Live Wallpapers

Introduction

- Starting with Android 2.1 (API level 7) users can enjoy rich animated and interactive backgrounds on their home screens.
- The live wallpaper applications have access to most of the android platform, including the OpenGL, GPS, accelerometer, networking accessibility etc.

Develop Live Wallpaper

- Developing our new live wallpapers is very similar to developing a new service. Instead of extending `Service` we should extend `WallpaperService`.
- This class includes the definition of one abstract method only. The `onCreateEngine()` method. Its purpose is to create and return a `WallpaperService.Engine` object.
- This engine is responsible for drawing the wallpapers and for handling its life cycle.

User Interaction

- We can add support for user interaction by implementing various methods in our implementation for the `WallpaperService.Engine` class.
- Implementing the `onOffsetsChanged()` method will scroll the wallpaper along with the user swipes from one home screen to another.
- Implementing the `onTouchEvent()` method our wallpaper will react touch events.

User Interaction

- Applications can send commands to the live wallpaper in use. Sending a command is done by calling the `onCommand()` method.
- Currently, the standard home application is the only one that sends commands to the `onCommand()` method of the live wallpaper.

User Interaction

- When the user taps an empty space on the home screen workspace the `android.wallpaper.tap` command is passed over to the `onCommand()` method.
- When the user drops an icon or an app widget on the home screen workspace the `android.home.drop` command is passed over to the `onCommand()` method.

The `uses-sdk` Element

- When placing a live wallpapers for sale it is important that the AndroidManifest.xml file will include the following:

```
<uses-sdk android:minSdkVersion="7" />
```

The `uses-feature` Element

- When placing a live wallpapers for sale it is important that the AndroidManifest.xml file will include the following:

```
<uses-feature android:name=
    "android.software.live_wallpaper" />
```

Performance

- It is highly important to have the live wallpaper drawing the surface on its visible parts only. That will minimize the wallpaper's impact on the system performance and on its battery.

Code Sample

```
public class MyLiveWallpaper extends WallpaperService
{
    private int[] colors = {0xaabbaaff,0x040404ff,0x101010ff,0x858585ff};
    private final Handler handler = new Handler();

    @Override
    public Engine onCreateEngine()
    {
        return new MyWallpaperEngine();
    }

    class MyWallpaperEngine extends Engine
    {

        private final Paint paint = new Paint();
        private float x = -1;
        private float y = -1;

        private final Runnable runnable = new Runnable()
        {
            public void run()
            {
                drawFrame();
            }
        };
```

Code Sample

```
private boolean visible;

MyWallpaperEngine()
{

}

@Override
public void onCreate(SurfaceHolder surfaceHolder)
{
    super.onCreate(surfaceHolder);
    setTouchEventsEnabled(true);
}

@Override
public void onDestroy()
{
    super.onDestroy();
    handler.removeCallbacks(runnable);
}
```

Code Sample

```
@Override
public void onVisibilityChanged(boolean visible)
{
     this.visible = visible;
     if (visible)
     {
          drawFrame();
     }
     else
     {
          handler.removeCallbacks(runnable);
     }
}

@Override
public void onSurfaceChanged(SurfaceHolder holder, int format,
          int width, int height)
{
     super.onSurfaceChanged(holder, format, width, height);
     x = width / 2.0f;
     y = height / 2.0f;
     drawFrame();
}
```

Code Sample

```
@Override
public void onSurfaceCreated(SurfaceHolder holder)
{
    super.onSurfaceCreated(holder);
}

@Override
public void onSurfaceDestroyed(SurfaceHolder holder)
{
    super.onSurfaceDestroyed(holder);
    visible = false;
    handler.removeCallbacks(runnable);
}

@Override
public void onTouchEvent(MotionEvent event)
{
    if (event.getAction() == MotionEvent.ACTION_DOWN)
    {
        x = event.getX();
        y = event.getY();
    }
    else
    {
        x = -1;
        y = -1;
    }
    super.onTouchEvent(event);
}
```

Code Sample

```
void drawFrame()
{
    final SurfaceHolder holder = getSurfaceHolder();
    Canvas c = null;
    try
    {
        c = holder.lockCanvas();
        if (c != null)
        {
            drawTouchPoint(c);
        }
    }
    finally
    {
        if (c != null)
        {
            holder.unlockCanvasAndPost(c);
        }
    }
    handler.removeCallbacks(runnable);
    if (visible)
    {
        handler.postDelayed(runnable, 200);
    }
}
```

Code Sample

```
        void drawTouchPoint(Canvas c)
        {
            int color = colors[(int)(4*Math.random())];
            paint.setColor(color);
            if (x >= 0 && y >= 0)
            {
                c.drawCircle(x, y, 20, paint);
            }
        }

    }
}
```

Code Sample

```
<?xml version="1.0" encoding="UTF-8"?>

<wallpaper
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:settingsActivity="com.abelski.samples.LiveWallpaperActivity"/>
```

wlp.xml

Code Sample

```
package com.abelski.samples;

import android.app.Activity;
import android.os.Bundle;

public class LiveWallpaperActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Code Sample

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="possible settings..."
    />
</LinearLayout>
```

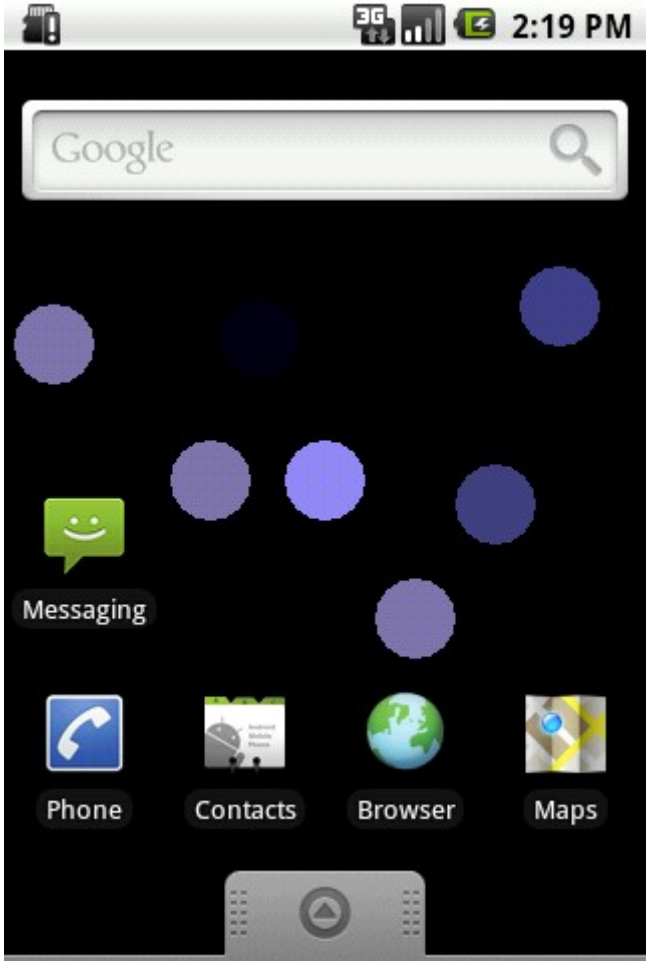
Code Sample

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.abelski.samples" android:versionCode="1"
    android:versionName="1.0">

    <application
        android:icon="@drawable/icon"
        android:label="@string/app_name">

        <activity android:name=".LiveWallpaperActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
```

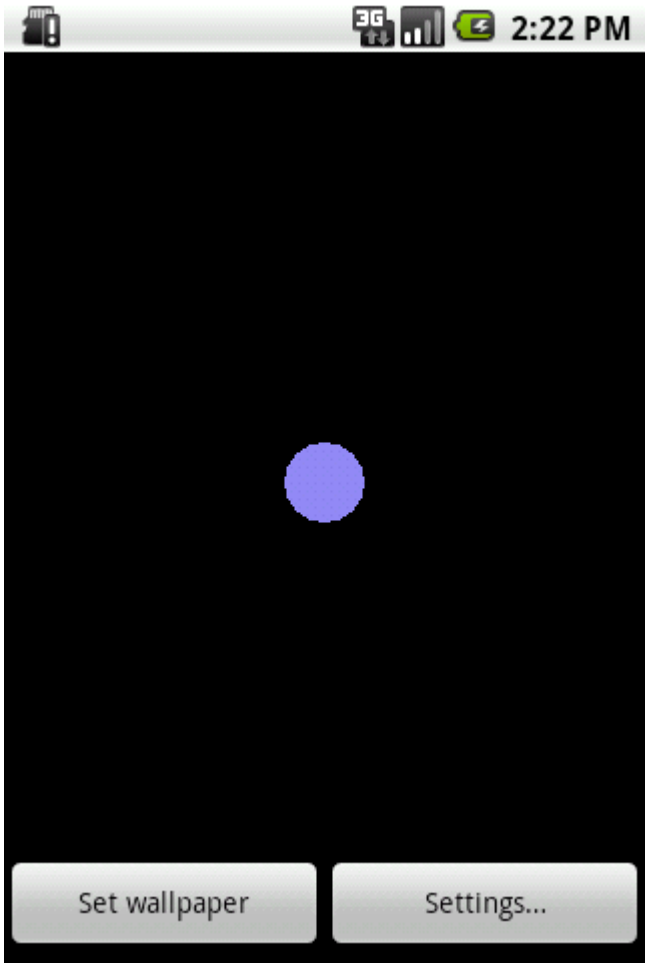
Code Sample

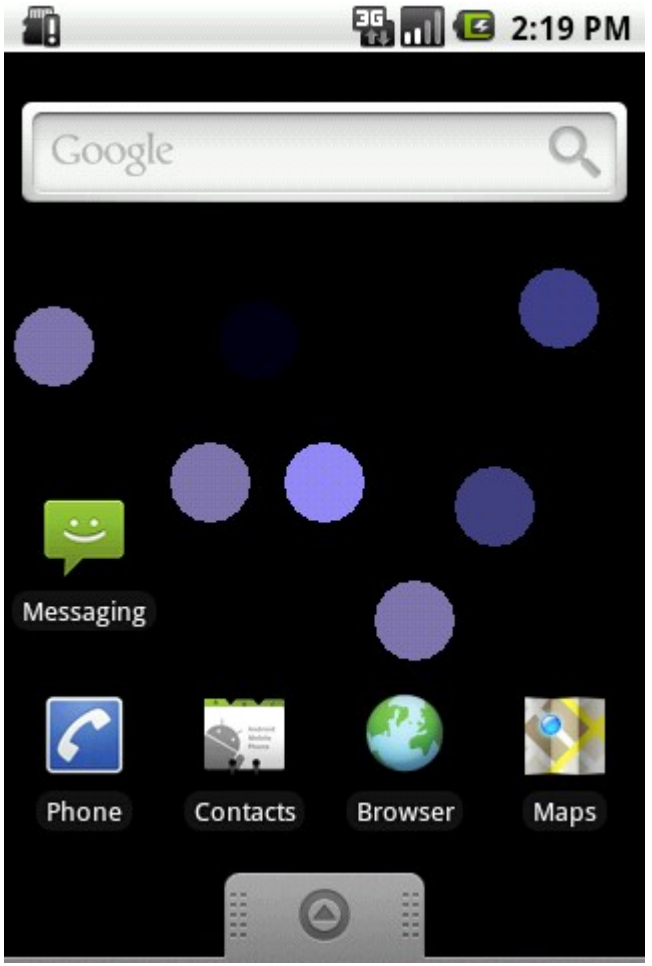
```
        <service android:label="@string/wallapapr_name"
            android:name=".MyLiveWallpaper"
            android:permission="android.permission.BIND_WALLPAPER">
            <intent-filter>
                <action
                android:name="android.service.wallpaper.WallpaperService" />
            </intent-filter>
            <meta-data android:name="android.service.wallpaper"
                android:resource="@xml/wlp" />
        </service>


    </application>

    <uses-sdk android:minSdkVersion="7" />

</manifest>
```

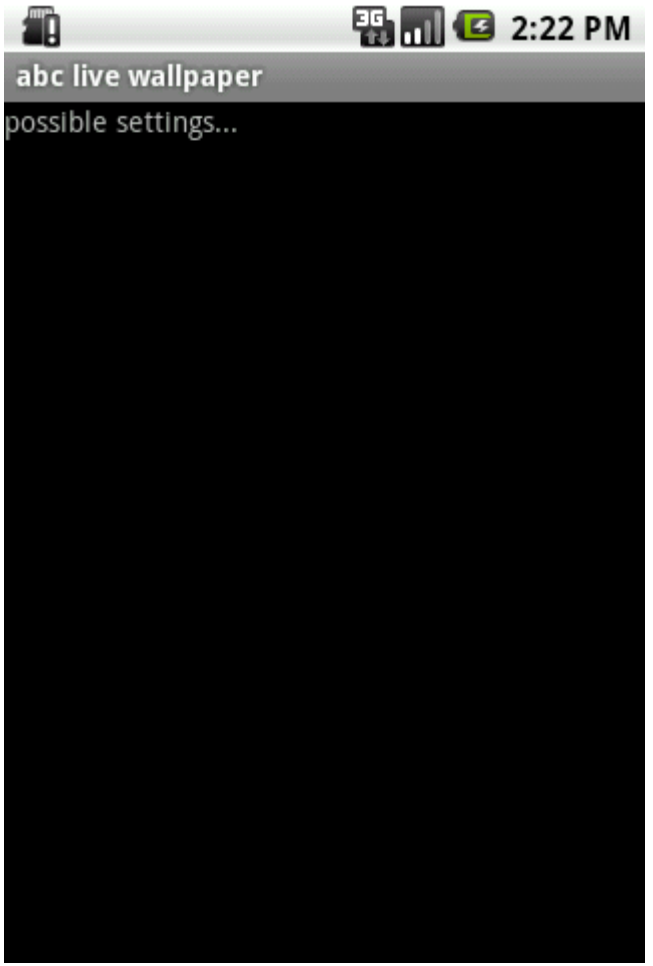
Code Sample



Code Sample







Live Wallpapers

Introduction

- Starting with Android 2.1 (API level 7) users can enjoy rich animated and interactive backgrounds on their home screens.
- The live wallpaper applications have access to most of the android platform, including the OpenGL, GPS, accelerometer, networking accessibility etc.

Develop Live Wallpaper

- Developing our new live wallpapers is very similar to developing a new service. Instead of extending `Service` we should extend `WallpaperService`.
- This class includes the definition of one abstract method only. The `onCreateEngine()` method. Its purpose is to create and return a `WallpaperService.Engine` object.
- This engine is responsible for drawing the wallpapers and for handling its life cycle.

User Interaction

- We can add support for user interaction by implementing various methods in our implementation for the `WallpaperService.Engine` class.
- Implementing the `onOffsetsChanged()` method will scroll the wallpaper along with the user swipes from one home screen to another.
- Implementing the `onTouchEvent()` method our wallpaper will react touch events.

User Interaction

- Applications can send commands to the live wallpaper in use. Sending a command is done by calling the `onCommand()` method.
- Currently, the standard home application is the only one that sends commands to the `onCommand()` method of the live wallpaper.

User Interaction

- When the user taps an empty space on the home screen workspace the `android.wallpaper.tap` command is passed over to the `onCommand()` method.
- When the user drops an icon or an app widget on the home screen workspace the `android.home.drop` command is passed over to the `onCommand()` method.

The `uses-sdk` Element

- When placing a live wallpapers for sale it is important that the AndroidManifest.xml file will include the following:

```
<uses-sdk android:minSdkVersion="7" />
```

The `uses-feature` Element

- When placing a live wallpapers for sale it is important that the AndroidManifest.xml file will include the following:

```
<uses-feature android:name=
    "android.software.live_wallpaper" />
```

Performance

- It is highly important to have the live wallpaper drawing the surface on its visible parts only. That will minimize the wallpaper's impact on the system performance and on its battery.

Code Sample

```
public class MyLiveWallpaper extends WallpaperService
{
    private int[] colors = {0xaabbaaff,0x040404ff,0x101010ff,0x858585ff};
    private final Handler handler = new Handler();

    @Override
    public Engine onCreateEngine()
    {
        return new MyWallpaperEngine();
    }

    class MyWallpaperEngine extends Engine
    {

        private final Paint paint = new Paint();
        private float x = -1;
        private float y = -1;

        private final Runnable runnable = new Runnable()
        {
            public void run()
            {
                drawFrame();
            }
        };
```

Code Sample

```
private boolean visible;

MyWallpaperEngine()
{

}

@Override
public void onCreate(SurfaceHolder surfaceHolder)
{
    super.onCreate(surfaceHolder);
    setTouchEventsEnabled(true);
}

@Override
public void onDestroy()
{
    super.onDestroy();
    handler.removeCallbacks(runnable);
}
```

Code Sample

```
@Override
public void onVisibilityChanged(boolean visible)
{
    this.visible = visible;
    if (visible)
    {
        drawFrame();
    }
    else
    {
        handler.removeCallbacks(runnable);
    }
}

@Override
public void onSurfaceChanged(SurfaceHolder holder, int format,
        int width, int height)
{
    super.onSurfaceChanged(holder, format, width, height);
    x = width / 2.0f;
    y = height / 2.0f;
    drawFrame();
}
```

Code Sample

```
@Override
public void onSurfaceCreated(SurfaceHolder holder)
{
    super.onSurfaceCreated(holder);
}

@Override
public void onSurfaceDestroyed(SurfaceHolder holder)
{
    super.onSurfaceDestroyed(holder);
    visible = false;
    handler.removeCallbacks(runnable);
}

@Override
public void onTouchEvent(MotionEvent event)
{
    if (event.getAction() == MotionEvent.ACTION_DOWN)
    {
        x = event.getX();
        y = event.getY();
    }
    else
    {
        x = -1;
        y = -1;
    }
    super.onTouchEvent(event);
}
```

Code Sample

```
void drawFrame()
{
    final SurfaceHolder holder = getSurfaceHolder();
    Canvas c = null;
    try
    {
        c = holder.lockCanvas();
        if (c != null)
        {
            drawTouchPoint(c);
        }
    }
    finally
    {
        if (c != null)
        {
            holder.unlockCanvasAndPost(c);
        }
    }
    handler.removeCallbacks(runnable);
    if (visible)
    {
        handler.postDelayed(runnable, 200);
    }
}
```

Code Sample

```
        void drawTouchPoint(Canvas c)
        {
            int color = colors[(int)(4*Math.random())];
            paint.setColor(color);
            if (x >= 0 && y >= 0)
            {
                c.drawCircle(x, y, 20, paint);
            }
        }

    }
}
```

Code Sample

```
<?xml version="1.0" encoding="UTF-8"?>

<wallpaper
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:settingsActivity="com.abelski.samples.LiveWallpaperActivity"/>
```

wlp.xml

Code Sample

```
package com.abelski.samples;

import android.app.Activity;
import android.os.Bundle;

public class LiveWallpaperActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Code Sample

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="possible settings..."
    />
</LinearLayout>
```

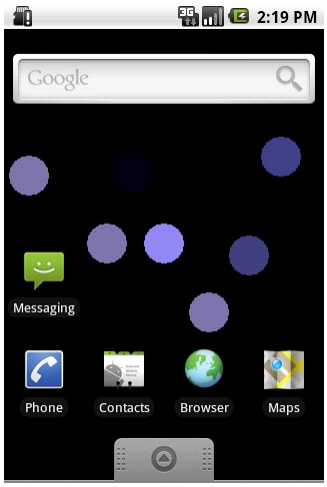
Code Sample

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.abelski.samples" android:versionCode="1"
    android:versionName="1.0">

    <application
        android:icon="@drawable/icon"
        android:label="@string/app_name">

        <activity android:name=".LiveWallpaperActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
```

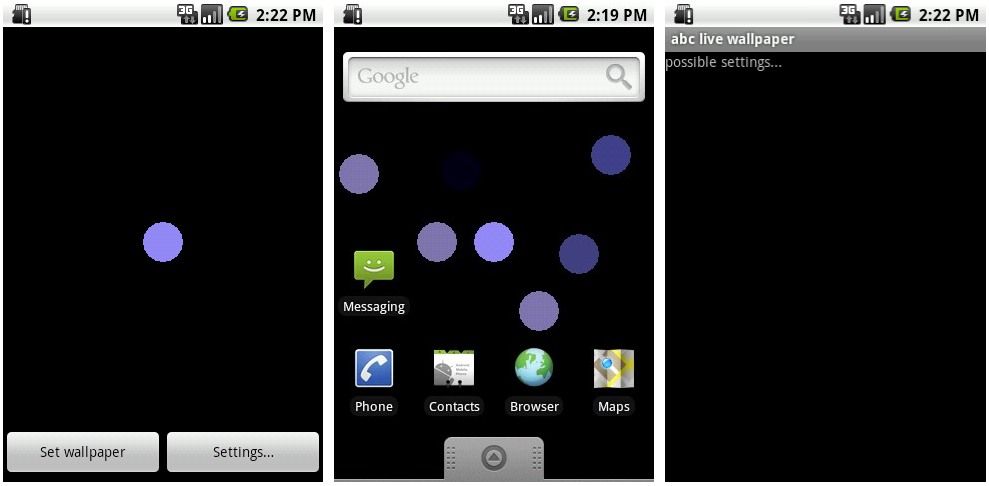
Code Sample

```
        <service android:label="@string/wallapapr_name"
            android:name=".MyLiveWallpaper"
            android:permission="android.permission.BIND_WALLPAPER">
            <intent-filter>
                <action
                android:name="android.service.wallpaper.WallpaperService" />
            </intent-filter>
            <meta-data android:name="android.service.wallpaper"
                android:resource="@xml/wlp" />
        </service>


    </application>

    <uses-sdk android:minSdkVersion="7" />

</manifest>
```

Code Sample



Code Sample


2:22 PM
Set wallpaper
Settings...
2:19 PM
Google
Messaging
Phone
Contacts
Browser
Maps
2:22 PM
abc live wallpaper
possible settings...