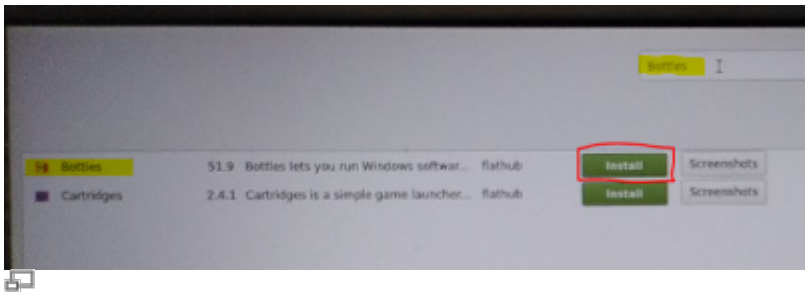
Bottles

Bottles is a virtual application/gaming platform for setting up pre-configured environments to support a large set of Windows applications/games on Linux. Be aware that Bottles is not officially supported by the Batocera developers, and no support can be given if issues are faced while using it.

Installation

On the Batocera main menu press [F1] on your keyboard, then navigate to **Applications → flatpak-config** (if a warning message appears, just confirm by clicking **OK**). Then, in the searchbar on the top of the screen, search for *Bottles* and install the *Bottles* application by clicking on **Install**:



“Bottles” installation

Confirm by clicking on **Yes** and wait for the installation process to be finished (click on **Show details** to show the installation progress, which can take a couple of minutes). As soon as the installation process has finished, click on **Close**, then click on **File → Close Window**. Now, back on the Batocera main screen, update the gamelist. The Bottles application should now be available in the Ports section. If so, you have successfully installed Bottles on your Batocera system!

Preparing the Bottles environment

To keep things abstracted it is recommended to create a new gaming system folder `/userdata/roms/bottles` to hold all the bottle data. To do so launch a SSH/Xterm session and execute the following command:

```
mkdir -p /userdata/roms/bottles
```

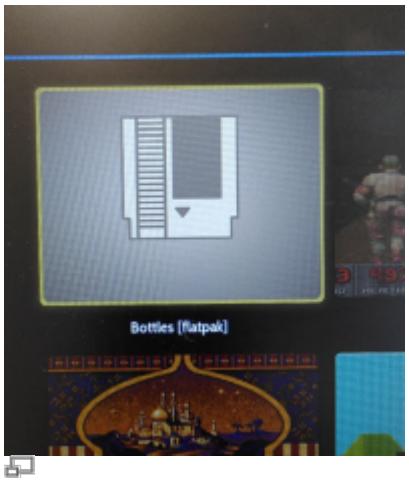
The according `/userdata/roms/bottles` folder has now being created. Grant Bottles access to the `/userdata` folder and all of its subfolders:

```
flatpak override com.usebottles.bottles --filesystem=/userdata
```

Setting up a game (example)

The following example will cover the game Twin Cobra.

Copy the game's main source data folder to `/userdata/roms/bottles`. Then start *Bottles* via the Batocera `Ports` system section:

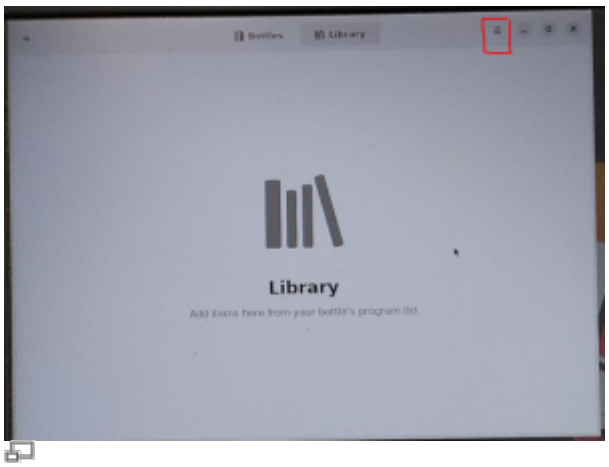


Start “Bottles”

Navigate through the initial *Welcome to Bottles* screen (don't close it!). Then, on the last screen, confirm by clicking on the **Continue** button and then on **Start using Bottles** (the initial setup can take a couple of minutes, this step only has to be done once after a fresh Bottles installation).

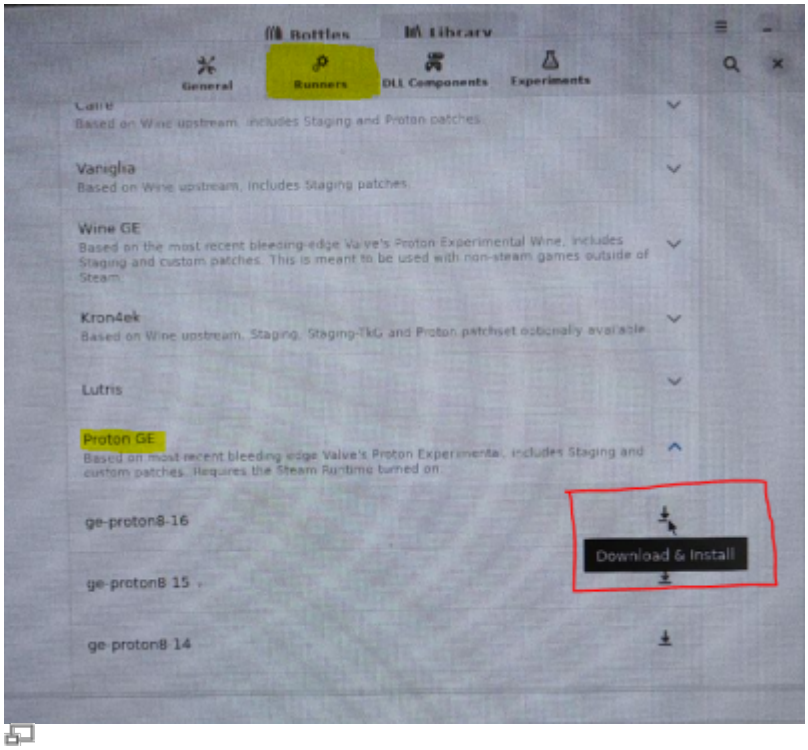
First of all, additionally to the bottle runners already installed by default, you should add the most common and most compatible one(s) as follows:

From the *Bottles* main menu click on the “Hamburger” icon:



“Hamburger” icon

Click on `Preferences` and navigate to `Runners`. From there click on `Proton GE` which opens a dropdown menu. Select the most recent one from the top of the list (currently `ge-proton8-16`) by clicking on the download icon (`Download & Install`):

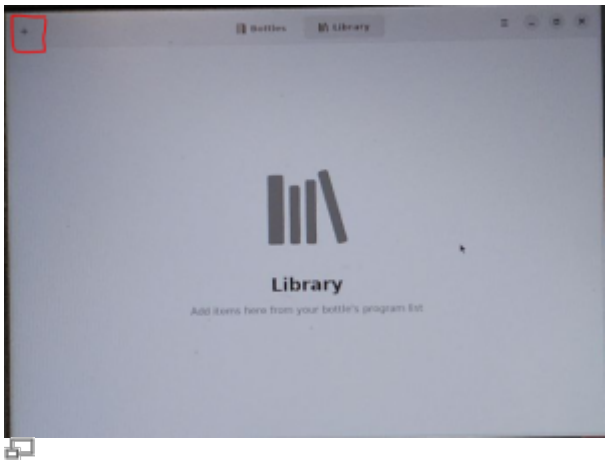


Install additional runner(s)

It can take a couple of minutes for an additional runner to install, just wait for it to finish installing.

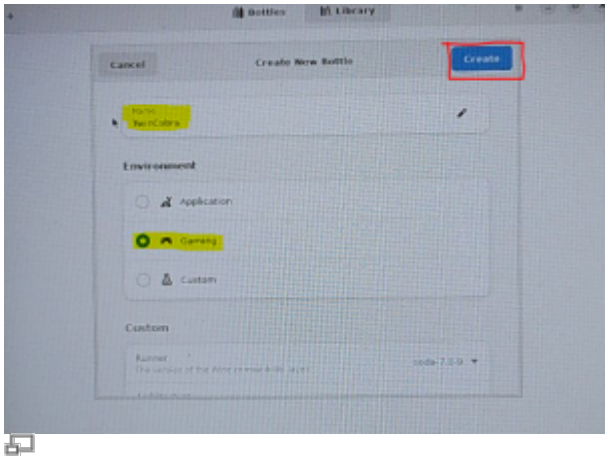
Now close the Runners window which brings you back to the main *Bottles* menu.

From there, on the left upper corner, create a new bottle by clicking on the **+** symbol:



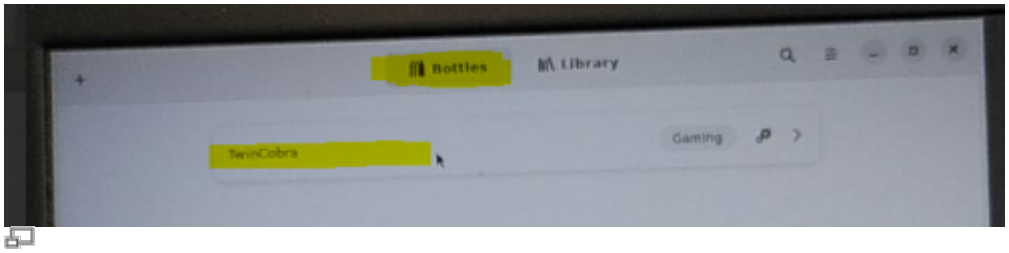
Create a new bottle

In the *Name* input field give the bottle a meaningful custom name by inserting the game's name (“TwinCobra” in this case). For the *Environment* section select the **Gaming** radio button (which will set up an optimized gaming environment bottle automatically) and then click on **Create** (the runner can be changed later):



Initial bottle settings

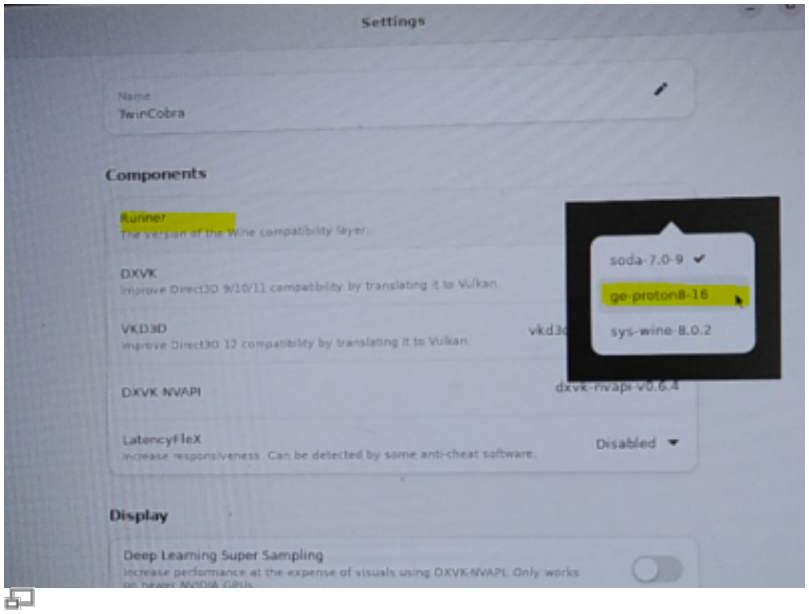
Wait for the bottle to be set up which can take a couple of minutes. Confirm the success message by clicking on **Close**. Navigate to **Bottles** on the top register card menu. From there you can see all available bottle(s) you have created, in this case the one you have just created:



Bottle

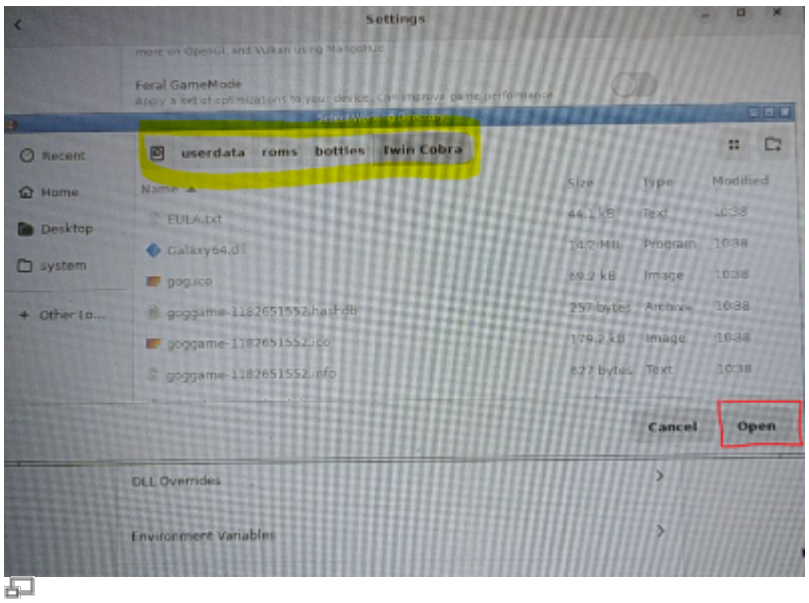
Click on the bottle, which will open up the bottle's settings menu. From there you can do a lot of stuff to customize the according bottle. We will only cover the most common settings here as every game potentially needs its own settings to run successfully and optimized. First of all let's take on look on how you can change the according bottle runner:

Click on **Runner** and from the appearing drop-down menu select the according *ge-proton* runner you have just installed previously (Sidenote: This is just an example on how to install/change a bottle runner. For the specific game *Twin Cobra* leave the runner on `soda` as `ge-proton` will currently not work for *Twin Cobra*!):



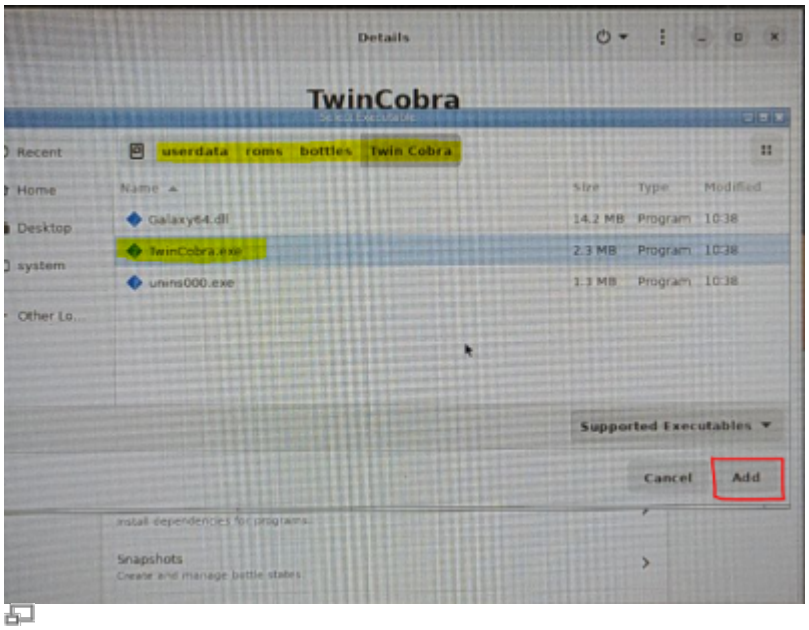
ge-proton runner

Wait for the runner to be changed which can take a couple of seconds. Scroll down to the *Compatibility* section and click on **Working Directory**. From the newly opened window click on **+ Other Locations** and navigate to the according game's main folder, then click on **Open**:



Working directory

Now scroll up to the top and on the top left corner click on the back arrow symbol (**Go Back**), which brings you back to the main bottle screen. From there click on **Add Shortcuts...**, navigate to your game's main folder where the according game's executable file (`<myGame>.exe`) is located, highlight the according game's executable file and click on **Add**:

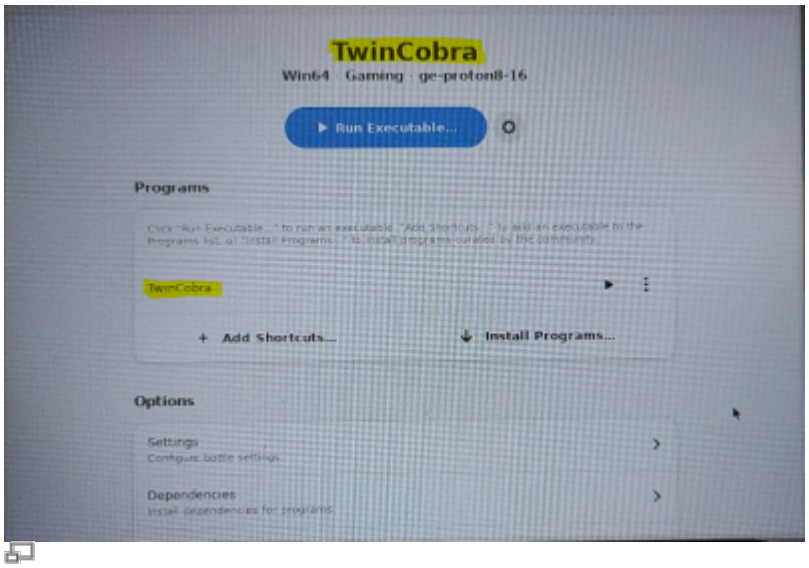


Executable (.exe) file

Now, for later automation reasons, you have to remember two things:

- There is the bottle's unique **main name** being shown on top of the main bottle menu screen.
- There is the bottle's unique **executable name** being shown under the *Programs* section (This is due to you can have multiple executables per bottle which in theory gives you the possibility to install multiple games within a single bottle, but for abstraction and compatibility/dependency

reasons it is strongly recommended to set up an unique bottle per game).



Bottle's name and game's executable name

Keeping the bottle's name and the game's executable name in mind, let's now do the automation part which lets you start the bottle's according executable with just a single click like usually in every other gaming system in Batocera.

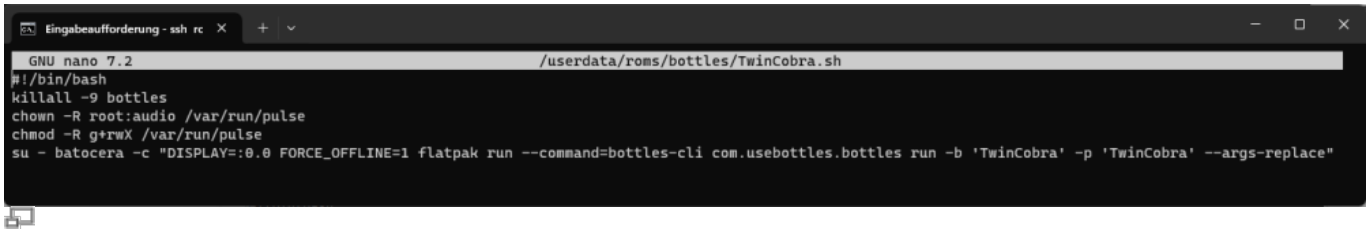
Via SSH/Xterm create an according start script file by executing the following command:

```
nano /userdata/roms/bottles/<myGame>.sh
```

Paste the following content (adjust the according `<myBottleMainName>` and `<myBottleExecutableName>` variables with the according values according to your setup:

```
#!/bin/bash
killall -9 bottles
chown -R root:audio /var/run/pulse
chmod -R g+rwX /var/run/pulse
su - batocera -c "DISPLAY=:0.0 flatpak run --command=bottles-cli
com.usebottles.bottles run -b '<myBottleMainName>' -p
'<myBottleExecutableName>' --args-replace"
```

Here's an example for the game `Twin Cobra`:



Start script example

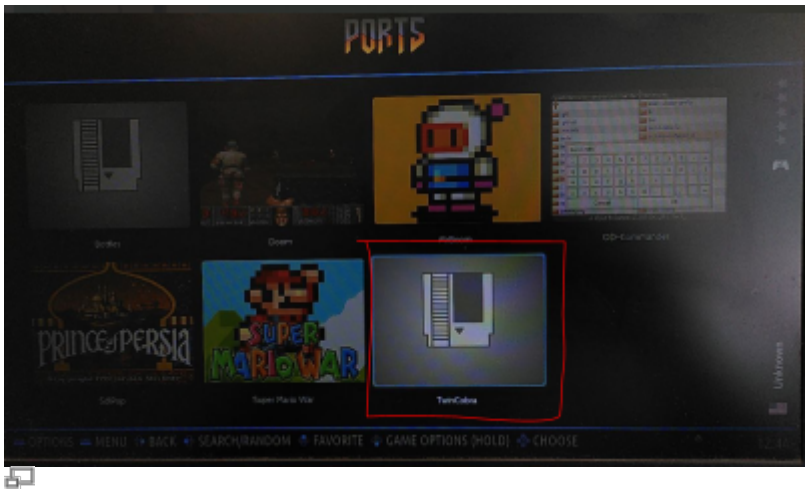
Save the file and quit the editor. Now make the according start script executable by executing the following command (replace `<myGame>` with the according game name (avoid spaces if possible)):

```
chmod +x /userdata/roms/bottles/<myGame>.sh
```

Now symlink the according start script to your `/userdata/roms/ports` folder by executing the following command (replace <myGame> with the according game name (avoid spaces if possible)):

```
ln -s /userdata/roms/bottles/<myGame>.sh /userdata/roms/ports/<myGame>.sh
```

Now, from the Batocera main screen, update the gamelist. The according game's start script you've just created should now appear in the `Ports` system section:



Ports

You can now start the according game like every other game in Batocera.

Optional: Feel free to add the according `/userdata/roms/bottles/videos` and `/userdata/roms/bottles/images` folders manually via SSH/Xterm to add the according media data per game afterwards. You can create those folders by executing the following command:

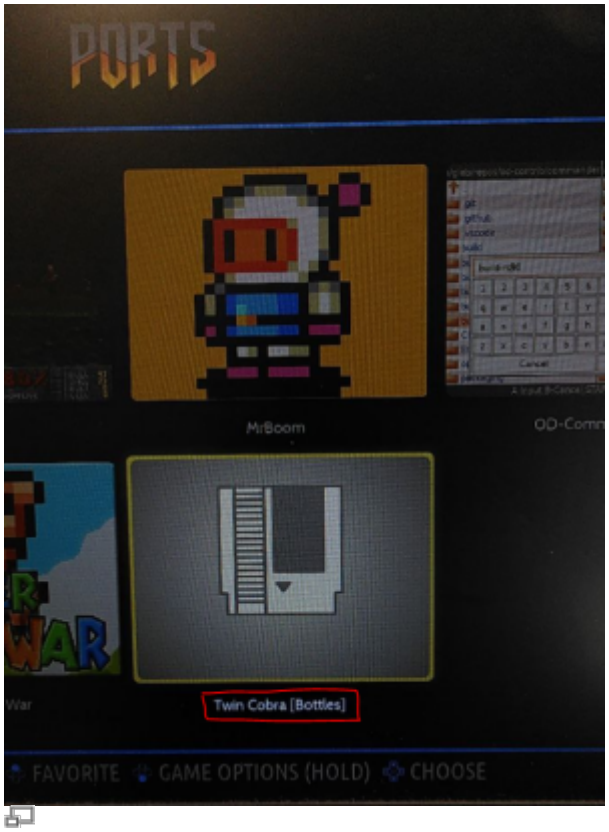
```
mkdir -p /userdata/roms/bottles/videos /userdata/roms/bottles/images
```

Optional: If for the future you want to keep in mind which games are set up by *Bottles*, you can add a `[Bottles]` tag to the according games by renaming them via ES as follows:
Highlight the according game → long-press the `[Enter]` key on the keyboard → **EDIT THIS GAME'S METADATA** → Select **Name** → Add a `[Bottles]` tag to the game's name:



“[Bottles]” tag

This way the according games will appear with an additional `[Bottles]` tag on the `Ports` section:



"[Bottles]" tag

Troubleshooting

Game not starting

As always, when a game does not start as expected, this can have multiple reasons. From experience it can be said that the most common problems for games failing to run via *Bottles* is due to missing dependencies (see below on how to install dependencies), wrong bottle runners (see tutorial step *Setting up a game (example)* above on how to install/change a bottle runner), and/or wrong CPU synchronization settings.

For changing CPU synchronization settings, open the according bottle and then navigate to:

Options → Settings → Performance → Synchronization:

Changing this setting from `Esync` (default) to `Fsync` often resolves the problem of games not starting as expected.

You can also have a look into the main Batocera logfiles which may be helpful sometimes too.

Dependencies

If you know or you want to try which additional dependencies (e.g. *Microsoft Visual C++ Redistributable* (vcrun), *.NET*, etc.) your game needs to run, you can add/remove those dependencies as follows:

Open the according bottle and then navigate to:

Options → Dependencies

From here you can add/remove any dependencies listed.

Controller issues

Open the according bottle and then navigate to:

Tools → Legacy Wine Tools → Control panel

From there check the `dinput` and/or `xinput` controllers being recognized by *Bottles*. This option is only available if the according chosen runner for your bottle supports this setting (e.g. *ge-proton*).

Map hotkeys to quit bottle games

If you want to map the hotkey-combo keys to the according game to quit the according game as for other Batocera gaming systems, you can do so by creating an according `.keys` file as follows by executing:

```
nano /userdata/roms/ports/<myGame>.sh.keys
```

Paste in the following content:

```
{
    "actions_player1": [
        {
            "trigger": [
                "hotkey",
                "start"
            ],
            "type": "exec",
            "target": [
            "killall -9 bottles"
            ]
        }
    ]
}
```

Save the file and quit the editor.

If you need the same for the *Bottles* application itself, this can be done by creating the following file with the same content as described above:

`/userdata/roms/flatpak/Bottles.flatpak.keys`

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

Permanent link:
https://wiki.batocera.org/systems:bottles?rev=1705699976

Last update: **2024/01/19 21:32**