**[Magic-Sand-with-Games-v1.5.4.1](https://github.com/thomwolf/Magic-Sand/releases/tag/v1.5.4.1)**

@RasmusRPaulsen[**RasmusRPaulsen**](https://github.com/RasmusRPaulsen) released this 19 days ago

**Release v1.5.4.1 of the Magic-Sand with games.**

Based on the source in the branch called "Magic-Sand-with-Games"

(NC notes indented)

* Hardware
  + This runs on V1 of the Kinect for 360. Model numbers 1414, 1473 and maybe 1517 (Kinect for Windows) but I have not tried that one. I tried the other 1.

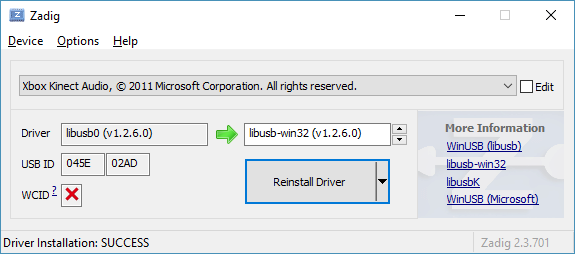
**Windows 10**  
You need to download

* the [Visual C++ run-time components](https://www.microsoft.com/en-us/download/details.aspx?id=48145) ,
  + NC - Download Visual C++ the 2013 release for x86 and x64 bit from the microsoft website: <https://www.microsoft.com/en-us/download/details.aspx?id=40784>
  + You do not need the “arm version.
  + Earlier and later releases do not have the right “vcomp” file.
  + Without this the application complains it does not have “vcomp120.dll”
* the Free USB Driver tool: [Zadig](http://zadig.akeo.ie/) and
  + NC – This is OK
* the Magic Sand software it-self
  + NC - This is OK

**Quick start**

Download and unzip the files wherever you want.

**Windows**

1. Update your video drivers.
   * NC - Use Device Manager
2. Install the Visual C++ run-time components (in particular the x86 component even if you run on a 64bits Windows).
   * NC - As mentioned, 2013 version
3. Install the Kinect drivers with the Free USB Driver tool Zadig:
   * Select your Xbox Camera, Xbox Audio and Xbox Motor in the drop down and then select the libusb-win32( v1.2.6.0) driver from the driver menu (click on the up/down arrows) and click install.
     + NC -Zadig screen
   *   
     You need to do this operation **two to three times**: once for the camera, once for motor and once for the audio.
     + NC - We did not see the motor but had a security item instead so we installed the same driver for that.
   * See the [freenect github](https://github.com/OpenKinect/libfreenect#windows) and [ofxKinect github](https://github.com/openframeworks/openFrameworks/tree/master/addons/ofxKinect#windows) for more information.  
     If there is a problem with the detection of the Kinect, you can install the [Kinect V1 (SDK 1.0)](https://www.microsoft.com/en-us/download/details.aspx?id=28782) and run Zadig again.
     + NC - We installed the Kinect V1 SDK because we did not see the Motor listed as USB device in Zadig or on Device Manager (even in “Other”). We only saw Kinect Audio and Video and no Motor but a Security in addition. So we ran Zadig and installed the USB driver ***lbusb-win32(v1.2.6.0)*** on all 3 devices we saw.
   * The drivers will identify different deices in the Kinect depending on model 1414 or 1473. That does not matter, all that matters is that you assign all of the Kinect deices to the libusb-win32( v1.2.6.0) driver.
   * Kinect model 1517 I have not verified.

**Setting up the system**

Connect and turn on the projector and the Kinect and start the Magic-Sand software.

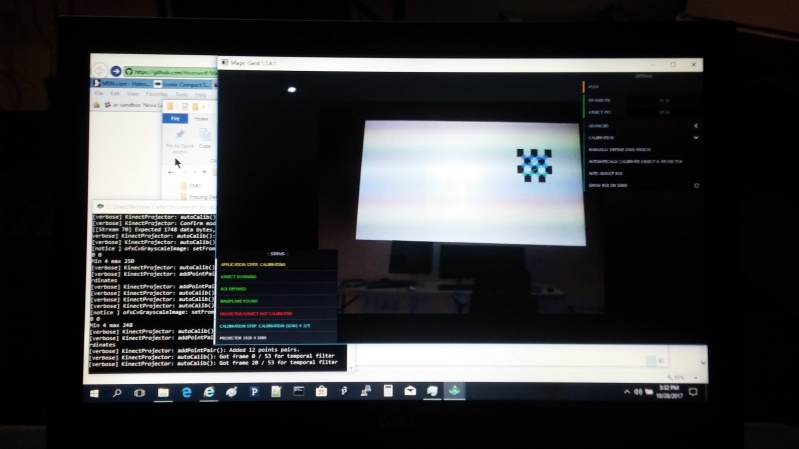
By default the software starts in a **setup** mode where the depth or color image from the Kinect can be seen in the user interface and the projector projects a completely white image. This way it is easy to check if the Kinect is running (on Windows 10 machines it can be necessary to plug and unplug the Kinect several times before it starts) and if the projector is working. The status of the Kinect and the projector can be seen in the status window to the lower left in the user interface.

In **setup** mode the physical positions of the Kinect and projector can be optimized.

* NC - After getting past the missing vcomp driver (this took a long time to figure out) we saw the projection go white and had trouble seeing anything else until we configured the PC display with a monitor second screen extension feeding the projector. At Bryce’s suggestion we configured the Projector screen above the PC screen. Then the Sandbox output went on the wall (projector) and we could see the program other windows on the PC screen.
* NC - There is a screen where you see the Kinect Camera view and have the program controls and also a command line window that shows what is happening. This was useful to see errors and progress.
* NC - Physical Positions. After some experimenting with trying to get it to calibrate, we figured out that we needed to get the camera to be a square onto the projected image as possible so we put the Kinect on a tall PC on a table

**Calibration**

To calibrate the system so the Kinect and the projector is in correspondence a few steps are needed:

* Flatten the sand in the sand box.
  + NC - The projector wall image was flat
* Make sure that you see either the depth image or the color image from the Kinect (click **advanced|Display Kinect Depth View**)
  + NC - This we could see on the PC window
* Press **Calibration|Manually Draw Sand Region**
* Define the sand region by drawing a rectangle with the mouse on the Kinect Depth or Color view
  + NC - We tried smaller areas to try to compensate for PC low processing, but determined after seeing the calibration process that we needed as much of the projected area as possible to be enclosed, as it uses the whole area for calibration.
* Press **Automatically Calibrate Kinect & Projector** - a series of Chessboard patterns are now projected on the sand.
  + NC - It will project a maybe 5x5 checkerboard on various places on the projected area then monitor it and subsequently show it on the PC Kinect camera window before displaying the next one with the projector. It does this a lot and takes a while. You can see the successful operations complete on the command line window.
  + 
* When a prompt appears cover the sand box with a light piece of cardboard or similar.
  + NC - What it is doing is finding the range to the top of the sandbox, assuming that the sand level will not exceed this. Based on this range and the level sand range determined during prior calibration activity, it can calculate the height allocation for the various sand height levels it will sense and subsequently illuminate with different colors.
  + NC - To give some effect of sandbox spacing between top of sandbox and sand level inside, we moved the Kinect camera about 4 inches closer to the screen before pressing OK.
* Press ok - a series of Chessboard patterns are now projected on the cardboard/plate.
  + NC - It did this OK
  + NC - We now had to move the camera back to the prior position.

If the calibration is successful the status window should be updated showing that all is ok.

* NC - It was.

**Debug mode for calibration**

If the calibration was not successful a debug mode can be enabled that will place debug files in the **data\DebugFiles** folder. These might point you in the direction of why the calibration failed. Do this by enabling **advanced|Dump Debug** and run the calibration routine again.

**Tips for a nice calibration:**

* Turn off the light in the room for the calibration, depending on the lighting, it may take several tries to get it to complete the calibration sometimes.
* Make sure you have a perfectly flat cover for the second part of the calibration. If it’s not flat, it may not calibrate properly. Also, make sure the cover has a matte surface otherwise the Kinect may have problem to detect the depth.

**Starting the Application**

If the calibration was successful or if a calibration was done before, the application can be started by pressing space or pushing the **Run** button.

Now a colored map with iso-lines should appear on the sand. The framerate should be close to 60 FPS for modern PCs.

* + NC - My PC showed about 10-15 FPS – lol.
  + NC - With this calibration, moving the camera closer to the wall will raise the effective sand height and change the displayed colors. By moving it in and out and tilting it we could induce display of high ground and water.
  + NC - By introducing white object like paper held above the screen image, we could show effect of making hills.
  + NC - The Projected image would show the new sand height colors after about 1 second as this is a low performance PC.
  + NC - Ideally a delay of less than 1/10 second is needed for good user interface response.
  + NC – Re-imaging the loaner LINUX laptop with Windows and using that would result in a much faster update rate and smoother response.

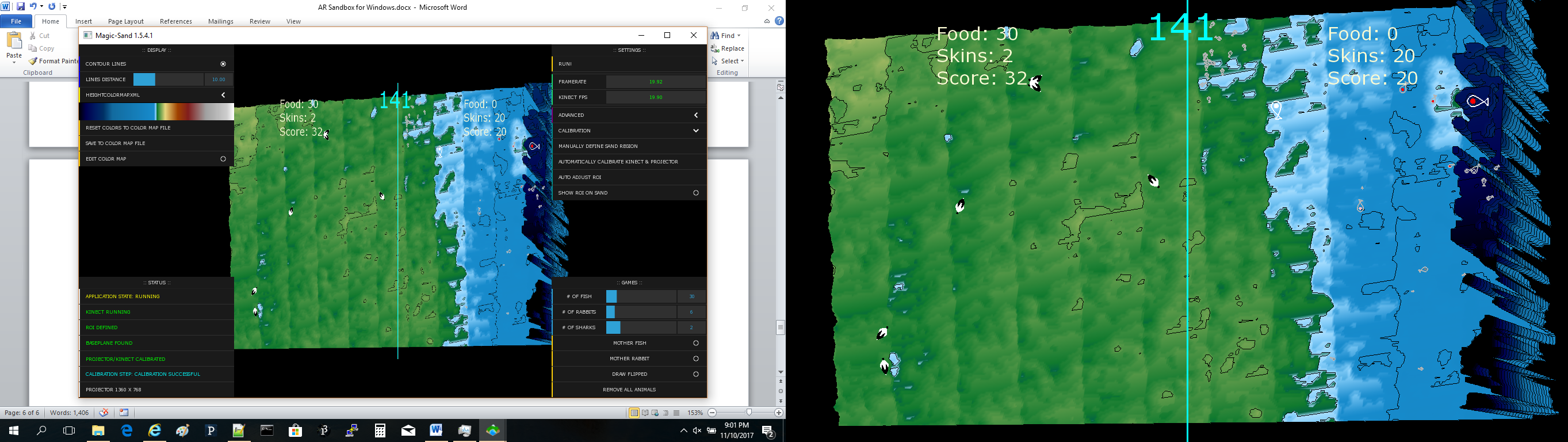


Update 11/10/2017

I have the cheap projector and the NL Kinect and I can make rthe sandbox on my wall quite bright enough in artificial light. Wall is beige.

Image is 34x20.5 at distance of 4 feet from projector lens. Ear.ier using the PC screen image not SB image I got closer for same size image. Probably I am not using full view field or camera needs moving back. It is 6 inches towards screen from projector.

Fish Game starts by pushing “f” key.



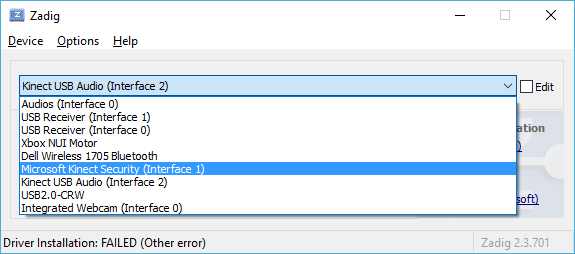
Projector is 8x6x3 inches. Line cord is fairly short.

With HW version 1414 Kinect – shows the 3 devices. Motor, Audio and Security

Motor I could install new driver. Security and Audio failed do not see Kinect video.

Devices seen keep scanning while Kinect it plugged in and a variable number show up.

Same thing happens with the Nova Labs Kinect too this is HW version 1473. And Sandbox program cannot find them.



<https://answers.microsoft.com/en-us/surface/forum/surfaccesso/kinect-v2-does-not-work-with-surface-book-with/fb4b3f74-e494-46f2-8144-23f2ec88a844?status=AllReplies&auth=1>

<https://support.microsoft.com/en-us/help/4032123/kinect-sensor-is-not-recognized-on-a-surface-book>

Cause

This issue occurs because of a change in the Surface USB Hub Firmware Update driver (SurfaceUsbHubFwUpdate.sys).

Workaround

To work around this issue, delete the **Value data** contents of the **LowerFilters** string in the following registry subkey:

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{36fc9e60-c465-11cf-8056-444553540000}**

**Note** Delete only the Value data and keep the **LowerFilters** string

Far AS I CAN GET IS VERSION 1709 BUILD 16299.125

Updated on reboot back to VERSION 1709 BUILD 16299.192

There are no FILTER parameters in the USB.

Updated all 3542 drivers that seemed applicable for Win 7.

Now Kinect is OK by itself but when plug in the Projector HDMI, the kintect keeps disconnecting.

Options –

Projector has USB input

No, that uses files only like from USB drive.

Use USB hub for Kinect input

YES this works on Win7 and also on Win 10 Creator :-D

Use HDMI🡨🡪 USB converter

Did not need to try (and this Was $40).