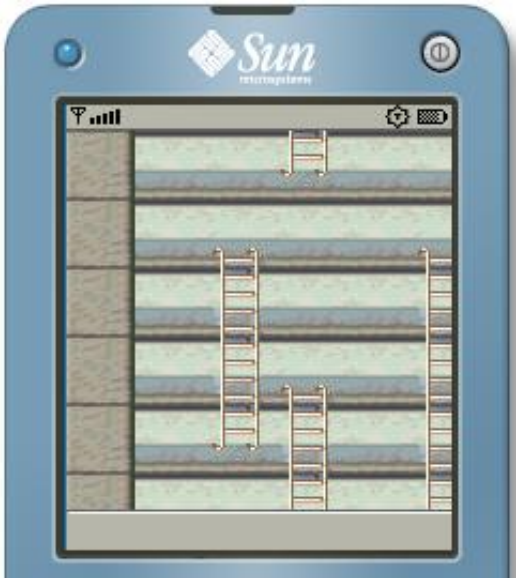
```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class ExampleTiledLayer extends MIDlet {
  private Display display;

  public void startApp() {
    try {
      display = Display.getDisplay(this);
      ExampleGameCanvas gameCanvas = new ExampleGameCanvas();
      gameCanvas.start();
      display.setCurrent(gameCanvas);
    } catch (Exception ex) {
      System.out.println(ex);
    }
  }

  public Display getDisplay() {
    return display;
  }

  public void pauseApp() {
  }

  public void destroyApp(boolean unconditional) {
    exit();
  }

  public void exit() {
    System.gc();
    destroyApp(false);
    notifyDestroyed();
  }
}
```

Main Midlet Source Code

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

public class ExampleGameCanvas extends GameCanvas implements Runnable {
  private boolean isPlay;   // Game Loop runs when isPlay is true
  private long delay;       // To give thread consistency
  private int width;        // To hold screen width
  private int height;       // To hold screen height


  // Layer Manager
  private LayerManager layerManager;

  // TiledLayer
  private TiledLayer tiledBackground;

  // Constructor and initialization
  public ExampleGameCanvas() throws Exception {
    super(true);
    width = getWidth();
    height = getHeight();
    delay = 20;

    //tiledBackground = initBackground();
    layerManager = new LayerManager();
    layerManager.append(new Image("Tiles.jpeg"));
    //layerManager.append(tiledBackground);
  }

  // Automatically start thread for game loop
  public void start() {
    isPlay = true;
    Thread t = new Thread(this);
    t.start();
  }

  public void stop() { isPlay = false; }

  // Main Game Loop
  public void run() {
    Graphics g = getGraphics();

    while (isPlay == true) {
      input();
      drawScreen(g);
      try {
        Thread.sleep(delay);
      } catch (InterruptedException ie) {
      }
    }
  }

  // Method to Handle User Inputs
  private void input() {
     // no inputs
```

```
  }

  // Method to Display Graphics
  private void drawScreen(Graphics g) {
    g.setColor(0xffffff);
    g.fillRect(0, 0, getWidth(), getHeight());
    g.setColor(0x0000ff);

    layerManager.paint(g,0,0);
    flushGraphics();
  }

  private TiledLayer initBackground() throws Exception {
    Image tileImages = Image.createImage("/tiles.png");
    TiledLayer tiledLayer = new TiledLayer(10,10,tileImages,32,32);

    int[] map = {
       5,  1,  1,  4,  1,  1,  1,  1,  1,  6,
       5,  1,  3,  1,  1,  3,  1,  1,  1,  6,
       5,  1,  2,  1,  1,  2,  1,  1,  1,  6,
       5,  1,  2,  3,  1,  2,  1,  1,  1,  6,
       5,  1,  4,  2,  1,  2,  1,  1,  1,  6,
       5,  1,  1,  4,  1,  2,  1,  1,  1,  6,
       5,  1,  1,  1,  1,  4,  1,  1,  1,  6,
       5,  1,  1,  1,  1,  1,  1,  1,  1,  6,
       5,  1,  1,  1,  1,  1,  1,  1,  1,  6,
       5,  1,  1,  1,  1,  1,  1,  1,  1,  6
    };
    for (int i=0; i < map.length; i++) {
      int column = i % 10;
      int row = (i - column) / 10;
      tiledLayer.setCell(column,row,map[i]);
    }

    return tiledLayer;
  }
}
```

Screen Shot Output



Animated TiledLayerExample

The following example illustrates a simple example of to animated tile, though logically the tile that is being animated makes no sense but that is not the point. The point is to clearly show an animated example.

GameCanvas Source Code

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

public class ExampleGameCanvas extends GameCanvas implements Runnable {
  private boolean isPlay;   // Game Loop runs when isPlay is true
  private long delay;       // To give thread consistency
  private int currentX, currentY;  // To hold current position of the 'X'
  private int width;        // To hold screen width
  private int height;       // To hold screen height


  // Layer Manager
  private LayerManager layerManager;

  // TiledLayer
  private TiledLayer tiledBackground;

  // Flag to indicate tile switch
  private boolean switchTile;

  // To hold the animated tile index
  private int animatedIdx;

  // Constructor and initialization
  public ExampleGameCanvas() throws Exception {
    super(true);
    width = getWidth();
    height = getHeight();

    currentX = width / 2;
    currentY = height / 2;
    delay = 20;

    tiledBackground = initBackground();
    layerManager = new LayerManager();
    layerManager.append(tiledBackground);
  }

  // Automatically start thread for game loop
  public void start() {
    isPlay = true;
    Thread t = new Thread(this);
    t.start();
  }

  public void stop() { isPlay = false; }

  // Main Game Loop
  public void run() {
    Graphics g = getGraphics();
```

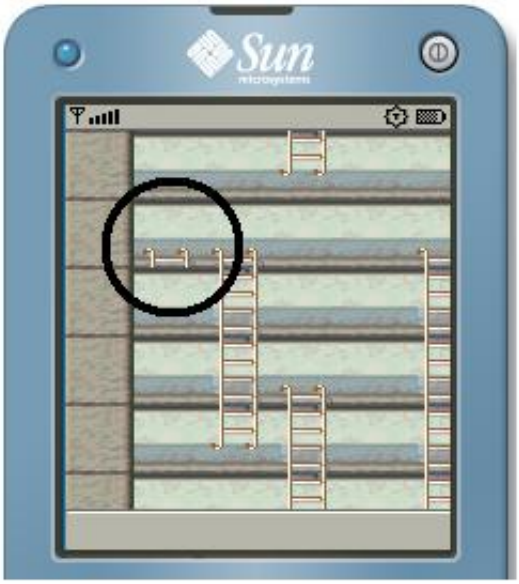
```
    while (isPlay == true) {
      input();
      drawScreen(g);
      try {
        Thread.sleep(delay);
      } catch (InterruptedException ie) {
      }
    }
  }

  // Method to Handle User Inputs
  private void input() {
     // no inputs
  }

  // Method to Display Graphics
  private void drawScreen(Graphics g) {
    g.setColor(0xffffff);
    g.fillRect(0, 0, getWidth(), getHeight());
    g.setColor(0x0000ff);

    // Determine which tile to show
    if (switchTile) {
      tiledBackground.setAnimatedTile(animatedIdx,3);
    } else {
      tiledBackground.setAnimatedTile(animatedIdx,4);
    }

    // Set tile file to opposite boolean value
    switchTile = !switchTile;

    layerManager.paint(g,0,0);
    flushGraphics();
  }

  private TiledLayer initBackground() throws Exception {
    Image tileImages = Image.createImage("/tiles.png");
    TiledLayer tiledLayer = new TiledLayer(10,10,tileImages,32,32);

    int[] map = {
       5,  1,  1,  4,  1,  1,  1,  1,  1,  6,
       5,  1,  3,  1,  1,  3,  1,  1,  1,  6,
       5,  1,  2,  1,  1,  2,  1,  1,  1,  6,
       5,  1,  2,  3,  1,  2,  1,  1,  1,  6,
       5,  1,  4,  2,  1,  2,  1,  1,  1,  6,
       5,  1,  1,  4,  1,  2,  1,  1,  1,  6,
       5,  1,  1,  1,  1,  4,  1,  1,  1,  6,
       5,  1,  1,  1,  1,  1,  1,  1,  1,  6,
       5,  1,  1,  1,  1,  1,  1,  1,  1,  6,
       5,  1,  1,  1,  1,  1,  1,  1,  1,  6
    };
    for (int i=0; i < map.length; i++) {
      int column = i % 10;
      int row = (i - column) / 10;
      tiledLayer.setCell(column,row,map[i]);
    }
```

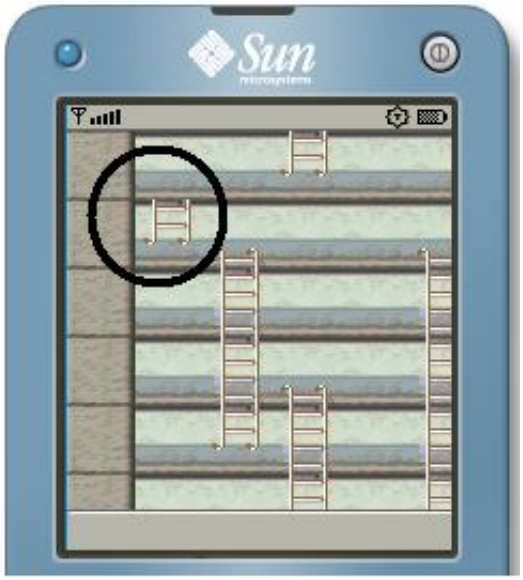
```
    // Created animate tile and hold animated tile index
    animatedIdx = tiledLayer.createAnimatedTile(5);

    // Set Cell with animated tile index
    tiledLayer.setCell(1,1,animatedIdx);

    return tiledLayer;
  }
}
```

Main Midlet Source Code

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

public class ExampleTiledLayerAnimated extends MIDlet {
  private Display display;

  public void startApp() {
    try {
      display = Display.getDisplay(this);
      ExampleGameCanvas gameCanvas = new ExampleGameCanvas();
      gameCanvas.start();
      display.setCurrent(gameCanvas);
    } catch (Exception ex) {
      System.out.println(ex);
    }
  }

  public Display getDisplay() {
    return display;
  }

  public void pauseApp() {
  }

  public void destroyApp(boolean unconditional) {
    exit();
  }

  public void exit() {
    System.gc();
    destroyApp(false);
    notifyDestroyed();
  }
}
```

Animated Frames





j2meSalsa goodies

Execute Sample online	Execute on your browsers left frame.
Download demo code for deploying directly to WTK	Download tiles example, Download animated tiles example

Back | Tutorial Home | Next

3 comments

3

Leave a message...

Best Community Share

Avatar Jitubaba10 • 3 years ago

gr8 tutorial...

1 Reply Share ›

Avatar xuan dung • 7 months ago

helf full

Reply Share ›

Avatar NishaTellis • a year ago

How to do tile animation???? like sprite we have nextframe()

Reply Share ›

Subscribe Add Disqus to your site

This page is a part of a frames based web site. If you have landed on this page from a search engine click here to view the complete page.