Saving a captured image in Java ME

This code example demonstrates how to take a picture by camera and save it to file.

Overview

Perform the following steps:

1. Check support of video capturing by calling the `System.getProperty` method with the `supports.video.capture` parameter.
2. Call `Manager.createPlayer` method to create a camera player.
3. Initialize and start player.
4. Get VideoControl object from the player and execute the `VideoControl.getSnaphot` method. This method returns array of bytes which represents captured image in specified format.
5. Open file and send array of bytes to it. After this, the file should be closed.
6. On exit from application, the camera player will be stopped and closed.

Source file: CaptureAndSaveImage.java

```
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Item;
import javax.microedition.lcdui.Alert;

import javax.microedition.media.Manager;
import javax.microedition.media.Player;
import javax.microedition.media.control.VideoControl;
import javax.microedition.media.MediaException;

import javax.microedition.io.Connector;
import javax.microedition.io.file.FileConnection;

import java.io.OutputStream;
import java.io.IOException;

public class CaptureAndSaveImage extends MIDlet implements CommandListener {

    private Display display;

    // Form where camera viewfinder is placed
    private Form cameraForm;

    // Command for capturing image by camera and saving it.
    // Placed in cameraForm.
    private Command cmdCapture;
    // Command for exiting from midlet. Placed in cameraForm.
    private Command cmdExit;

    // Player for camera
    private Player player;
    // Video control of camera
```


(C) Copyright Nokia 2014. All rights reserved.

```
    private VideoControl videoControl;

    // Alert to be displayed if error occurs.
    private Alert alert;

    /**
     * Constructor.
     */
    public CaptureAndSaveImage() {
        InitializeComponents();
    }

    /**
     * Initializes components of midlet.
     */
    private void InitializeComponents() {
        display = Display.getDisplay(this);

        if(checkCameraSupport() == false) {
            showAlert("Alert", "Camera is not supported!", null);
            return;
        }

        try {
            createCameraForm();
            createCamera();
            addCameraToForm();
            startCamera();
        } catch(IOException ioExc) {
            showAlert("IO error", ioExc.getMessage(), null);
        } catch(MediaException mediaExc) {
            showAlert("Media error", mediaExc.getMessage(), null);
        }
    }

    /**
     *  Creates and returns form where the camera control will be placed.
     */
    private void createCameraForm() {
        // Create camera form
        cameraForm = new Form("Camera");
        // Create commands for this form
        cmdCapture = new Command("Capture", Command.OK, 0);
        cmdExit = new Command("Exit", Command.EXIT, 0);
        // Add commands to form
        cameraForm.addCommand(cmdCapture);
        cameraForm.addCommand(cmdExit);
        // Set midlet as command listener for this form
        cameraForm.setCommandListener(this);
    }

    /**
     * Check camera support.
     * @return true if camera is supported, false otherwise.
     */
    private boolean checkCameraSupport() {
        String propValue = System.getProperty("supports.video.capture");
```


(C) Copyright Nokia 2014. All rights reserved.

```
        return (propValue != null) && propValue.equals("true");
    }

    /**
     * Creates camera control and places it to cameraForm.
     */
    private void createCamera() {
try {
    player = Manager.createPlayer("capture://image");
}
catch (MediaException e) {
    try {
        System.out.println("Couldn't set \"capture://video\"");
     player = Manager.createPlayer("capture://video");
    }
    catch (MediaException ex) {

    }
    catch (IOException ex) {

    }
}
catch (IOException e) {

}

try {
    player.realize();
    player.prefetch();
}
catch (MediaException e) {

}

videoControl = (VideoControl)player.getControl("VideoControl");
    }

    /**
     * Adds created camera as item to cameraForm.
     */
    private void addCameraToForm() {
        cameraForm.append((Item)videoControl.
                initDisplayMode(VideoControl.USE_GUI_PRIMITIVE, null));
    }

    /**
     * Start camera player
     * @throws IOException if starting of player is failed.
     * @throws MediaException if starting of player is failed.
     */
    private void startCamera() throws IOException, MediaException  {
        if(player.getState() == Player.PREFETCHED) {
            player.start();
        }
    }
```


(C) Copyright Nokia 2014. All rights reserved.

```java
    /**
     * Saves image captured by camera.
     */
    private void captureAndSaveImage() {
        FileConnection file = null;
        OutputStream outStream = null;

        try {
            if(checkPngEncodingSupport() == false) {
                throw new Exception("Png encoding is not supported!");
            }

            // Capture image
            byte[] capturedImageData =
                    videoControl.getSnapshot("encoding=png");

            // Get path to photos folder.
            String dirPhotos = System.getProperty("fileconn.dir.photos");
            if(dirPhotos == null) {
                throw new Exception("Unable get photos folder name");
            }

            String fileName = dirPhotos + "CapturedImage.png";
            // Open file
            file = (FileConnection)Connector.open(fileName,
                    Connector.READ_WRITE);
            // If there is no file then create it
            if(file.exists() == false) {
                file.create();
            }
            // Write data received from camera while making snapshot to file
            outStream = file.openOutputStream();
            outStream.write(capturedImageData);

            showAlert("Info", "Image is saved in " + fileName, cameraForm);

        } catch(IOException ioExc) {
            showAlert("IO error", ioExc.getMessage(), cameraForm);
        } catch(MediaException mediaExc) {
            showAlert("Media error", mediaExc.getMessage(), cameraForm);
        } catch(Exception exc) {
            showAlert("Error", exc.getMessage(), cameraForm);
        } finally {
            // Try to close file
            try {
                if(outStream != null) {
                    outStream.close();
                }
                if(file != null) {
                    file.close();
                }
            } catch(Exception exc) {
                // Do nothing
            }
        }
    }
```


(C) Copyright Nokia 2014. All rights reserved.

```
    /**
     * Checks png encoding support
     * @return true if png encoding is supported false otherwise.
     */
    private boolean checkPngEncodingSupport() {
        String encodings = System.getProperty("video.snapshot.encodings");
        return (encodings != null) && (encodings.indexOf("png") != -1);
    }

    /**
     * From MIDlet.
     * Signals the MIDlet that it has entered the Active state.
     */
    public void startApp() {
        if ( videoControl != null ) {
            display.setCurrent(cameraForm);
        }
    }

    /**
     * From MIDlet.
     * Signals the MIDlet to enter the Paused state.
     */
    public void pauseApp() {
        // TODO: pause player if it is running.
    }

    /**
     * Performs exit from midlet.
     */
    public void exitMIDlet() {
        notifyDestroyed();
    }

    /**
     * Shows alert with specified title and text. If next displayable is not
     * specified then application will be closed after alert closing.
     * @param title - Title of alert.
     * @param message - text of alert.
     * @param nextDisp - next displayable. Can be null.
     */
    private void showAlert(String title, String message, Displayable nextDisp) {
        alert = new Alert(title);
        alert.setString(message);
        alert.setTimeout(Alert.FOREVER);

        if(nextDisp != null) {
            display.setCurrent(alert, nextDisp);
        } else {
            display.setCurrent(alert);
            alert.setCommandListener(this);
        }
    }

    /**
     * From MIDlet.
```


(C) Copyright Nokia 2014. All rights reserved.

```
     * Signals the MIDlet to terminate and enter the Destroyed state.
     */
    public void destroyApp(boolean unconditional) {
        if(player != null) {
            player.deallocate();
            player.close();
        }
    }

    /**
     * From CommandListener.
     * Indicates that a command event has occurred on Displayable displayable.
     * @param command - a Command object identifying the command.
     * @param displayable - the Displayable on which this event has occurred.
     */
    public void commandAction(Command command, Displayable displayable) {
        // Handles "Capture image" command from cameraForm
        if(command == cmdCapture) {
            captureAndSaveImage();
        }
        // Handles "exit" command from forms
        if(command == cmdExit) {
            exitMIDlet();
        }
        // Handle "ok" command from alert
        if(displayable == alert) {
            exitMIDlet();
        }
    }
}
```

Postconditions

Camera viewfinder is shown on display.

By choosing "Capture image" menu command, the user can take a snapshot and save it to CapturedImage.png file in photos folder. Image will be saved to file with PNG encoding.

Supplementary material

Executables and source files can be found in Media:SavingCapturedImage.zip.

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.