When Pigs Fly

May 2013
ver 1.1

Created by: Kayla Johnson
matsukeira@gmail.com

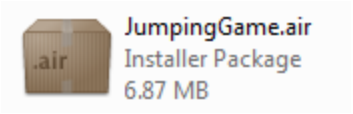
Table of Contents

I. OVERVIEW

When Pigs Fly is a platformer game that allows victims of stroke to exercise their finger dexterity and strength. The user plays the role of Hamlett the pig who wishes to escape the farm. Through the use of an Arduino pressure sensor, the user makes the pig jump up clouds in level one and flap his wings in level two to win the game.

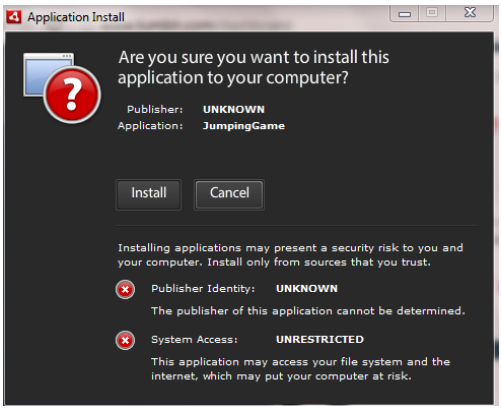
I. INSTALLATION

A. INSTALLING THE GAME



Run *JumpingGame.air* file to begin installation.

Click *Install* when the dialog box pops up.



Then click *Continue*; there's no need to change the destination folder. After that the installation will continue on its own.

By default, the program installs in *C:\Program Files (x86)\When Pigs Fly*

When the installation is complete, an icon will be added to your desktop and the game will open up automatically. However, before you can play, Arduino drivers must be installed and running.

<u>If you already have the program installed</u> and need to install a newer version at a later time, there is a chance you'll see an error. If you receive an error that the installer file is corrupt, go into the computer's *control panel* and *uninstall a program* dialog box. Then uninstall the current version of *When Pigs Fly*. Then you should be able to double click the *JumpingGame.air* file again and the install will go through as it should.

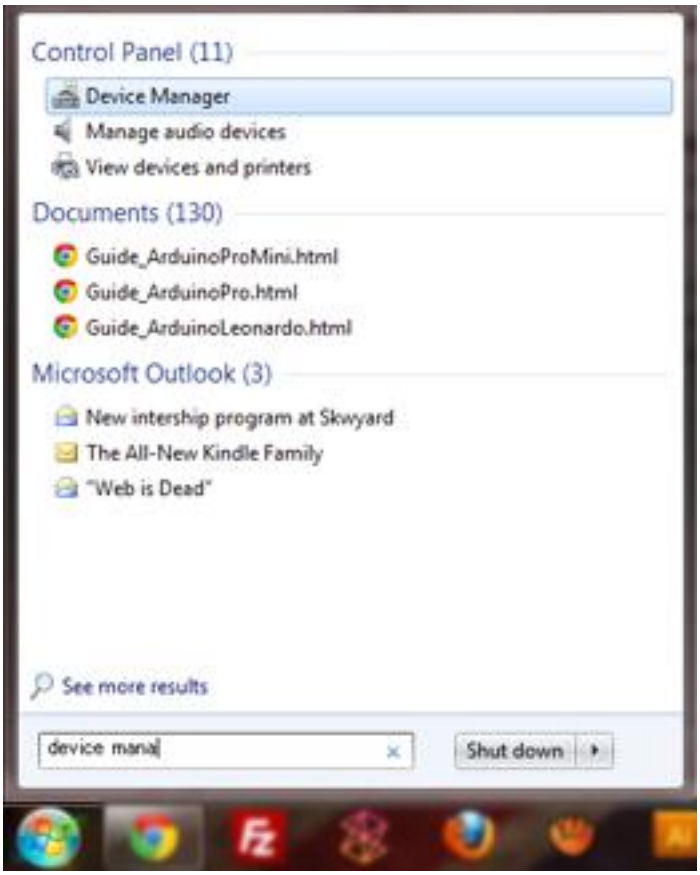
B. INSTALLING ARDUINO

If you haven't already, make sure that the *arduino for when pigs fly and bale jump* folder is on the desktop, complete with *analogRead, drivers,* and *serproxy* folders. These are vital to the functionality of the game.

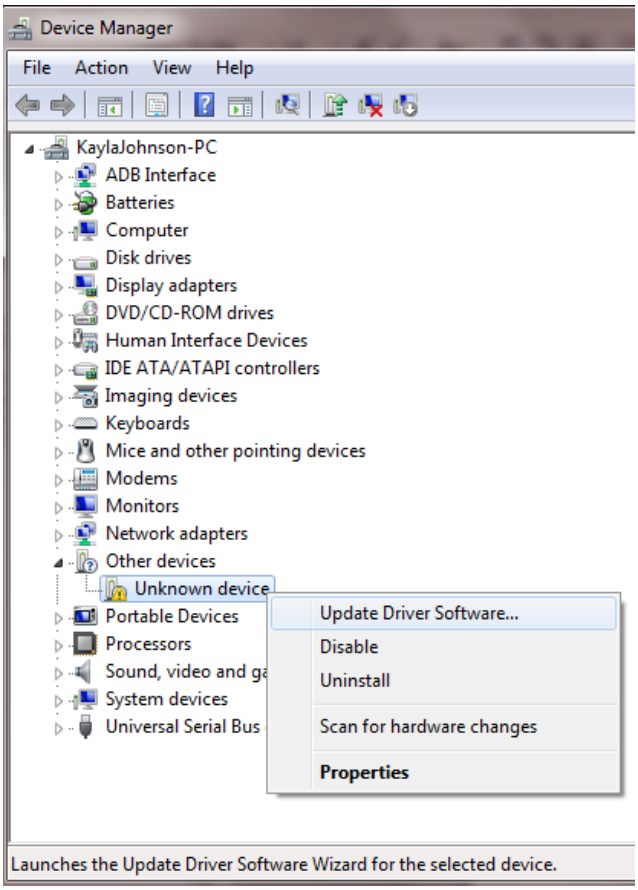
Then plug in the Arduino Pressure Sensor. Having the squeeze ball on the end of the sensor is optional; the game will function the same with or without it.

The first time you plug in the Arduino pressure sensor to the USB port of a computer it will install the device. However it won't be able to find the drivers for it. You need to manually install them.

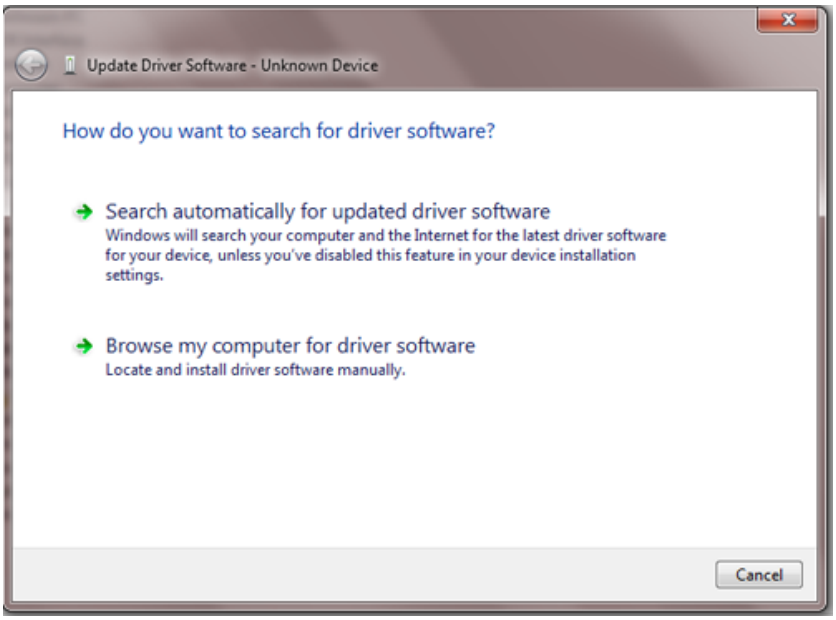
1. Open up *Device Manager*



2. Under *Other devices,* right-click the *Unknown device* and click *Update Driver Software…*

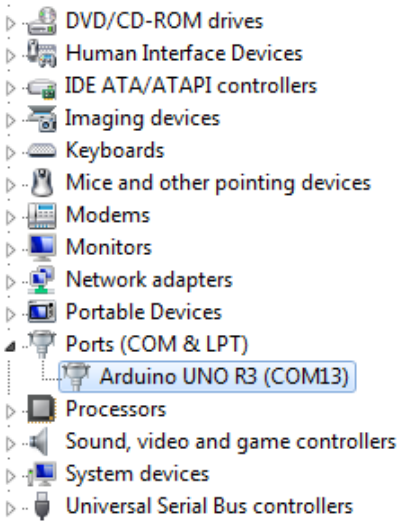


3. Click *Browse my computer for driver software.* Then navigate to the *arduino* folder on your desktop and select the drivers folder.



4. If Windows gives you a warning, Click *Install this driver software anyway*

If you've done this correctly, the *Unknown device* from earlier will now be an *Arduino UNO R3* with a specified numbered COM port 1 to 256 (*COM13* as shown).



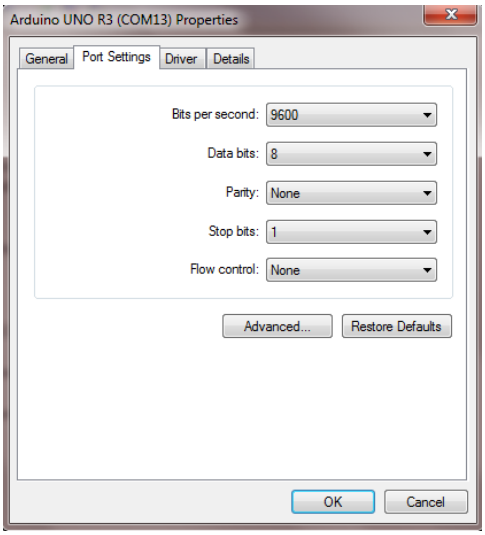
C. CONFIGURING ARDUINO

If you've completed the steps above correctly it's time to configure the Arduino device for gameplay. Arduino cannot function properly when set to a COM port higher than 9, therefore we need to change the COM port the device is operating on (if it's higher than 9) as well as changing the configuration file.

This should only need to be done once. Once you plug in the Arduino pressure sensor to the USB port and set the COM port, it should be set to the same COM port subsequent times it's plugged in. However, if at a subsequent time, that specified COM port is in use from another USB device (like a mouse, flash drive, etc), the Arduino pressure sensor will take the next available COM port number. If this happens, you will have to modify the configuration file as demonstrated below.

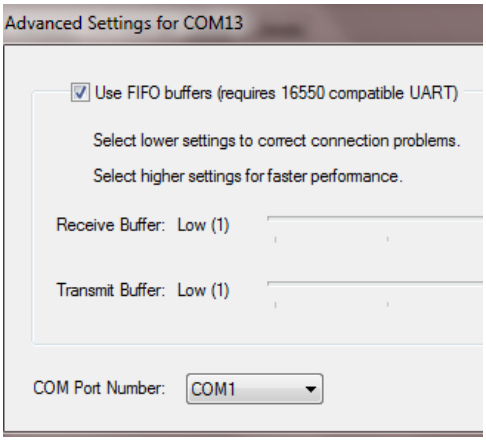
For example, if you set the COM port for the Arduino pressure sensor to COM1 and on a later day want to play the game and there is a flash drive or other USB device plugged in, there is a *chance* that one of those devices may be using COM1 because it was the first one available. In that case you will need to modify to the configuration file to match the COM port number that the Arduino pressure sensor has taken place of its usual COM port.

i. CONFIGURING THE COM PORT NUMBER



Still in the device manager window, right click the *Arduino Uno R3* and click *Properties.*

Click the *Port Settings* tab on the top, and then click the *Advanced…* button as shown on the right.



In the drop-down list of COM ports, choose the smallest port that's not "in use."

If there are no COM ports available with a number less than 10, check to see if you have anything else plugged in the USB drives. If they're not vital, try unplugging the unnecessary devices as well as the Arduino pressure sensor. Then plug the Arduino pressure sensor back in and repeat the earlier steps. If any lower ports were freed up, it should take the lowest one available.

As seen on the left, I'm using COM1. Leave the other settings the same.

Then click OK on the dialog box. Click OK again on the *Properties* dialog box and close the *Device Manager* window.

ii. CHANGING THE CONFIGURATION FILE TO MATCH THE COM PORT NUMBER

Once you've changed the COM port number to a number below 10, it's time to change the configuration file to match the corresponding port number. Every time you change the COM port the Arduino pressure sensor is running on, you need to change the configuration file.

It's recommended to check which COM port the device is using every time you plug it in to the computer. It shouldn't change unless there is another device using its usual COM port (as described before), but it's better to be safe and check.

In the *arduino* folder that's on your desktop, navigate to the *serproxy* folder and open up the *serproxy.cfg* file in a text editor. It should open up in your default text editor if you double click it.

Once opened inside the text editor, the only two lines that need to be changed are

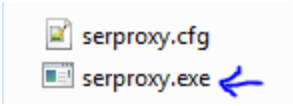
`comm_ports=#` and `net_port#=5331` where # is the number your COM port that you selected in the first step.

As shown below, I'm using COM1 so `comm_ports` is set to 1 and `net_port1=5331`.

```
# Transform newlines coming from the serial port into nils
# true (e.g. if using Flash) or false
newlines_to_nils=true

# on a mac you will need to add this
# serial_device1=/dev/tty.usbserial-A6004osh


# Comm ports used
comm_ports=1

# Default settings
#comm_baud=57600
comm_baud=9600
comm_databits=8
comm_stopbits=1
comm_parity=none

# Idle time out in seconds
timeout=300

# Port settings (ttyS0)
net_port1=5331
```

Once you've changed these two lines only, you can save and close the configuration file.

II. Playing The Game



Every time you play the game, before running the application, you need to make sure the pressure sensor is plugged in to the USB port and then run *serproxy.exe* file located in the *serproxy* folder of the *arduino* folder on your desktop. A black box will pop up. Don't close it until users are done playing the game; minimize the window during gameplay.

A. Understanding the Home Screen and its menus

The home screen of the game has three buttons the user can click: *play, how to,* and *calibrate.*

i. Play

Clicking this button will bring the user to the level one gameplay screen and play the game under default ability conditions (*see Calibrate for more information)*. Even though there are two levels, the only way to reach level two is to play and beat level one. The game does not save scores for each specific user.

Between the two levels of gameplay, the user is given 5 lives. If, in either level, the user loses all 5 lives, the user is shown a button that when clicked, brings them back to the home screen.

ii. How To

Clicking this button will bring the user to the rules and explanation pages. These explain the story, rules, and 'how to' for the game. The user can either read all the way through these 4 explanation pages and click the play button on the last page to begin playing level one, or at any time, they can navigate backwards to the home page once again.

iii. Calibrate

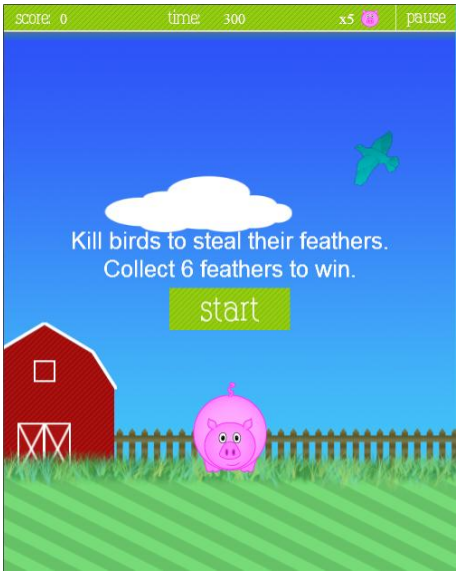
Clicking this button will bring the user to the calibration page. If the user wants to play to the best of their personal ability, they can calibrate the sensor to their skill level. The page prompts the user to squeeze the pressure sensor for 3 continuous seconds in order to determine their personal strength. From there, the application will calculate the average amount of pressure received by the pressure sensor in those 3 seconds. That value is stored internally and used to adjust either of the two levels' controls. The calibration value is stored until the game is closed or recalibrated.

For example, if a user with low finger strength calibrates the device before playing the game, they'll need to use less pressure to make the pig jump than someone who calibrated the pressure sensor to their stronger finger strength.

With that being said, it's recommended to recalibrate the pressure sensor or close and restart the game every time a new users comes up to play.

Also, if the user does not calibrate the device and goes immediately to gameplay from the home screen, default ability conditions will be used in the two levels of the game.

B. Level One



i. Synopsis

Level one begins with Hamlett the pig eager to escape the farm. The goal of this level is to obtain six feathers and escape the farm for good.

ii. How To

When the user clicks start, the timer begins counting down and they're able to make Hamlett jump by squeezing the pressure sensor.

Squeeze the sensor to jump and land on clouds. As Hamlett makes his way up, hit the the birds who will, in turn, transform into an obtainable feather. Catch the feather before it floats away. The goal

of level one is to obtain 6 feathers from the birds. This will be enough for Hamlett to obtain his wings and fly away from the farm permanently. Don't stress if you miss a feather. There are far more than 6 birds flying above from whom you can take their feathers. Although it is easier to keep jumping up and up, if the user puts little bits of pressure on the pressure sensor, Hamlett can inch his way to the edge of the cloud and fall back down. Keep in mind that the feathers can also land on the clouds.

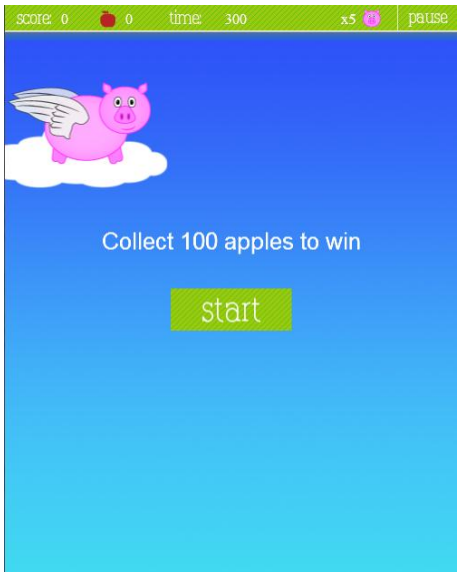
iii. LOSING THE LEVEL

Although the user hits the birds to steal and obtain their feathers, the birds can not hurt Hamlett. They can however push you off the cloud. Be careful, because if Hamlett falls more than 6 cloud platforms, the user will lose a life.

The other way to lose the level is by running out of time. There is plenty of time, but it's important to not take advantage of the abundance of time.

Every time the user loses a life, the level starts over. Lose all 5 lives and the game is over.

C. LEVEL TWO



I. SYNOPSIS

Level two begins with Hamlett the pig sitting sky high on top of a cloud. But he's hungry after his crazy adventure of leaving the farm. The goal of this level is to obtain 100 apples to feed Hamlett.

II. HOW TO

When the user clicks start, the timer begins counting down and they're able to make Hamlett flap his new wings by squeezing the pressure sensor.

Just like a young bird, Hamlett needs help flapping and learning how to use his new wings. Help Hamlett navigate the rolling hills of the countryside by squeezing the pressure sensor to make him flap his wings and gain momentum on the hills. Floating in the sky are numerous apples. By gaining momentum on the hills, Hamlett is able to reach and eat the apples. He's a hungry pig, so the level (and the game) is one when he's eaten 100 apples.

III. LOSING THE LEVEL

Just like in level one, running out of time will make the user lose a life. This time around time is more precious. Time is more numerous; Hamlett has one extra minute available to complete the level compared to level one. However, collecting 100 apples is much more time consuming. Be aware of how much time is left on the clock.

In this level, however, the birds can also attack you and cause damage. Hit three of them and the user will lose a life.

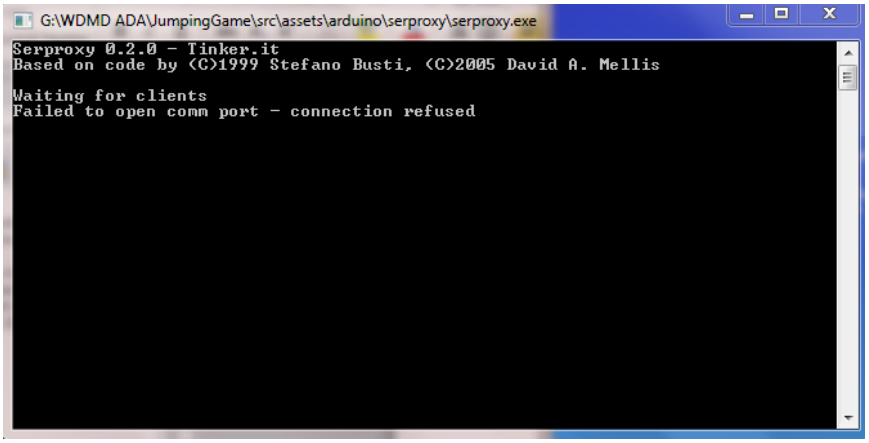
Every time the user loses a life, the level starts over. Lose all 5 lives and the game is over.

III. TROUBLESHOOTING

There are few problems forseen for functionality of this game. Please see the explanations for troubleshooting your various issues below. If these don't fix your problem, please contact Anthony Ellertson or myself (Kayla Johnson) at matsukeira@gmail.com

A. NOT CONNECTING TO THE ARDUINO PRESSURE SENSOR DEVICE

The biggest issue users would run into is not connecting to the pressure sensor. As shown on the right, the *serproxy.exe* dialog box will say "*Failed to open comm port – connection refused*" if it was not able to connect to the device.



First, close the serproxy program and then unplug and plug the pressure sensor back into the USB port of your computer.

Go back to your device manager and properties of the pressure sensor device as explained when installing and configuring the device.

Make sure the COM# port number is the same as `comm_ports=#` and `net_port#=5331` in the *serproxy.cfg* file. If they are not, modify the config file to match the corresponding COM port number the device is running on. Then save the config file and re-run *serproxy.exe*.

You may want to close out of the When Pigs Fly game before trying to play again. It's not necessary, but it may give the program a chance to refresh and use the new settings you just configured. or myself (Kayla Johnson) at kjohn983@uwsp.edu

B. SHAKING PIG

If you're playing level one and the pig seems to be shaking or vibrating in place even when the sensor is not being squeezed, the cause is the sqeeze ball placed on the sensor. The weight of the squeeze ball is adding a small amount of extra pressure to the pressure sensor device. The sensor is very sensative to slight differences in pressure, so even though the squeeze ball adds a miniscule amount of pressure, the device is still able to read and sense that.

You can try moving the sensor around in the squeeze ball to see if it releaves some pressure from being in a different position or you can remove the squeeze ball completely and just pinch the pressure sensor to play.