Run Batocera in a Virtual Machine

Running Batocera in a virtual machine is not officially supported, nor will it be an accurate representation of how it will behave, but it can be useful for quickly testing custom images or UI changes.

Except massive performance deficits compared to bare-metal installation. Virtual Machines don't take kindly to newer/modern graphics APIs and standards.

Oracle VirtualBox

1. Download the latest Batocera x86_64 image.
2. Install Oracle's VirtualBox.
3. Install the VirtualBox #.#.# Oracle VM VirtualBox Extension Pack (a.k.a. “Guest Additions”) too.

   If you need further help with installing VirtualBox, refer to their helpful user manual

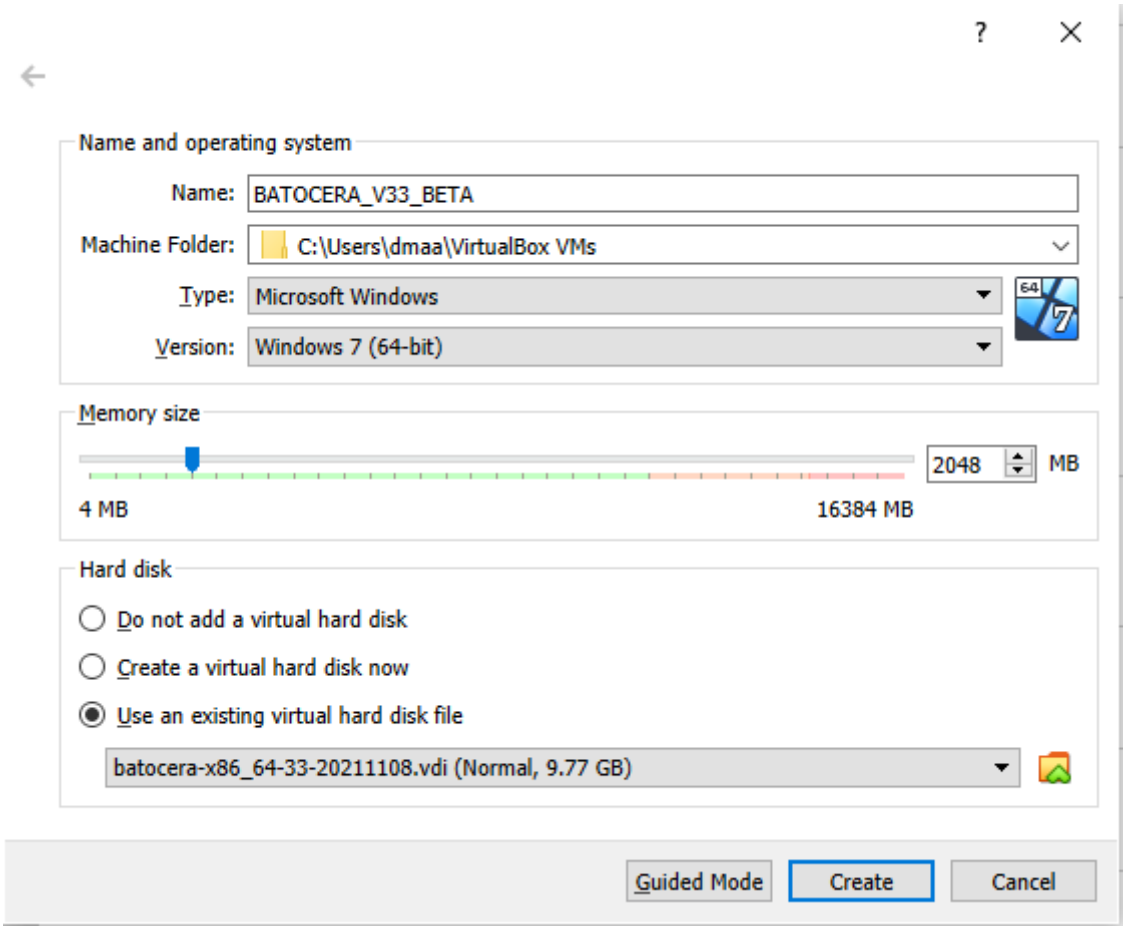
4. Convert the Batocera IMG file to a VDI using the command line (`[Win] + [R]`, then type `cmd`). For example, for the default VirtualBox installation using Batocera v33 beta:

   ```
   "C:\Program Files\Oracle\VirtualBox\VBoxManage.exe" convertdd
   C:\batocera-x86_64-33-20211108.img C:\batocera-x86_64-33-20211108.vdi
   ```

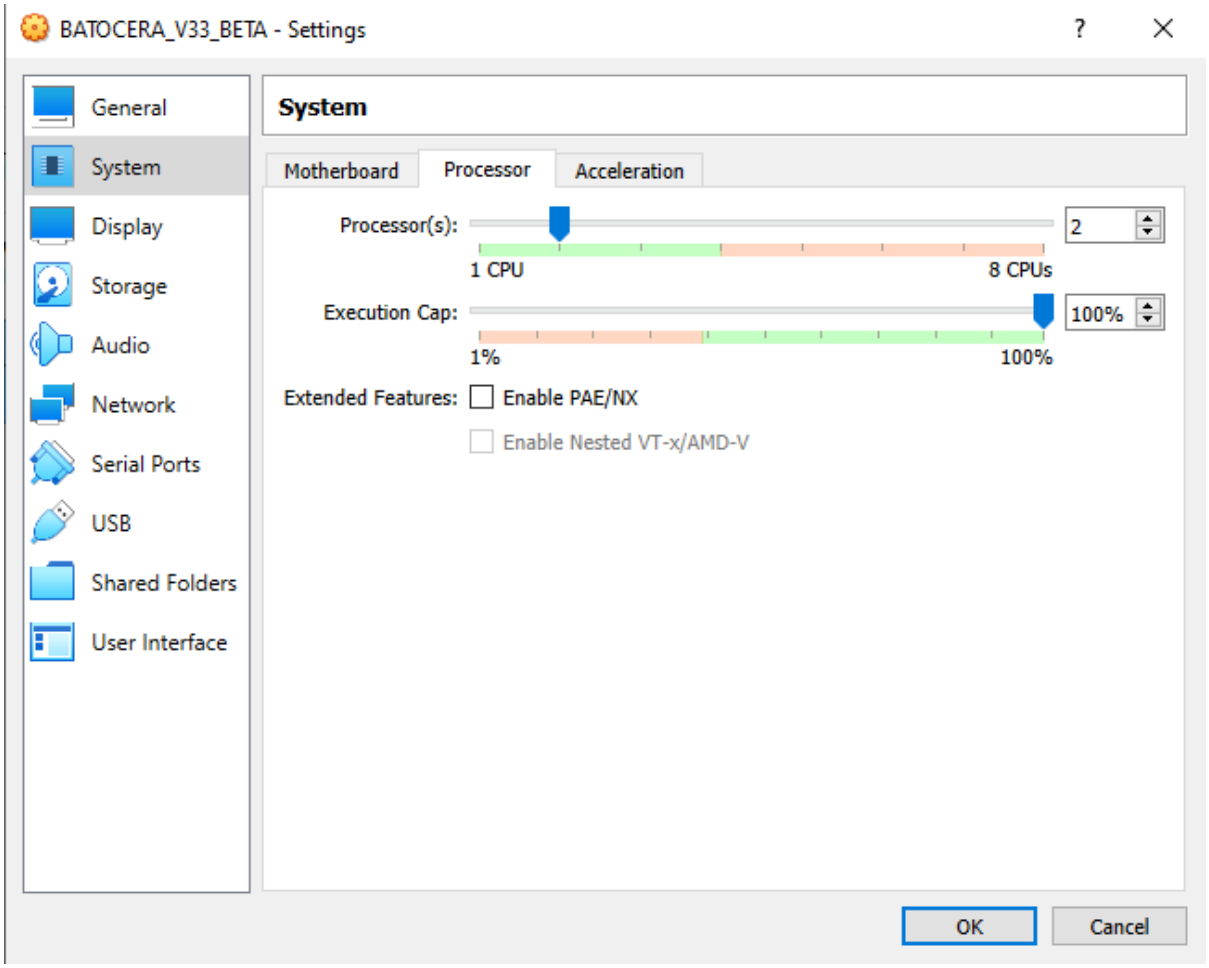
5. Resize it to be larger, can also be done via command line. For example, to create an image that's 20GB in physical size:

   ```
   "C:\Program Files\Oracle\VirtualBox\VBoxManage.exe" modifyhd
   "C:\batocera-x86_64-33-20211108.vdi” --resize 20000
   ```

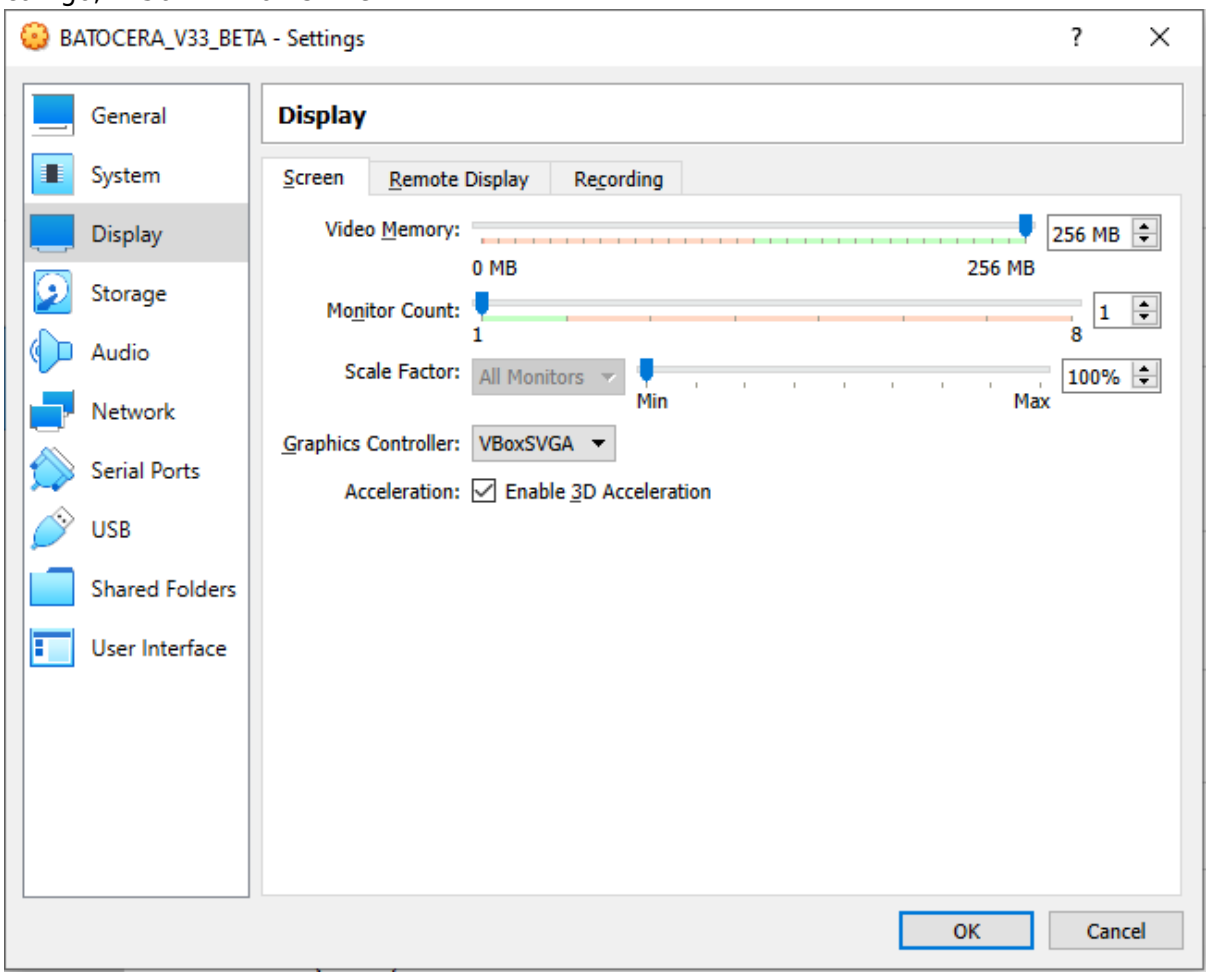
6. Run VirtualBox proper and create a new VM and select “Expert Mode”.
   1. Set the type “Microsoft Windows” and version to “Windows 7 (64bits)”; (It's just a label for predefined setup, maybe linux or oracle will be better for you, but try this one first)
   2. Set available RAM to at least 2GB (2048MB);
   3. Use the Batocera VDI file created earlier:



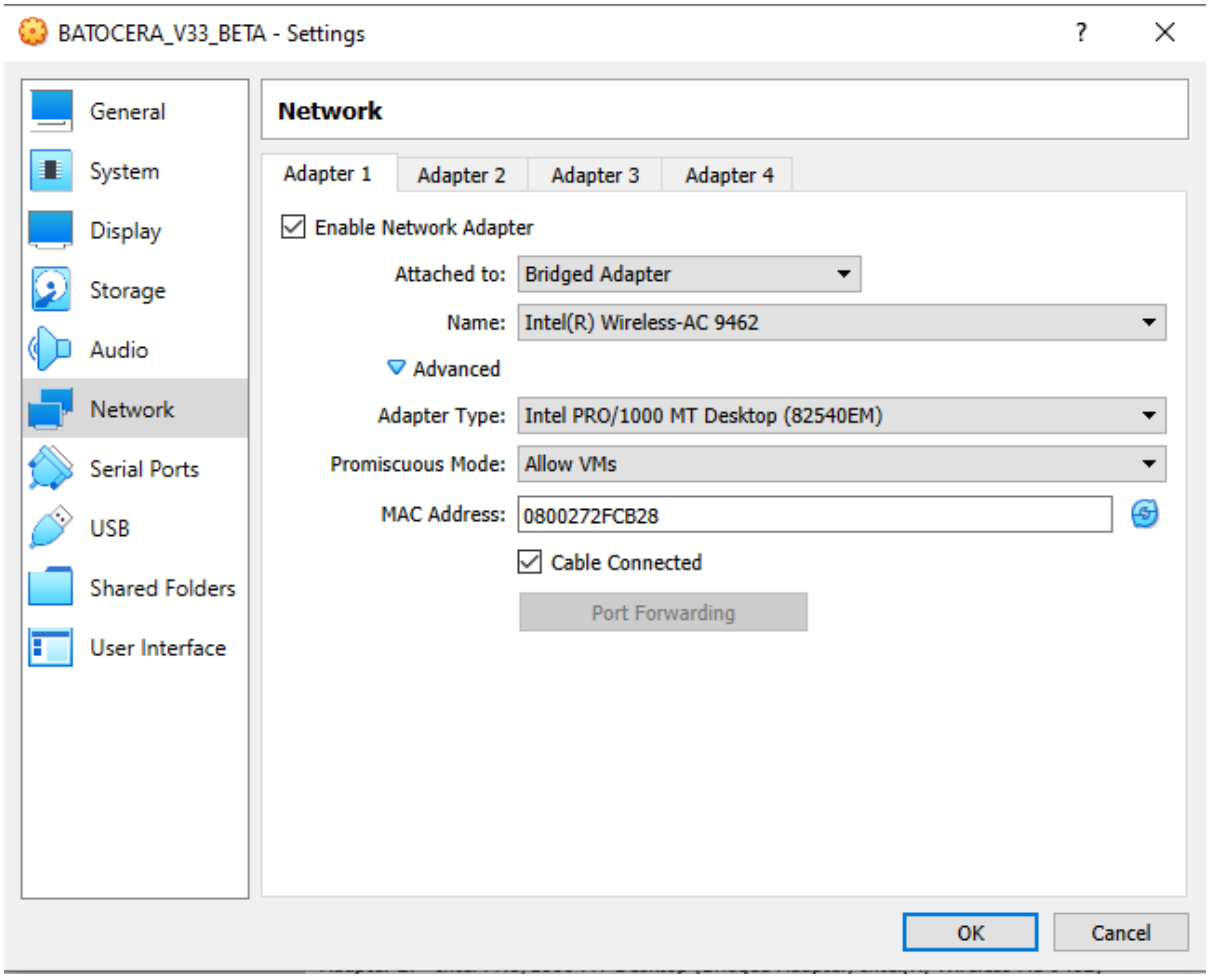
7. Once the VM has been created, adjust its settings like so:
    1. Processor: **Processors:** “2”. A higher setting might provide better performance, but try to stay within the green range for your computer. For host computers that only have two cores, use “1” here. **Execution Cap:** *100%*.



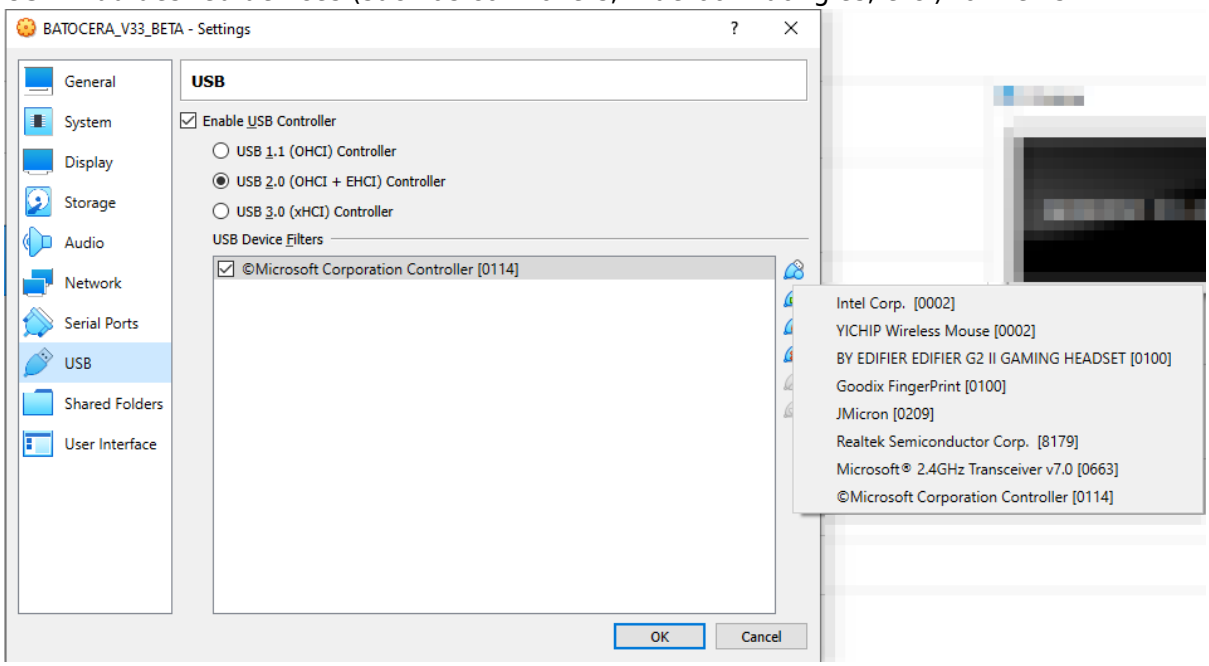
2. Display: **Acceleration:** Tick “Enable 3D Acceleration”. **Video Memory:** As high as you can go, “256MB” works fine.



3. Network: **Adapter 02:** Tick “Enable Network”. **Attached to:** “Bridge Adapter”. **Promiscuous mode:** “Allow VMs”



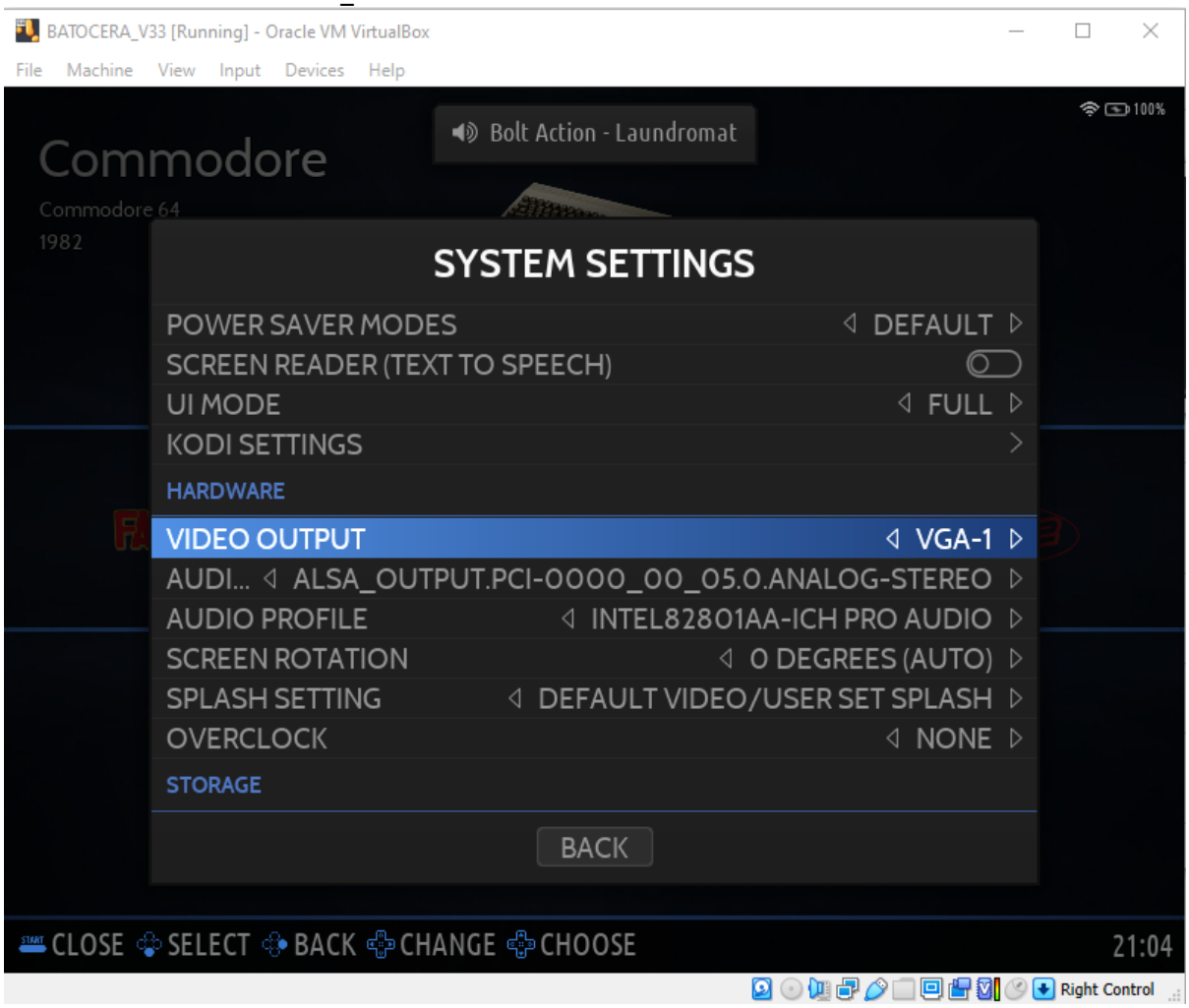
4. USB: Add desired devices (such as controllers, Bluetooth dongles, etc.) to the list:



Peripherals that the VM automatically forwards through such as mouse, keyboard and network cards are not necessary to add here.

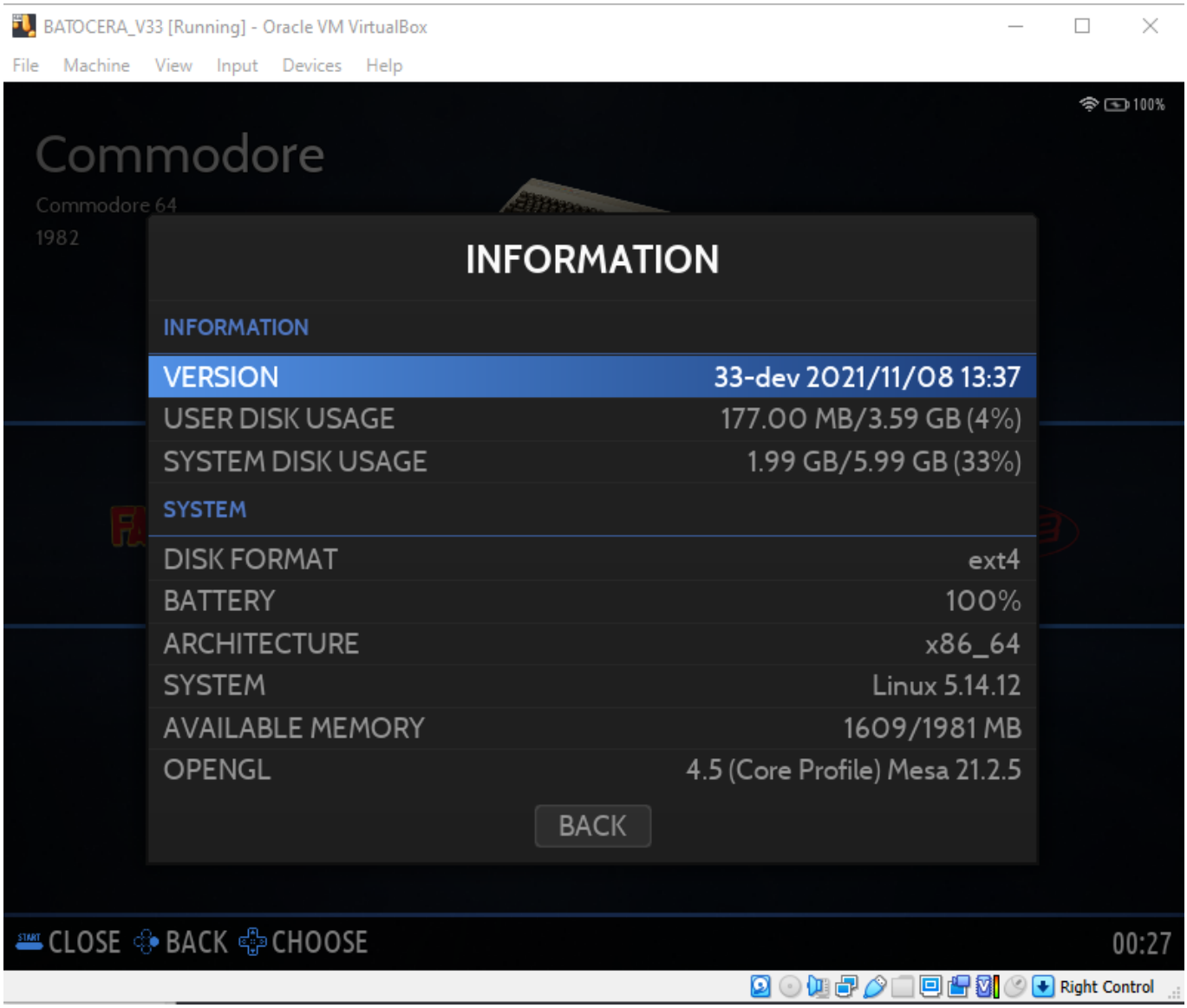
8. If AUTO Audio and AUTO Video didn't work, change the following settings within Batocera:
    1. **SYSTEM SETTINGS → VIDEO OUTPUT** to “VIRTUAL-1”,
    2. **AUDIO OUTPUT** to “ALSA_OUTPUT_PCI.STEREO” and

3. **AUDIO PROFILE** to “PRO_AUDIO”.



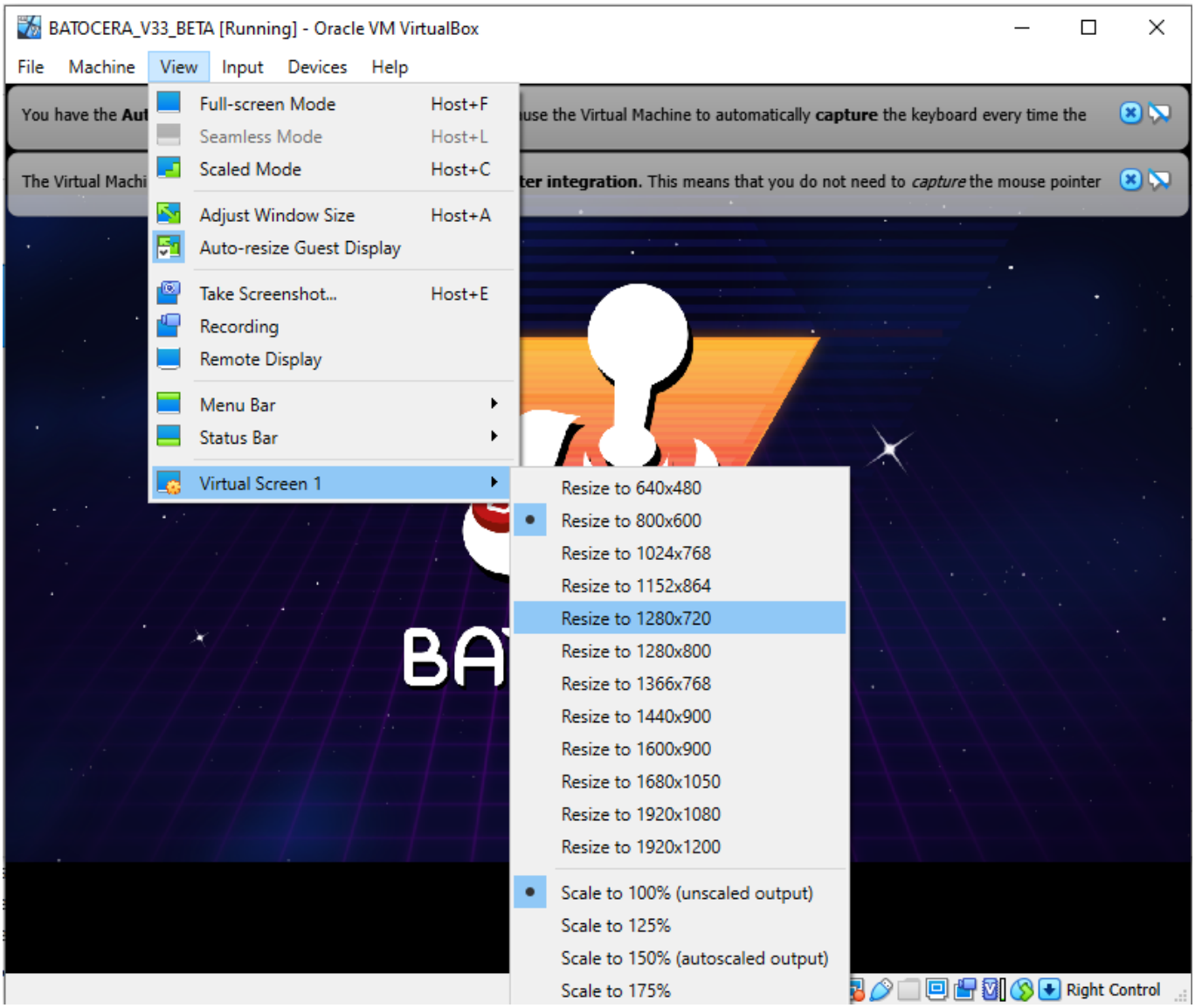
If those settings don't produce audio, keep trying all permutations of **AUDIO PROFILE** and **AUDIO OUTPUT** until you find one that does. Remember to exit the menu each time to test the change!

And we're done.



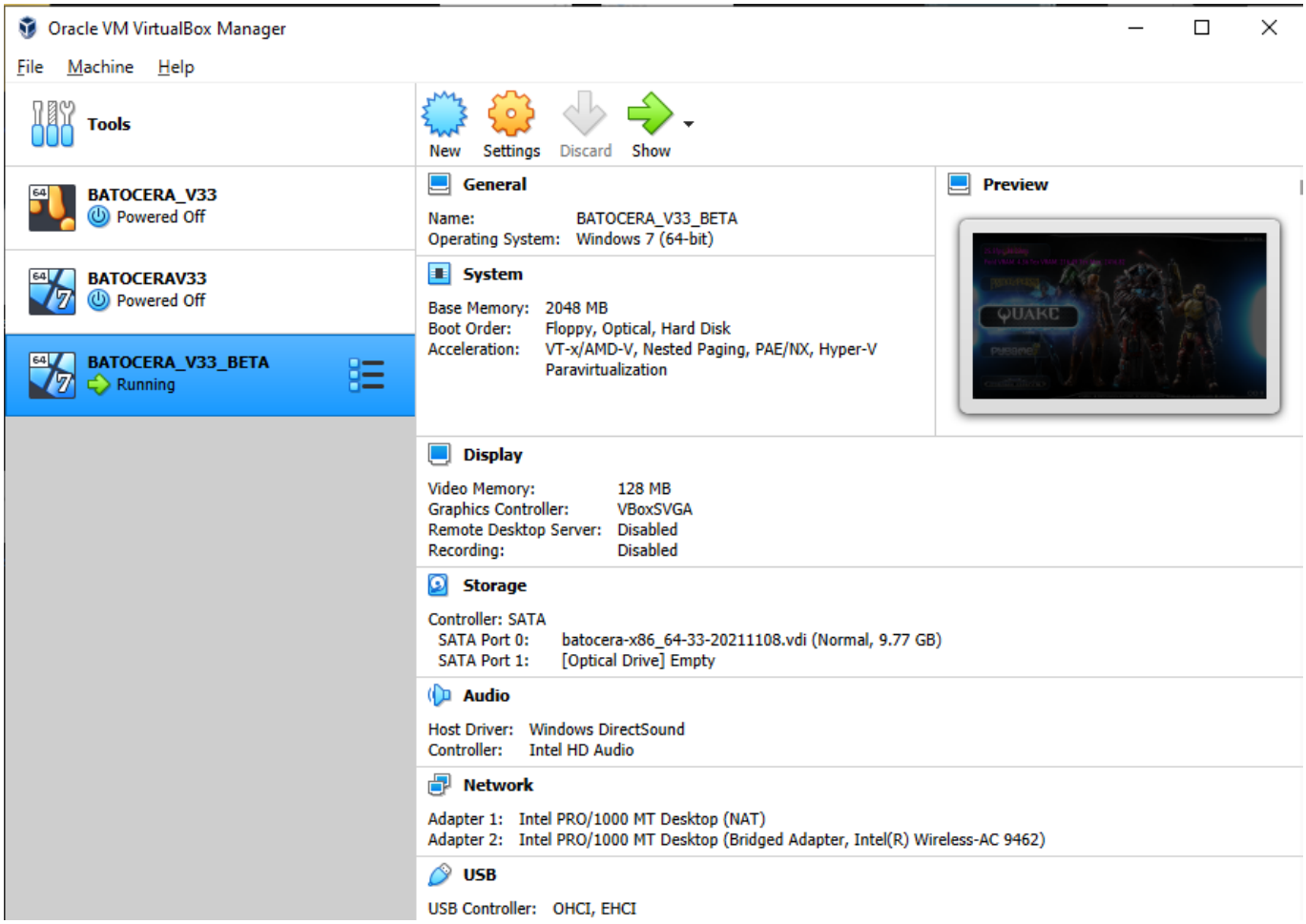
Post-installation setup

You can increase the output resolution releasing your cursor (right `[Ctrl]`) and clicking **VIEW** → **VIRTUAL SCREEN-1** in the menu toolbar of VirtualBox.



You can still add files to Batocera using the usual methods, even the network share.

As you already have the image .vdi, you can setup different environments for the same machine and test which one will be better for your PC, just make sure you are using a 64-bit one (or 32-bit if appropriate).



Qemu

Be aware, last time this was tested in **v30** there were some issues with emulation speed being uncapped. This may have been fixed since then.

1. Download the image for Batocera from the main site.
2. Download the following script and mark it as executable: https://raw.githubusercontent.com/batocera-linux/batocera.linux/master/scripts/linux/runQemu.sh
3. Create an empty “share” image containing the partition Batocera will use for its userdata and name it `share.img` (replacing 5G with the size you desire, make it at least 5GB to support future upgrades):

```
dd if=/dev/zero of=share.img count=5 bs=5G
```

   To confirm that the blank share image has been created, run

```
ls -lh share.img
```

   to check its reported space. For instance, a 5GB blank image should report back something like:

```
-rw-rw-r-- 1 XXX XXX 5.0G XXX share.img
```

1. Optionally, if you don't want to use Batocera itself to format the partition, create the filesystem manually with something such as:

   ```
   mkfs.ext4 share.img
   ```

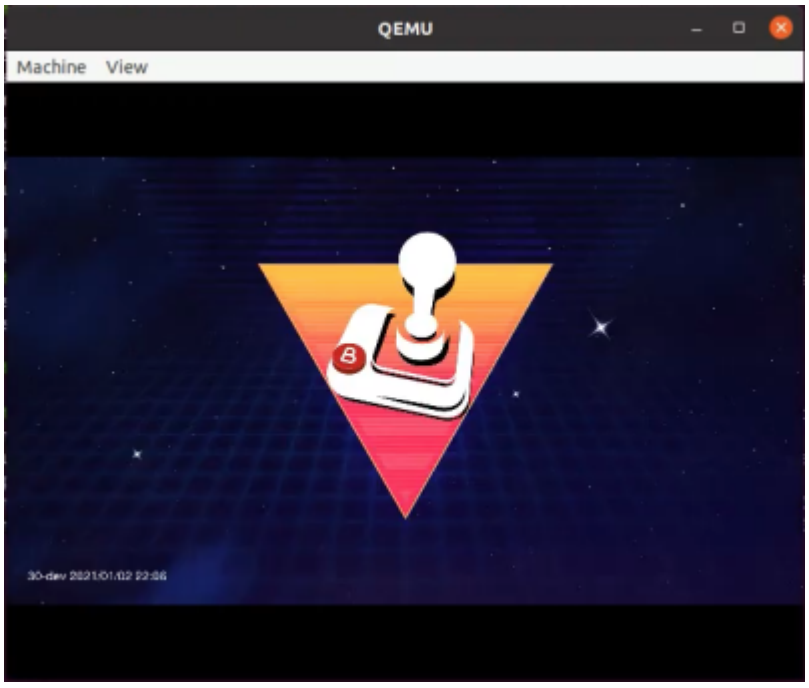
   Otherwise it's fine to, after booting Batocera, go to **SYSTEM SETTINGS** → **DEVELOPER** → **FORMAT A DISK** and format it there.

2. Optionally, forward a gamepad connected to the host computer by running the following:

   ```
   udevadm info -q all -n /dev/bus/usb/001/002
   echo 'SUBSYSTEM=="usb", ENV{ID_MODEL}=="Usb_Gamepad", MODE="0666"'
   >> /etc/udev/rules.d/99-joysticks-rw.rules
   ```

4. Then run the following to start up the Qemu virtual machine:

   ```
   ./runQemu.sh ~/batocera-x86_64-XX-XXXXXXXX.img share.img
   ```



The `runQemu.sh` script itself contains further instructions (that may get updated faster than this wiki). For an example of what an ordinary installation procedure might look like:

```
nicolas@setsouko:/media/ssd/dev/batocera.linux/scripts/linux$ ./runQemu.sh
./runQemu.sh <batocera.img>
./runQemu.sh <batocera.img> <share image>
You can create a share image with : # dd if=/dev/zero of=share.img count=5 bs=1G
You can format it from batocera
Then you can mount it on linux with :
X=$(sudo losetup -f)
mkdir -p BATOCERA
sudo losetup -P $X share.img
sudo mount ${X}p1 BATOCERA
sudo umount BATOCERA
sudo losetup -D $X
nicolas@setsouko:/media/ssd/dev/batocera.linux/scripts/linux$ dd if=/dev/zero of=share.img count=5 bs=1G
5+0 enregistrements lus
5+0 enregistrements écrits
5368709120 octets (5,4 GB, 5,0 GiB) copiés, 10,5023 s, 511 MB/s
nicolas@setsouko:/media/ssd/dev/batocera.linux/scripts/linux$ ls -lh share.img
-rw-rw-r-- 1 nicolas nicolas 5,0G janv.  3 20:54 share.img
nicolas@setsouko:/media/ssd/dev/batocera.linux/scripts/linux$ ./runQemu.sh ~/dev/tests/batocera-x86_64-30-20210103.img share.img
```

A video demonstration of Batocera being run inside a Qemu VM.

Post-installation setup

You can still add files to Batocera using the usual methods, even the network share.

In case you'd like to mount the share image to add files directly, run the following:

```
X=$(sudo losetup -f)
mkdir -p BATOCERA
sudo losetup -P $X share.img
sudo mount ${X}p1 BATOCERA
sudo chmod -R 777 BATOCERA/
```

Then it should be automatically detected and mounted by your desktop environment.

Proxmox

Untested. Reportedly, following this tutorial to set it up in addition to passing through all devices enables it to work:
https://www.nicksherlock.com/2020/12/running-tails-as-a-vm-with-persistence-on-proxmox/

Troubleshooting

My controller isn't working!

First, make sure you have passed it through as a USB device to the VM itself (step 7d. in the instructions above).

If it's still not working, check that the controller itself is working fine on your host machine outside of VirtualBox.

I've done that and it's still not working! I'm on a Linux-based host.

If it's still not being recognized, add the user to the correct group:

1. Close VirtualBox
2. Open a terminal and run `sudo usermod -a -G vboxusers $USER && sg vboxusers virtualbox`

Windows shouldn't have this issue as long as the program is run in administrator mode.

From:
https://wiki.batocera.org/ - **Batocera.linux - Wiki**

Permanent link:
https://wiki.batocera.org/run_batocera_in_vm?rev=1645343531

Last update: **2022/02/20 07:52**