

Lab Exercise 6

Please follow the instruction in Workshop Note 6 to complete this exercise.

1. Turn on your computer and startup **Windows XP** (*English version*), download and install **J2ME Wireless Toolkit 2.2** to your computer. Then download the laboratory resource files from <http://www.peter-lo.com/Teaching/I123-1-A/Source6.zip>.
2. Search for some Wav and Midi music in Internet, then create a MIDlet to play audio (WAV format) in local JAR file. (*Page 1 – 2*)
3. Search for some Wav and Midi music in Internet, then create a MIDlet to play audio (Midi format) in local JAR file. (*Page 3 – 4*)
4. Change the above program to play a music file (WAV format) from specified Internet address. (*Page 5 – 6*)
5. Change the above program to play a music file (Midi format) from specified Internet address. (*Page 7 – 8*)
6. Add looping control to the playback the music (*Page 9 – 10*)
7. Use the VolumeControl object to control the volume of the output audio. (*Page 11 – 12*)
8. Set the start time and stop time for the Audio Player. (*Page 13 – 16*)
9. Add an event handler to capture and process the event. (*Page 17 – 18*)
10. Play single tone, and then create the tone sequence for your ring tone. (*Page 19 – 23*)
11. Play MPEG video from Internet. (*Page 24 – 25*)
12. Add background music to the action game you develop in last lesson. (*Page 26 – 27*)
13. Create an animated Xmas card with Christmas song. (*Page 27 – 30*)



14. Apply all the knowledge you learn in past six lessons to develop an action game for the coming Christmas. (*Page 31 - 34*)



1. Play WAV file from JAR file

1. Create two java source files “MyClass.java” and “ImageCanvas.java” in the “src” folder under your project folder (C:\WTK22\apps\MyMultimedia\src).

MyClass.java

```
// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the I/O library
import java.io.*;

/*-----*/
/* MMAPI Demo Program                               */
/*-----*/
public class MyClass extends MIDlet implements CommandListener {
    // Define the GUI components
    private ImageCanvas MyCanvas;           // Canvas
    private Command cmdExit;                // Exit Button
    private Player p;                       // Player
    InputStream infile;                    // Input Buffer

    // Define the no-argument constructor
    public MyClass() {
        // Define the Exit Button with "Exit" label
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the canvas
        MyCanvas = new ImageCanvas();
    }

    // Called by application manager to start the MIDlet
    public void startApp() {
        // Create the Exit Button
        MyCanvas.addCommand(cmdExit);

        // Add the Command Listener
        MyCanvas.setCommandListener(this);

        // Set the current display of the midlet to the Canvas
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Read wav file from resource folder, packaged in the jar file
            infile = getClass().getResourceAsStream("/sample.wav");

            // Set the player to play audio
            p = Manager.createPlayer(infile, "audio/x-wav");

            // Constructs Player without acquiring the scarce and exclusive resources
            p.realize();

            // Acquires the scarce and exclusive resources and processes
            // as much data as necessary to reduce the start latency
            p.prefetch();

            // Start playing multimedia file
            p.start();
        } catch (Exception e) {
```

```

        System.out.print ("Error! Unable to play multimedia file!");
    }
}
// PauseApp is used to suspend background activities and release resources
// on the device when the midlet is not active.
public void pauseApp() {
}

// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
public void destroyApp(boolean unconditional) {
    // Release the resource
    p.deallocate();

    // Close the Player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == cmdExit) {
        destroyApp(true);
        notifyDestroyed();
    }
}
}
}

```

ImageCanvas.java

```

// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;

/*-----*/
/* Canvas for graphic display */
/*-----*/
public class ImageCanvas extends Canvas {
    public void paint(Graphics g) {
        try {
            // Clear the screen
            g.setColor(255, 255, 255);
            g.fillRect(0, 0, getWidth(), getHeight());
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
}

```

2. Compile and execute the MIDlet, you can enjoy the song if you have headphone or speaker.

2. Play Midi file from JAR file

1. Open the previous project “**MyMultimedia**”, and copy the midi file “**sample.mid**” to the resource folder (C:\WTK22\apps\MyMultimedia\res). Don’t forget to delete your previous wav file!
2. Modify the java source file “**MyClass.java**” under your project folder (C:\WTK22\apps\MyMultimedia\src).

```

// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the I/O library
import java.io.*;

/*-----*/
/* MMAPI Demo Program                               */
/*-----*/
public class MyClass extends MIDlet implements CommandListener {
    // Define the GUI components
    private ImageCanvas MyCanvas;           // Canvas
    private Command cmdExit;                // Exit Button
    private Player p;                       // Player
    InputStream infile;                     // Input Buffer

    // Define the no-argument constructor
    public MyClass() {
        // Define the Exit Button with "Exit" label
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the canvas
        MyCanvas = new ImageCanvas();
    }

    // Called by application manager to start the MIDlet
    public void startApp() {
        // Create the Exit Button
        MyCanvas.addCommand(cmdExit);

        // Add the Command Listener
        MyCanvas.setCommandListener(this);

        // Set the current display of the midlet to the Canvas
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Read wav file from resource folder, packaged in the jar file
            infile = getClass().getResourceAsStream("/sample.mid");

            // Set the player to play audio
            p = Manager.createPlayer(infile, "audio/midi");

            // Constructs Player without acquiring the scarce and exclusive resources
            p.realize();

            // Acquires the scarce and exclusive resources and processes
            // as much data as necessary to reduce the start latency
            p.prefetch();
        }
    }
}

```

```
    // Start playing multimedia file
    p.start();
} catch (Exception e) {
    System.out.print ("Error! Unable to play multimedia file!");
}
}
// PauseApp is used to suspend background activities and release resources
// on the device when the midlet is not active.
public void pauseApp() {
}

// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
public void destroyApp(boolean unconditional) {
    // Release the resource
    p.deallocate();

    // Close the Player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == cmdExit) {
        destroyApp(true);
        notifyDestroyed();
    }
}
}
```

3. Compile and execute the MIDlet, can you hear the song from your headphone or speaker?

3. Play WAV file from Internet

1. Open the previous project “MyMultimedia”, and then modify the java source file “MyClass.java” as follow (Don’t forget to delete your previous wav and midi files).

```

// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;

/*-----*/
/* MMAPI Demo Program */
/*-----*/

public class MyClass extends MIDlet implements CommandListener {
    // Define the GUI components
    private ImageCanvas MyCanvas; // Canvas
    private Command cmdExit; // Exit Button
    private Player p; // Player

    // Define the no-argument constructor
    public MyClass() {
        // Define the Exit Button with "Exit" label
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the canvas
        MyCanvas = new ImageCanvas();
    }

    // Called by application manager to start the MIDlet
    public void startApp() {
        // Create the Exit Button
        MyCanvas.addCommand(cmdExit);

        // Add the Command Listener
        MyCanvas.setCommandListener(this);

        // Set the current display of the midlet to the Canvas
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Create the player for playing wav file from Internet
            p = Manager.createPlayer ("http://www.peter-lo.com/Teaching/I123-1-A/sample.wav" );

            // Constructs Player without acquiring the scarce and exclusive resources
            p.realize();

            // Acquires the scarce and exclusive resources and processes
            // as much data as necessary to reduce the start latency
            p.prefetch();

            // Start playing multimedia file
            p.start();
        } catch (Exception e) {
            System.out.print ("Error! Unable to play multimedia file!");
        }
    }

    // PauseApp is used to suspend background activities and release resources
    // on the device when the midlet is not active.
    public void pauseApp() {
    }
}

```

```

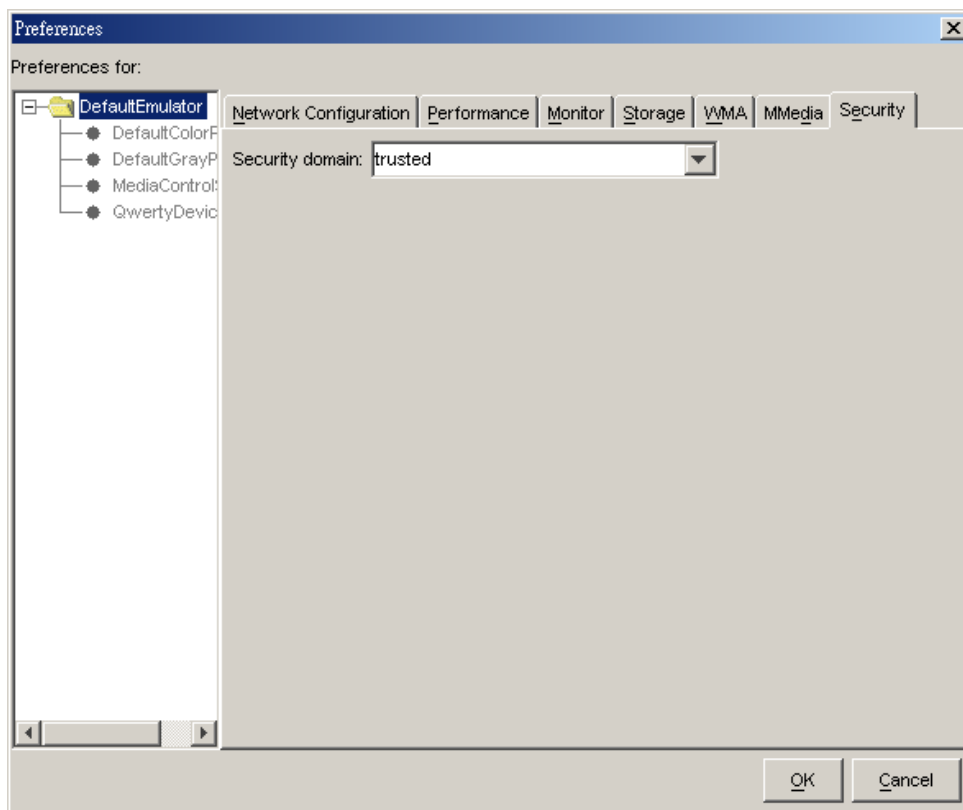
// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
public void destroyApp(boolean unconditional) {
    // Release the resource
    p.deallocate();

    // Close the Player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == cmdExit) {
        destroyApp(true);
        notifyDestroyed();
    }
}
}

```

2. Select **[Edit] → [Preference...]** from the menu bar and switch to “**Security**” tab. Change the security level to “**trusted**” and press **[OK]** to confirm.



3. Compile and execute the MIDlet, you can enjoy the song if you have headphone or speaker.

4. Play Midi file from Internet

1. Open the previous project “MyMultimedia”, and then modify the java source file “MyClass.java” as follow:

```
// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;

/*-----*/
/* MMAPI Demo Program */
/*-----*/
public class MyClass extends MIDlet implements CommandListener {
    // Define the GUI components
    private ImageCanvas MyCanvas; // Canvas
    private Command cmdExit; // Exit Button
    private Player p; // Player

    // Define the no-argument constructor
    public MyClass() {
        // Define the Exit Button with "Exit" label
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the canvas
        MyCanvas = new ImageCanvas();
    }

    // Called by application manager to start the MIDlet
    public void startApp() {
        // Create the Exit Button
        MyCanvas.addCommand(cmdExit);

        // Add the Command Listener
        MyCanvas.setCommandListener(this);

        // Set the current display of the midlet to the Canvas
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Create the player for playing midi file from Internet
            p = Manager.createPlayer ("http://www.peter-lo.com/Teaching/I123-1-A/sample.mid");

            // Constructs Player without acquiring the scarce and exclusive resources
            p.realize();

            // Acquires the scarce and exclusive resources and processes
            // as much data as necessary to reduce the start latency
            p.prefetch();

            // Start playing multimedia file
            p.start();
        } catch (Exception e) {
            System.out.print ("Error! Unable to play multimedia file!");
        }
    }

    // PauseApp is used to suspend background activities and release resources
    // on the device when the midlet is not active.
    public void pauseApp() {
    }
}
```

```
// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
public void destroyApp(boolean unconditional) {
    // Release the resource
    p.deallocate();

    // Close the Player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == cmdExit) {
        destroyApp(true);
        notifyDestroyed();
    }
}
}
```

2. Compile and execute the MIDlet to enjoy your midi song.

5. Simple Audio Playback with Looping

1. Open the previous project “MyMultimedia”, and then modify the java source file “MyClass.java” as follow:

```

// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;

/*-----*/
/* MMAPI Demo Program                               */
/*-----*/

public class MyClass extends MIDlet implements CommandListener {
    // Define the GUI components
    private ImageCanvas MyCanvas;           // Canvas
    private Command cmdExit;                // Exit Button
    private Player p;                       // Player

    // Define the no-argument constructor
    public MyClass() {
        // Define the Exit Button with "Exit" label
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the canvas
        MyCanvas = new ImageCanvas();
    }

    // Called by application manager to start the MIDlet
    public void startApp() {
        // Create the Exit Button
        MyCanvas.addCommand(cmdExit);

        // Add the Command Listener
        MyCanvas.setCommandListener(this);

        // Set the current display of the midlet to the Canvas
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Create the player for playing midi file from Internet
            p = Manager.createPlayer ("http://www.peter-lo.com/Teaching/I123-1-A/sample.wav");

            // Constructs Player without acquiring the scarce and exclusive resources
            p.realize();

            // Replay again it for 3 times
            p.setLoopCount(3);

            // Acquires the scarce and exclusive resources and processes
            // as much data as necessary to reduce the start latency
            p.prefetch();

            // Start playing multimedia file
            p.start();
        } catch (Exception e) {
            System.out.print ("Error! Unable to play multimedia file!");
        }
    }
}

```

```
// PauseApp is used to suspend background activities and release resources
// on the device when the midlet is not active.
public void pauseApp() {
}

// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
public void destroyApp(boolean unconditional) {
    // Release the resource
    p.deallocate();

    // Close the Player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == cmdExit) {
        destroyApp(true);
        notifyDestroyed();
    }
}
}
```

2. Compile and execute the MIDlet to reply your song.

6. Volume Control

1. Open the previous project “MyMultimedia”, and then modify the java source file “MyClass.java” as follow:

```

// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the media control library
import javax.microedition.media.control.*;

/*-----*/
/* MMAPI Demo Program */
/*-----*/
public class MyClass extends MIDlet implements CommandListener {
    // Define the GUI components
    private ImageCanvas MyCanvas; // Canvas
    private Command cmdExit; // Exit Button
    private Player p; // Player
    private VolumeControl vc; // Volume Control

    // Define the no-argument constructor
    public MyClass() {
        // Define the Exit Button with "Exit" label
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the canvas
        MyCanvas = new ImageCanvas();
    }

    // Called by application manager to start the MIDlet
    public void startApp() {
        // Create the Exit Button
        MyCanvas.addCommand(cmdExit);

        // Add the Command Listener
        MyCanvas.setCommandListener(this);

        // Set the current display of the midlet to the Canvas
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Create the player for playing wav file from Internet
            p = Manager.createPlayer ("http://www.peter-lo.com/Teaching/I123-1-A/sample.wav");

            // Constructs Player without acquiring the scarce and exclusive resources
            p.realize();

            // Grab volume control for the player
            vc = (VolumeControl) p.getControl("VolumeControl");

            // Set volume to max (Range: 0 - 100)
            if (vc != null) {
                vc.setLevel(100);
            }

            // Acquires the scarce and exclusive resources and processes
            // as much data as necessary to reduce the start latency
            p.prefetch();
        }
    }
}

```

```
    // Start playing multimedia file
    p.start();
} catch (Exception e) {
    System.out.print ("Error! Unable to play multimedia file!");
}
}

// PauseApp is used to suspend background activities and release resources
// on the device when the midlet is not active.
public void pauseApp() {
}

// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
public void destroyApp(boolean unconditional) {
    // Release the resource
    p.deallocate();

    // Close the Player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == cmdExit) {
        destroyApp(true);
        notifyDestroyed();
    }
}
}
```

2. Compile and execute the MIDlet to enjoy your song. Let's adjust the volume to other value (such as 30) and compile the MIDlet, is it any different you find?

7. Set Player Start Time

1. Open the previous project “MyMultimedia”, and then modify the java source file “MyClass.java” as follow:

```

// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the media control library
import javax.microedition.media.control.*;

/*-----*/
/* MMAPI Demo Program */
/*-----*/
public class MyClass extends MIDlet implements CommandListener {
    // Define the GUI components
    private ImageCanvas MyCanvas; // Canvas
    private Command cmdExit; // Exit Button
    private Player p; // Player

    // Define the constant for 1 second
    final long SECS_TO_MICROSECS = 1000000L;

    // Define the no-argument constructor
    public MyClass() {
        // Define the Exit Button with "Exit" label
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the canvas
        MyCanvas = new ImageCanvas();
    }

    // Called by application manager to start the MIDlet
    public void startApp() {
        // Create the Exit Button
        MyCanvas.addCommand(cmdExit);

        // Add the Command Listener
        MyCanvas.setCommandListener(this);

        // Set the current display of the midlet to the Canvas
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Create the player for playing wav file from Internet
            p = Manager.createPlayer ("http://www.peter-lo.com/Teaching/I123-1-A/sample.wav");

            // Constructs Player without acquiring the scarce and exclusive resources
            p.realize();

            // Set a start time.
            p.setMediaTime(3 * SECS_TO_MICROSECS);

            // Acquires the scarce and exclusive resources and processes
            // as much data as necessary to reduce the start latency
            p.prefetch();

            // Start playing multimedia file
            p.start();
        }
    }
}

```

```
} catch (Exception e) {
    System.out.print ("Error! Unable to play multimedia file!");
}
}

// PauseApp is used to suspend background activities and release resources
// on the device when the midlet is not active.
public void pauseApp() {
}

// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
public void destroyApp(boolean unconditional) {
    // Release the resource
    p.deallocate();

    // Close the Player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == cmdExit) {
        destroyApp(true);
        notifyDestroyed();
    }
}
}
```

2. Compile and execute the MIDlet and prepare the timer, your player will skip the first 3 second of your song.

8. Set Player Stop Time

1. Open the previous project “MyMultimedia”, and then modify the java source file “MyClass.java” as follow:

```

// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the media control library
import javax.microedition.media.control.*;

/*-----*/
/* MMAPI Demo Program */
/*-----*/
public class MyClass extends MIDlet implements CommandListener {
    // Define the GUI components
    private ImageCanvas MyCanvas; // Canvas
    private Command cmdExit; // Exit Button
    private Player p; // Player
    private StopTimeControl stc; // Set Time Control

    // Define the constant for 1 second
    final long SECS_TO_MICROSECS = 1000000L;

    // Define the no-argument constructor
    public MyClass() {
        // Define the Exit Button with "Exit" label
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the canvas
        MyCanvas = new ImageCanvas();
    }

    // Called by application manager to start the MIDlet
    public void startApp() {
        // Create the Exit Button
        MyCanvas.addCommand(cmdExit);

        // Add the Command Listener
        MyCanvas.setCommandListener(this);

        // Set the current display of the midlet to the Canvas
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Create the player for playing wav file from Internet
            p = Manager.createPlayer ("http://www.peter-lo.com/Teaching/I123-1-A/sample.wav");

            // Constructs Player without acquiring the scarce and exclusive resources
            p.realize();

            // Grab Set Time Control for the player
            stc = (StopTimeControl) p.getControl("StopTimeControl");

            // Stop the Player after playing for 3 seconds
            stc.setStopTime(3 * SECS_TO_MICROSECS);

            // Acquires the scarce and exclusive resources and processes
            // as much data as necessary to reduce the start latency

```

```
p.prefetch();

    // Start playing multimedia file
    p.start();
} catch (Exception e) {
    System.out.print ("Error! Unable to play multimedia file!");
}
}

// PauseApp is used to suspend background activities and release resources
// on the device when the midlet is not active.
public void pauseApp() {
}

// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
public void destroyApp(boolean unconditional) {
    // Release the resource
    p.deallocate();

    // Close the Player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == cmdExit) {
        destroyApp(true);
        notifyDestroyed();
    }
}
}
```

2. Compile and execute the MIDlet and prepare the timer, your player should be stopped after playing the song for 3 seconds.

9. Event Handling

1. Open the previous project “MyMultimedia”, and then modify the java source file “MyClass.java” as follow:

```

// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the media control library
import javax.microedition.media.control.*;

/*-----*/
/* MMAPI Demo Program */
/*-----*/
public class MyClass extends MIDlet implements CommandListener, PlayerListener {
    // Define the GUI components
    private ImageCanvas MyCanvas; // Canvas
    private Command cmdExit; // Exit Button
    private Player p; // Player

    // Define the constant for 1 second
    final long SECS_TO_MICROSECS = 1000000L;

    // Define the no-argument constructor
    public MyClass() {
        // Define the Exit Button with "Exit" label
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the canvas
        MyCanvas = new ImageCanvas();
    }

    // Called by application manager to start the MIDlet
    public void startApp() {
        // Create the Exit Button
        MyCanvas.addCommand(cmdExit);

        // Add the Command Listener
        MyCanvas.setCommandListener(this);

        // Set the current display of the midlet to the Canvas
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Create the player for playing wav file from Internet
            p = Manager.createPlayer ("http://www.peter-lo.com/Teaching/I123-1-A/sample.wav");

            // Constructs Player without acquiring the scarce and exclusive resources
            p.realize();

            // Add a player listener
            p.addPlayerListener(this);

            // Acquires the scarce and exclusive resources and processes
            // as much data as necessary to reduce the start latency
            p.prefetch();

            // Start playing multimedia file
            p.start();
        }
    }
}

```

```
} catch (Exception e) {
    System.out.print ("Error! Unable to play multimedia file!");
}
}

// PauseApp is used to suspend background activities and release resources
// on the device when the midlet is not active.
public void pauseApp() {
}

// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
public void destroyApp(boolean unconditional) {
    // Release the resource
    p.deallocate();

    // Close the Player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface
public void commandAction(Command c, Displayable s) {
    if (c == cmdExit) {
        destroyApp(true);
        notifyDestroyed();
    }
}

// Implement the player event handling method
public void playerUpdate(Player p, String event, Object eventData) {
    if (event == END_OF_MEDIA || event == STOPPED_AT_TIME) {
        try {
            // skip for first 5 seconds
            p.setMediaTime(5 * SECS_TO_MICROSECS);

            // Start to play the song
            p.start();
        } catch (MediaException me) {
        }
    }
}
}
```

2. Compile and execute the MIDlet and prepare the timer, your player should be stopped after playing the song for 3 seconds.

10. Play Single Tone

1. Open the previous project “MyMultimedia”, and then modify the java source file “MyClass.java” as follow:

```

// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the media control library
import javax.microedition.media.control.*;

/*-----*/
/* MMAPI Demo Program */
/*-----*/
public class MyClass extends MIDlet implements CommandListener {
    // Define the GUI components
    private ImageCanvas MyCanvas; // Canvas
    private Command cmdExit; // Exit Button

    // Define the no-argument constructor
    public MyClass() {
        // Define the Exit Button with "Exit" label
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the canvas
        MyCanvas = new ImageCanvas();
    }

    // Called by application manager to start the MIDlet
    public void startApp() {
        // Create the Exit Button
        MyCanvas.addCommand(cmdExit);

        // Add the Command Listener
        MyCanvas.setCommandListener(this);

        // Set the current display of the midlet to the Canvas
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Play the C4 tone for 5 second (500 milliseconds) at maximum volume level (100)
            Manager.playTone(ToneControl.C4, 500, 100);
        } catch (Exception e) {
            System.out.print ("Error! Unable to play the tone!");
        }
    }

    // PauseApp is used to suspend background activities and release resources
    // on the device when the midlet is not active.
    public void pauseApp() {
    }

    // DestroyApp is used to stop background activities and release
    // resources on the device when the midlet is at the end of its life cycle.
    public void destroyApp(boolean unconditional) {
    }

    // Implement the event handling method defined in the CommandListener interface.
    public void commandAction(Command c, Displayable s) {

```

```
    if (c == cmdExit) {  
        destroyApp(true);  
        notifyDestroyed();  
    }  
}
```

2. Compile and execute the MIDlet, you can hear the C4 (“DO”) tone for 5 seconds.

11. Play Tone Sequence

1. Open the previous project “MyMultimedia”, and then modify the java source file “MyClass.java” as follow:

```

// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the media control library
import javax.microedition.media.control.*;
// include the I/O library
import java.io.*;

/*-----*/
/* MMAPI Demo Program */
/*-----*/
public class MyClass extends MIDlet implements CommandListener {
    // Define the GUI components
    private ImageCanvas MyCanvas; // Canvas
    private Command cmdExit; // Exit Button
    private Player p; // Player
    private ToneControl tc; // Tone Control

    // Define the note
    final byte full_note = 64; // 1 Note
    final byte half_note = 32; // 1/2 Note
    final byte quarter_note = 16; // 1/4 Note
    final byte eighth_note = 8; // 1/8 Note

    // All notes will be based on Middle C as starting point.
    final byte C4 = (byte) ToneControl.C4; // Middle C
    final byte D4 = (byte) (C4 + 2); // Middle D
    final byte E4 = (byte) (C4 + 4); // Middle E
    final byte F4 = (byte) (C4 + 5); // Middle F
    final byte G4 = (byte) (C4 + 7); // Middle G
    final byte A4 = (byte) (C4 + 9); // Middle A
    final byte B4 = (byte) (C4 + 11); // Middle B
    final byte C5 = (byte) (C4 + 12); // C5

    // Silence for a while
    final byte silence = (byte) ToneControl.SILENCE;

    // Play your song here
    private byte[] tonesequene = new byte[] {
        ToneControl.VERSION, 1, // Define the version
        ToneControl.TEMPO, 30, // Set the Tempo 120bpm
        ToneControl.BLOCK_START, 0, // Define the Block 0
        G4, eighth_note,
        E4, eighth_note,
        E4, eighth_note,
        silence, eighth_note,

        F4, eighth_note,
        D4, eighth_note,
        D4, eighth_note,
        silence, eighth_note,
        ToneControl.BLOCK_END, 0,

        ToneControl.BLOCK_START, 1, // Define the Block 1

```

```

C4, eighth_note,
D4, eighth_note,
E4, eighth_note,
F4, eighth_note,
G4, eighth_note,
G4, eighth_note,
G4, eighth_note,
silence, eighth_note,
ToneControl.BLOCK_END, 1,

ToneControl.BLOCK_START, 2,           // Define the Block 2
C4, eighth_note,
E4, eighth_note,
G4, eighth_note,
G4, eighth_note,
C4, quarter_note,
silence, eighth_note,
ToneControl.BLOCK_END, 2,

ToneControl.BLOCK_START, 3,           // Define the Block 3
D4, eighth_note,
D4, eighth_note,
D4, eighth_note,
D4, eighth_note,
D4, eighth_note,
E4, eighth_note,
F4, eighth_note,
silence, eighth_note,

E4, eighth_note,
E4, eighth_note,
E4, eighth_note,
E4, eighth_note,
E4, eighth_note,
F4, eighth_note,
G4, eighth_note,
silence, eighth_note,
ToneControl.BLOCK_END, 3,

ToneControl.PLAY_BLOCK, 0,           // Play the Block 0
ToneControl.PLAY_BLOCK, 1,           // Play the Block 1
ToneControl.PLAY_BLOCK, 0,           // Play the Block 0
ToneControl.PLAY_BLOCK, 2,           // Play the Block 2
ToneControl.PLAY_BLOCK, 3,           // Play the Block 3
ToneControl.PLAY_BLOCK, 0,           // Play the Block 0
ToneControl.PLAY_BLOCK, 2,           // Play the Block 2
};

// Define the no-argument constructor
public MyClass() {
    // Define the Exit Button with "Exit" label
    cmdExit = new Command("Exit", Command.EXIT, 0);

    // Define the canvas
    MyCanvas = new ImageCanvas();
}

// Called by application manager to start the MIDlet
public void startApp() {
    // Create the Exit Button
    MyCanvas.addCommand(cmdExit);

    // Add the Command Listener
    MyCanvas.setCommandListener(this);

    // Set the current display of the midlet to the Canvas
    Display.getDisplay(this).setCurrent(MyCanvas);

    try {
        // Create the player

```



```

    p = Manager.createPlayer(Manager.TONE_DEVICE_LOCATOR);

    // Constructs Player without acquiring the scarce and exclusive resources
    p.realize();

    // Enable playback sequence by creating tone control
    tc = (ToneControl)p.getControl("javax.microedition.media.control.ToneControl");

    // Reference the sequence of bytes we created
    tc.setSequence(tonesequence);

    // Acquires the scarce and exclusive resources and processes
    // as much data as necessary to reduce the start latency
    p.prefetch();

    // Start playing multimedia file
    p.start();
} catch (MediaException e) {
} catch (IOException ioe) {
}
}

// PauseApp is used to suspend background activities and release resources
// on the device when the midlet is not active.
public void pauseApp() {
}

// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
public void destroyApp(boolean unconditional) {
    // Release the resource
    p.deallocate();

    // Close the Player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == cmdExit) {
        destroyApp(true);
        notifyDestroyed();
    }
}
}
}

```

2. Compile and execute the MIDlet and enjoy the song I design for you. Now you can edit the variable “*tonesequence*” to design your own song.

12. Play MPEG from Internet

1. Create a new project, name the project and the class at “*VideoPlayer*” and “*MyClass*”. Then create a java source file called “*MyClass.java*” in the “*src*” folder under your project folder (C:\WTK22\apps\VideoPlayer\src).

```
// include the MIDP library
import javax.microedition.midlet.*;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the media control library
import javax.microedition.media.control.*;
// include the I/O library
import java.io.*;

/*-----*/
/* MMAPI Demo Program */
/*-----*/
public class MyClass extends MIDlet implements CommandListener {
    private Display midletDisplay; // Reference to Display object
    private Form frmVideo; // Form
    private Command cmdExit; // Exit Button
    private Player p; // Player
    private VideoControl vc; // Video Control
    private Item itmVideo; // Custom Item for Video Control

    // Define the no-argument constructor
    public MyClass() {
        // Retrieve the display from the static display object
        midletDisplay = Display.getDisplay(this);

        // Define the Exit Button with "Exit" label
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the canvas
        frmVideo = new Form("Playing Video...");
    }

    public void startApp() {
        try {
            // Create the player to play MPEG
            p = Manager.createPlayer("http://www.peter-lo.com/Teaching/I123-1-A/sample.mpg");

            // Constructs Player without acquiring the scarce and exclusive resources
            p.realize();

            // Grab the video control and set it to the current display.
            vc = (VideoControl)p.getControl("VideoControl");

            if (vc != null) {
                // Create a custom item for video control
                itmVideo = (Item)vc.initDisplayMode(vc.USE_GUI_PRIMITIVE, null);

                // Create Form, add Commands and video item, listen for events
                frmVideo.append(itmVideo);
                frmVideo.addCommand(cmdExit);
                frmVideo.setCommandListener(this);

                // Set the current display of the midlet to the Form
                midletDisplay.setCurrent(frmVideo);
            }
        }
    }
}
```

```

// Acquires the scarce and exclusive resources and processes
// as much data as necessary to reduce the start latency
p.prefetch();

// Start playing multimedia file
p.start();
} catch (IOException ioe) {
p = null;
} catch (MediaException me) {
p = null;
}

// Set the current display of the midlet to the Form
midletDisplay.setCurrent(frmVideo);
}

// PauseApp is used to suspend background activities and release resources
// on the device when the midlet is not active.
protected void pauseApp() {
}

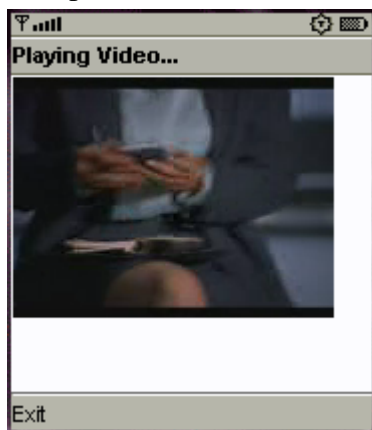
// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
protected void destroyApp(boolean unconditional) {
// Release the resource
p.deallocate();

// Close the Player
p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
if (c == cmdExit) {
destroyApp(true);
notifyDestroyed();
}
}
}
}

```

2. Compile and execute the MIDlet, can you see the movie on your screen?



13. Add Background Music to your Action Game

1. Open the action game project “**Action**” you created last week, and put the midi file “**action.mid**” to the resource folder (C:\WTK22\apps\Action\res). Then modify the source file “**MyClass.java**” in the “**src**” folder (C:\WTK22\apps\Action\src) as follow:

```
// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the I/O library
import java.io.*;

/*****
/* Sample Action Game.
*****/
public class MyClass extends MIDlet implements CommandListener {
    // Define the GUI components
    private ImageCanvas MyCanvas;           // Canvas
    private Command cmdExit;                // Exit Button
    private Command cmdStart;               // Start Button
    private Thread MyThread;                // Thread
    private Player p;                        // Player
    InputStream infile;                      // Input Buffer

    // Define the no-argument constructor
    public MyClass() {
        // Define the Exit Button
        cmdExit = new Command("Exit", Command.EXIT, 0);

        // Define the Start Button
        cmdStart = new Command("Start", Command.SCREEN, 1);

        // Define the canvas
        MyCanvas = new ImageCanvas();

        try {
            // Read midi file, packaged in the jar file
            infile = getClass().getResourceAsStream("/action.mid");

            // Set the player to play midi
            p = Manager.createPlayer(infile, "audio/midi");

            // Constructs Player without acquiring the scarce and exclusive resources
            p.realize();

            // Define the play back
            p.setLoopCount(99);

            // Acquires the scarce and exclusive resources and processes
            // as much data as necessary to reduce the start latency
            p.prefetch();

            // Start playing multimedia file
            p.start();
        } catch (Exception e) {
            System.out.print ("Error! Unable to play Midi file!");
        }
    }
}
```

```

// Called by application manager to start the MIDlet
public void startApp() {
    // Add the Command button and the Command Listener
    MyCanvas.addCommand(cmdExit);
    MyCanvas.addCommand(cmdStart);
    MyCanvas.setCommandListener(this);

    // Set the current display of the midlet to the Canvas
    Display.getDisplay(this).setCurrent(MyCanvas);
}

// PauseApp is used to suspend background activities and release resources
// on the device when the midlet is not active.
public void pauseApp() {
}

// DestroyApp is used to stop background activities and release
// resources on the device when the midlet is at the end of its life cycle.
public void destroyApp(boolean unconditional) {
    // Release the resource
    p.deallocate();

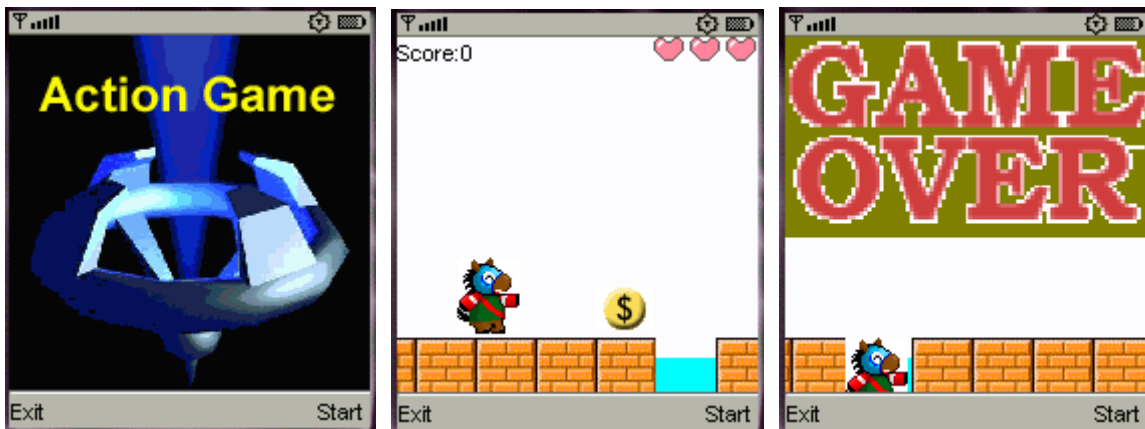
    // Close the Player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == cmdStart) {
        // Start the thread only one time
        if (MyThread == null) {
            // Define the thread
            MyThread = new Thread(MyCanvas);

            // Start the thread
            MyThread.start();
        }
    } else if (c == cmdExit) {
        destroyApp(true);
        notifyDestroyed();
    }
}
}

```

2. Compile and execute your MIDlet, it becomes a complete game this time.



14. Xmas Card

1. Create a new project, name the project and the class at “XmasCard” and “XmasCard”.
2. Put the following PNG images and the Christmas song file into the “res” folder.



3. Then create two java files named “XmasCard.java” and “ImageCanvas.java” in the “src” folder.

XmasCard.java

```
// include the MIDP library
import javax.microedition.midlet.*;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the I/O library
import java.io.*;

/*-----*/
/* Java class: XmasCard.java */
/*-----*/
/* Multimedia Xmas Card */
/*-----*/
public final class XmasCard extends MIDlet implements CommandListener {
    private ImageCanvas MyCanvas = new ImageCanvas();
    private Thread MyThread = new Thread(MyCanvas);
    private Command mExitCommand = new Command("Exit", Command.EXIT, 0);
    private Player p;

    public XmasCard() {
        // Start the thread
        MyThread.start();
    }

    public void startApp() {
        MyCanvas.addCommand(mExitCommand);
        MyCanvas.setCommandListener(this);
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Read sound file, packaged in the jar file
            InputStream in = getClass().getResourceAsStream("/xmas.mid");
```

```

// Create the player
p = Manager.createPlayer(in, "audio/midi");

// Allocate player resources
p.realize();

// Define the play back
p.setLoopCount(3);

// the player can start with the smallest latency
p.prefetch();

// Start playing Wav file
p.start();
} catch (Exception e) {
    System.out.print ("Error! Unable to play Midi file!");
}
}

// Implement the pauseApp() method
public void pauseApp() {
}

// implement the destroyApp() method
public void destroyApp(boolean unconditional) {
    // Close the player
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == mExitCommand) {
        destroyApp(false);
        notifyDestroyed();
    }
}
}
}

```

ImageCanvas.java

```

// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;

/*-----*/
/* Java class: ImageCanvas.java */
/*-----*/
/* Canvas for graphic display */
/*-----*/
public class ImageCanvas extends Canvas {
    private static Image img[] = new Image[10];
    private int ImageNum = 0;
    private String imageName;

    public void run() {
        try {
            for (int i=0 ; i<4; i++) {
                imageName = "/xmas" + i + ".png";
                img[i] = Image.createImage(imageName);
            }

            while (true) {
                repaint();
                Thread.sleep(500);
            }
        } catch (Exception e) {}
    }
}

```

```
public synchronized void paint(Graphics g) {  
    // Clear the screen  
    g.setColor(49, 0, 100);  
    g.fillRect(0, 0, 180, 180);  
  
    // Draw the image  
    g.drawImage(img[ImageNum++], 22, 0, g.LEFT|g.TOP);  
    if (ImageNum >= 3) {  
        ImageNum = 0;  
    }  
}
```

4. Compile and execute the program. You can send it to your friends to celebrate the Xmas.



15. Xmas Mini Game (Catch the Gift)

1. Create a new project, name the project and the class at “XmasGirl” and “XmasGirl”.
2. Put the following PNG images and the Christmas song file into the “res” folder.



3. Create two java files named “XmasGirl.java” and “ImageCanvas.java” in the “src” folder.

XmasGirl.java

```
// include the MIDlet supper class
import javax.microedition.midlet.MIDlet;
// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// include the media library
import javax.microedition.media.*;
// include the I/O library
import java.io.*;

/*-----*/
/* Java class: XmasGirl.java */
/*-----*/
/* Sample Xmas Action Game. */
/*-----*/
public class XmasGirl extends MIDlet implements CommandListener {
    private ImageCanvas MyCanvas = new ImageCanvas();
    private Thread MyThread = new Thread(MyCanvas);
    private Command mStartCommand = new Command("Start", Command.SCREEN, 0);
    private Command mExitCommand = new Command("Exit", Command.EXIT, 0);
    private Player p;

    // implement the startApp() method
    public void startApp() {
        MyCanvas.addCommand(mStartCommand);
        MyCanvas.addCommand(mExitCommand);
        MyCanvas.setCommandListener(this);
        Display.getDisplay(this).setCurrent(MyCanvas);

        try {
            // Read midi file, packaged in the jar file
            InputStream in = getClass().getResourceAsStream("/xmasgirl.mid");
            // Create the player
            p = Manager.createPlayer(in, "audio/midi");
            // Allocate player resources
            p.realize();
            // Define the play back

```

```

        p.setLoopCount(3);
        // the player can start with the smallest latency
        p.prefetch();
        // Start playing midi file
        p.start();
    } catch (Exception e) {
        System.out.print ("Error! Unable to play Midi file!");
    }
}

// implemente the pauseApp() method
public void pauseApp() {
}

// implement the destroyApp() method
public void destroyApp(boolean unconditional) {
    p.close();
}

// Implement the event handling method defined in the CommandListener interface.
public void commandAction(Command c, Displayable s) {
    if (c == mExitCommand) {
        destroyApp(false);
        notifyDestroyed();
    } else if (c == mStartCommand) {
        MyThread.start();
    }
}
}
}

```

ImageCanvas.java

```

// include the GUI libraries of MIDP
import javax.microedition.lcdui.*;
// import the random library
import java.util.Random;

/*-----*/
/* Java class: ImageCanvas.java */
/*-----*/
/* Canvas for graphic display */
/*-----*/
public class ImageCanvas extends Canvas implements Runnable {
    // Random seek
    private static Random rand = new Random();
    // Image in the game
    private static Image imgGift0;
    private static Image imgSanta;
    private static Image imgLife;
    private static Image imgOpening;
    private static Image imgGameOver;

    // Maximum number of Gift Drop
    private int maxGiftNumber = 4;
    // Level
    private int level = 1;
    // Score point
    private int point = 1;

    // Position of the Xmas Girl
    private static int SantaX = 45;
    private static int SantaY = 125;

    // Screen mode (Opening / Playing)
    private static int ScreenMode = 0;

    // Gift position (X, Y) and speed V
    private static int[] GiftX = new int[15];
    private static int[] GiftY = new int[15];
}

```

```

private static int[] GiftV = new int[15];
// Current Life for the Xmas Girl
private static int CurrentLife = 3;

// Looping variable
private static int i;

// Display the Score
private static int score = 0;

public void run() {
    try {
        // Initialize the Gift position
        for (i=0; i<maxGiftNumber; i++) {
            GiftX[i] = 0;
            GiftY[i] = -18;
            GiftV[i] = 0;
        }

        // Load the image
        imgGift0 = Image.createImage("/gift1.png");
        imgLife = Image.createImage("/life.png");
        imgGameOver = Image.createImage("/gameover.png");
        imgSanta = Image.createImage("/santa.png");

        while (true) {
            if (CurrentLife > 0) {
                ScreenMode = 1;
            }

            // Redraw the screen
            repaint();
            Thread.sleep(100);
        }
    } catch (Exception e) {}
}

protected synchronized void keyPressed(int keyCode) {
    if (getGameAction(keyCode) == LEFT && SantaX > 0) {
        SantaX -= 45;
    } else if (getGameAction(keyCode) == RIGHT && SantaX < 135) {
        SantaX += 45;
    }
}

public synchronized void paint(Graphics g) {
    try {
        if (ScreenMode == 1) {
            // Clear the screen
            g.setColor(255, 255, 255);
            g.fillRect(0, 0, 180, 180);
            // Draw the Xmas Girl
            g.drawImage(imgSanta, SantaX, SantaY, g.TOP|g.LEFT);
            // Draw the missile
            for (i=0; i<maxGiftNumber; i++) {
                if (GiftV[i] > 0) {
                    GiftY[i] = GiftY[i] + GiftV[i];
                    // Handle the event for Gift Drop to the ground
                    if (GiftY[i] > 150) {
                        // Reduce the Life
                        CurrentLife = CurrentLife - 1;
                        if (CurrentLife <= 0) {
                            ScreenMode = 9;
                        }
                    }
                    // Initialize the Gift position
                    for (i=0; i<maxGiftNumber; i++) {
                        GiftX[i] = 0;
                        GiftY[i] = -18;
                        GiftV[i] = 0;
                    }
                }
            }
        }
    }
}

```

```

    }
    // Pause the screen for 2 seconds
    Thread.sleep(2000);
} else if (SantaX == GiftX[i] && SantaY <= GiftY[i]+30) {
    // Identify the xmas girl catch the gift
    GiftY[i] = -18;
    GiftV[i] = 0;
    score = score + point;
    if ((score > 10*level*point) && (level < 10)) {
        level++;
        point++;
        maxGiftNumber++;
    }
}
} else if (Math.abs(rand.nextInt())%5 == 1) {
    GiftX[i] = Math.abs(rand.nextInt())%4*45;
    GiftY[i] = -18;
    GiftV[i] = Math.abs(rand.nextInt())%4 + level;
}
g.drawImage(imgGift0, GiftX[i], GiftY[i], g.TOP|g.LEFT);
}

g.setColor(255, 255, 255);
g.fillRect(0, 0, 180, 12);
// Draw the score
g.setColor(0, 0, 0);
g.drawString("Score:" + score, 0, 0, g.TOP|g.LEFT);
g.drawString("Level:" + level, 80, 0, g.TOP|g.LEFT);
// Display the number of life
for (i=1; i<=CurrentLife; i++) {
    g.drawImage(imgLife, 110+i*18, 0, g.TOP|g.LEFT);
}
} else if (ScreenMode == 9) {
    // Display the Game Over image
    g.setColor(255, 255, 255);
    g.fillRect(0, 0, 180, 180);
    g.drawImage(imgGameOver, 0, 0, g.TOP|g.LEFT);
} else {
    // Display the opening image
    imgOpening = Image.createImage("/opening.png");
    g.drawImage(imgOpening, 0, 0, g.TOP|g.LEFT);
}
} catch(Exception e) {
    System.out.println(e.getMessage());
}
}
}
}

```

4. Compile and execute the program. Hope you enjoy the game.

