
Retropie: support des jeux PC (x86) et des jeux Steam

Mathieu ABATI (mathieu-abati.com)

Sommaire

RetroPie: support des jeux PC (x86) et des jeux Steam	3
Ajout d'un nouveau système	3
Support des jeux 32-bits	3
Ajouter un jeu PC	4
Quitter les jeux PC	4
Support des manettes et joysticks	7
Padmapper	7
Antimicro	8
Forcer la désactivation des contrôleurs dans un jeu	11
Ajouter un jeu Steam	12
Copier le jeu Steam vers le PC RetroPie	13
Ajouter le jeu dans la collection PC d'EmulationStation	13
Dépendances	13
Avancé	13

RetroPie: support des jeux PC (x86) et des jeux Steam

Ce tutoriel explique comment accéder à ses jeux PC (x86) et Steam depuis EmulationStation. Cette méthode ne s'applique qu'à une installation PC de RetroPie, et est incompatible avec une installation RaspberryPi.

Ce tutoriel est lié au tutoriel [Installation de RetroPie sur PC](#).

Ajout d'un nouveau système

Tout d'abord, on copie la configuration par défaut des systèmes supportés:

```
1 cp /etc/emulationstation/es_systems.cfg /opt/retropie/configs/all/  
      emulationstation
```

ainsi, nos modifications ne seront pas altérées en cas de mise à jour de RetroPie.

Maintenant, on peut éditer le fichier **/opt/retropie/configs/all/emulationstation/es_systems.cfg** et y ajouter à la fin, avant </systemList>:

```
1 <system>  
2   <name>pc</name>  
3   <fullname>PC</fullname>  
4   <path>/home/pi/RetroPie/roms/pc</path>  
5   <extension>.sh</extension>  
6   <command>%ROM%</command>  
7   <platform>pc</platform>  
8   <theme>pc</theme>  
9 </system>
```

Support des jeux 32-bits

Suivez cette section si votre installation de Debian est 64-bits, auquel cas vous aurez besoin d'avoir effectué les opérations qui suivent pour lancer des jeux 32-bits.

Vous avez besoin d'effectuer cette procédure si l'exécution de la commande:

```
1 ldd executable_du_jeu
```

vous renvoie l'erreur "not a dynamic executable", et que la commande:

```
1 file executable_du_jeu
```

vous dit, entre autres, "ELF 32-bit".

Si vous êtes dans ce cas, exécutez:

```
1 sudo dpkg --add-architecture i386
2 sudo apt-get update
3 sudo apt-get install libc6-i386 lib32stdc++6
```

si vous avez une carte nVidia avec les drivers nVidia installés, vous pourriez avoir besoin de:

```
1 sudo apt install libgl1-nvidia-glx:i386
```

(un paquet similaire doit être disponible pour les autres cartes graphiques).

Ajouter un jeu PC

Pour ajouter un jeu, il suffit de créer un fichier portant le nom du jeu dans **/home/pi/RetroPie/roms/pc**, avec l'extension **.sh**. Exemple:

```
1 touch ~/RetroPie/roms/pc/un_jeu_pc.sh
```

On rend ensuite ce fichier exécutable:

```
1 chmod +x ~/RetroPie/roms/pc/un_jeu_pc.sh
```

Puis il faut éditer ce fichier pour y ajouter la commande de lancement du jeu comme suit:

```
1 #!/bin/bash
2 kill -USR1 $(cat /tmp/music_player.pid)
3 /chemin/absolu/vers/binaire/du/jeu
4 kill -USR2 $(cat /tmp/music_player.pid)
```

Si vous n'avez pas suivi le chapitre relatif à la musique de fond du tutoriel [Installation de RetroPie sur PC](#), retirez de votre script les deux lignes qui commencent par **kill**. Elles servent à interrompre la musique au lancement du jeu, et à la redémarrer lorsque le jeu est terminé.

Quitter les jeux PC

Il peut être pratique de quitter un jeu avec une touche dédiée, ou une combinaison de touches, à la manière des autres jeux émulés par RetroPie.

Dans l'exemple ci-dessous, nous avons choisi de presser deux touches simultanément sur notre contrôleur de jeu pour quitter un jeu PC.

Installons tout d'abord les paquets requis:

```
1 sudo apt install joystick xserver-xorg-input-joystick xbindkeys
```

Pour identifier les touches de notre contrôleur de jeu, nous avons besoin de **jstest**:

```
1 jstest
```

vous pourrez ainsi voir, lors de la pression sur une touche du contrôleur de jeu, son numéro. Dans notre cas d'exemple, nous utiliserons START et SELECT, dont les codes sont 9 et 8 respectivement.

De la même façon, il va falloir identifier le code des touches clavier CTRL et X. Nous utiliserons **xbindkeys** pour ça:

```
1 xbindkeys -k
```

qui nous révèle le code lors d'un appui touche. Pour CTRL nous obtenons 37, et pour X 53.

À présent, nous allons configurer Xorg de sorte que START et SELECT déclenchent la combinaison de touches CTRL+X. Pour cela, il faut éditer **/etc/X11/xorg.conf.d/50-joystick-all.conf** et y insérer:

```
1 Section "InputClass"
2     Identifier "joystick-all"
3     Driver "joystick"
4     MatchIsJoystick "on"
5     MatchDevicePath "/dev/input/event*"
6     Option "StartMouseEnabled" "false"
7     Option "MapButton9" "key=37"
8     Option "MapButton10" "key=53"
9 EndSection
```



Note: Il faut ajouter 1 au code de touche donné par **jstest**, ici nous avons donc MapButton9 et MapButton10.

À présent nous pouvons configurer tout ça. Nous utiliserons **xbindkeys** pour déclencher une action sur CTRL+X. Il faut créer le fichier **~/.xbindkeysrc** et y insérer:

```
1 "/home/pi/RetroPie/roms/pc/_exit_game.sh"
2 Control+Mod2 + x
```

Nous créons donc également **~/RetroPie/roms/pc/_exit_game.sh** avec ce contenu:

```
1 #!/bin/bash
2
3 PIDFILE="/tmp/pc_game.pid"
4
5 # Get all child processes PID
6 cpid ()
```

```

7      {
8          local child=$(ps -o pid= --ppid "$1")
9          for pid in $child; do
10              cpid "$pid"
11      done
12      echo "$child"
13  }
14
15  if [ -f ${PIDFILE} ]; then
16      PID=$(cat ${PIDFILE})
17      sudo kill $(cpid $PID)
18      sudo kill $PID
19      rm ${PIDFILE}
20  fi

```

et par la même occasion, nous créons, ~/RetroPie/roms/pc/_run_game.sh:

```

1 #!/bin/bash
2
3 echo $BASHPID > /tmp/pc_game.pid
4 "$@"

```

puis nous rendons ces deux script exécutables:

```

1 chmod +x ~/RetroPie/roms/pc/_run_game.sh
2 chmod +x ~/RetroPie/roms/pc/_exit_game.sh

```

Pour lancer un jeu PC, ça se passe légèrement différemment de ce que nous avons vu précédemment:

```

1 #!/bin/bash
2 kill -USR1 $(cat /tmp/music_player.pid)
3 ~/RetroPie/roms/pc/_run_game.sh /chemin/absolu/vers/binaire/du/jeu
4 kill -USR2 $(cat /tmp/music_player.pid)

```

Pour finir, il faut activer **xbindkeys** au démarrage pour qu'il gère l'action associée à l'appui sur CTRL+X (START+SELECT). Il faut donc créer le fichier **~/.config/autostart/xbindkeys.desktop** et y insérer:

```

1 [Desktop Entry]
2 Type=Application
3 Exec=/usr/bin/xbindkeys
4 X-GNOME-Autostart-enabled=true
5 Name=xbindkeys

```

Redémarrez la session X pour prendre tous ces paramètres en compte.

Support des manettes et joysticks

Certains jeux supportent mal, ou pas du tout, les manettes (et les joysticks). Pour ces cas-là, nous pouvons utiliser [Padmapper](#) ou bien [Antimicro](#).

[Padmapper](#) s'utilise uniquement en ligne de commande, et se configure en éditant un fichier de configuration. C'est un projet associé au projet VersusBox.

[Antimicro](#) s'utilise via une interface graphique.

Padmapper

Padmapper est un outil pour Linux qui permet de simuler le clavier et la souris à partir de manettes de jeu. Il devient donc possible de presser des touches du clavier à partir de boutons d'une manette, et de déplacer la souris à partir d'un joystick.

Installation:

```
1 sudo pip install padmapper
```

Nous allons créer une configuration et des profils de manettes pour chaque jeu indépendamment.

Vous pouvez, à ce stade, lancer le jeu, et noter la configuration par défaut des touches, ça nous sera utile pour paramétriser notre cartographie de manette.

Ensuite vous pouvez lancer l'outil de capture pour avoir les identifiants des boutons de votre manette:

```
1 padmapper-capture
```

Suivez ensuite la documentation officielle sur [la page GitHub de Padmapper](#) pour écrire un fichier de configuration pour votre jeu.

Placez votre fichier de configuration ici: `/home/pi/RetroPie/roms/pc/<nom_du_jeu>/padmapper.json`.

Le script de lancement du jeu, tel que décrit au chapitre [Ajouter un jeu PC](#), devra ressembler à ceci:

```
1 #!/bin/bash
2
3 WORKING_DIR="/home/pi/RetroPie/roms/pc/<nom_du_jeu>"
4 export XDG_CONFIG_HOME="${WORKING_DIR}"
5
6 # Arret de la musique
7 kill -USR1 $(cat /tmp/music_player.pid)
8
9 echo $BASHPID > /tmp/pc_game.pid
10 padmapper /home/pi/RetroPie/roms/pc/<nom_du_jeu>/padmapper.json &
```

```
11      PADMAPPER_PID=$!
12
13      # Lancement du jeu
14      ${WORKING_DIR}/<executable>
15
16      # Stoppe padmapper et redemarre la musique
17      kill -9 $PADMAPPER_PID
18      kill -USR2 $(cat /tmp/music_player.pid)
```

Remplacez <nom_du_jeu> par le nom du répertoire du jeu, et <executable> par le nom de l'exécutable.

Antimicro

Antimicro est une application permettant de simuler des appuis de touches de clavier lorsque l'on actionne une manette. La méthode ci-dessous fonctionne pour deux manettes permettant à deux joueurs de jouer au même jeu.

Commençons par récupérer et installer **Antimicro**, vous pouvez trouver une version à jour pour Debian ici: <https://packages.libregeek.org/debian/pool/main/a/antimicro/>

Téléchargez-là, puis installez-là avec:

```
1      sudo dpkg -i antimicro_X.Y.deb
```

Nous allons créer une configuration et des profils de manettes pour chaque jeu indépendamment.

Vous pouvez, à ce stade, lancer le jeu, et noter la configuration par défaut des touches, ça nous sera utile pour paramétrier notre cartographique de manette.

Ensuite, lancez **Antimicro** comme suit:

```
1      XDG_CONFIG_HOME=/home/pi/RetroPie/roms/pc/<nom_du_jeu> antimicro
```

ce qui va démarrer une interface de configuration.

Si les caractères sont trop petits et peu lisibles, vous pouvez relancer cette interface avec:

```
1      XDG_CONFIG_HOME=/home/pi/RetroPie/roms/pc/<nom_du_jeu>
          QT_SCALE_FACTOR=2 antimicro
```

Il faut tout d'abord paramétrier vos manettes. Si elles sont identiques, un seul paramétrage suffit, sinon il faut le faire pour chacune. Allez dans le menu *Options* puis Cartographie de contrôleur de jeu. Renseignez alors chaque touche. Vous pouvez assigner *Retour* [.title-ref] à votre touche [Select].

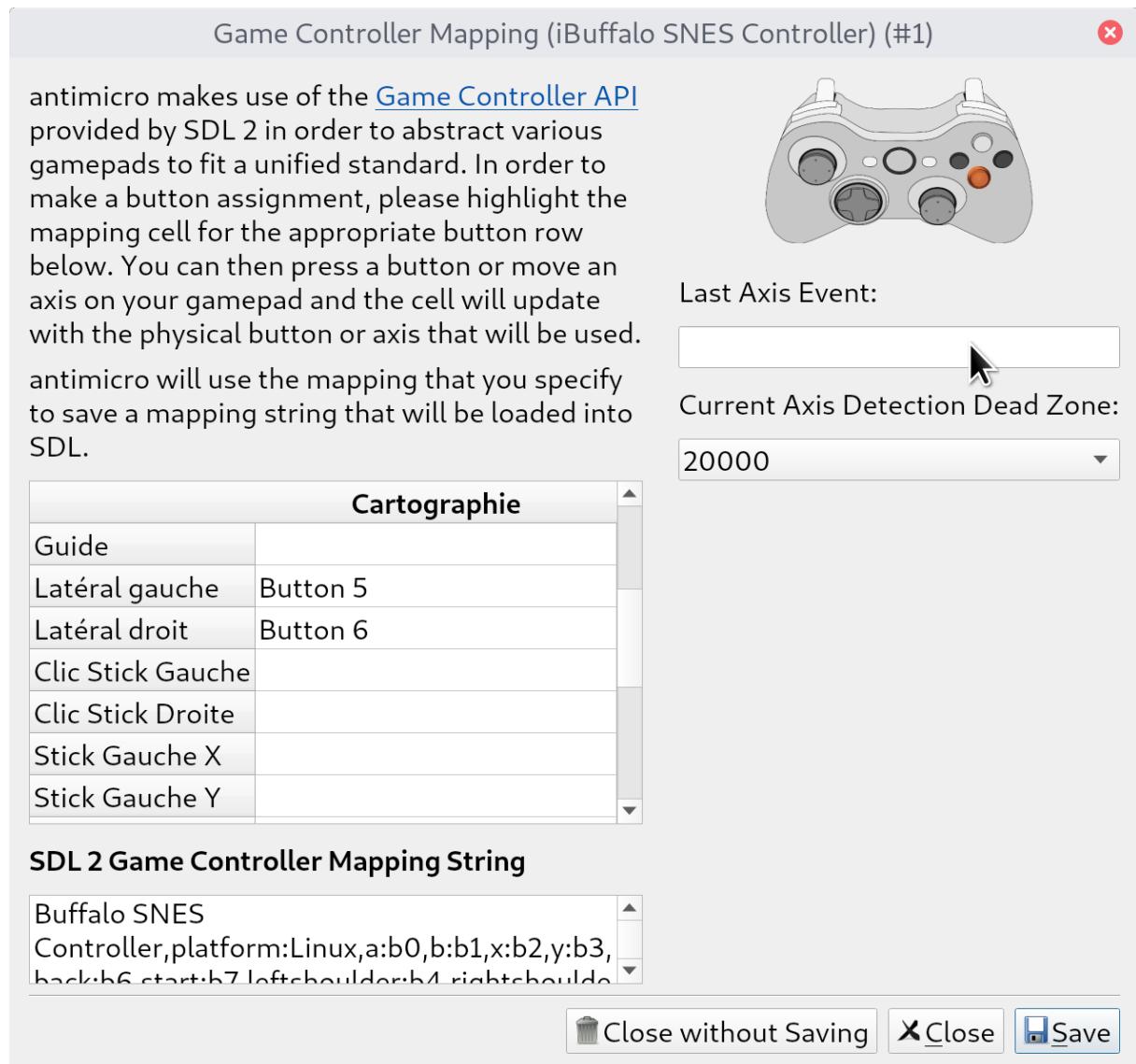


Figure 1: Cartographie de contrôleur de jeu

De retour dans l'interface principale, vous trouverez autant d'onglets que de manettes connectées. Vous pouvez alors, pour chaque touche de la manette, définir quelle touche du clavier sera simulée. Vous pouvez définir l'action de *Retour* à la touche *ECHAP*{} et l'action de *[Start]* à la touche *ENTREE*.



Figure 2: Correspondance contrôleur et touches simulées

Sauvegarder, pour chaque onglet de manette, le profil avec le bouton *Enregistrer*. Sauvez les profils dans un répertoire `/home/pi/RetroPie/roms/pc/<nom_du_jeu>/antimicro`

Enfin, rendez-vous dans le menu *Options* puis *Paramètres*, dans l'onglet général assurez-vous que la case permettant de charger automatiquement le dernier profil utilisé est cochée. Vous pouvez alors quitter **Antimicro**.

Le script de lancement du jeu, tel que décrit au chapitre [Ajouter un jeu PC](#), devra ressembler à ceci:

```

1 #!/bin/bash
2
3 WORKING_DIR="/home/pi/RetroPie/roms/pc/<nom_du_jeu>"
4 export XDG_CONFIG_HOME="${WORKING_DIR}"
5
6 # Arret de la musique
7 kill -USR1 $(cat /tmp/music_player.pid)
8
9 # Assignation du controleur 1 au profil p1
10 antimicro --hidden --no-tray --profile-controller 1 \
11     ${WORKING_DIR}/antimicro<nom_manette_1>.gamecontroller.amgp &
12 kill $!
13 # Assignation du controleur 2 au profil p2
14 # Le profil p1 est pre-charge car il etait le dernier profil
15 # utilise
15 antimicro --hidden --no-tray --profile-controller 2 \
16     ${WORKING_DIR}/antimicro/<nom_manette_2>.gamecontroller.amgp &
```

```

17      ANTIMICRO_PID=$!
18
19      # Lancement du jeu
20      ${WORKING_DIR}/<executable>
21
22      # Stoppe Antimicro et redemarre la musique
23      kill $ANTIMICRO_PID
24      kill -USR2 $(cat /tmp/music_player.pid)

```

Remplacez <nom_du_jeu> par le nom du répertoire du jeu, <nom_manette_1>, <nom_manette_2> par les noms des fichiers de profils précédemment sauvés pour chaque manette, et <executable> par le nom de l'exécutable.



Note: La méthode présentée dans ce script est un contournement, car la version d'**Antimicro** utilisée pour réaliser ce tutoriel ne permettait pas de charger plusieurs profils en ligne de commande, l'option -d de ce dernier, pour le lancer en tant que démon, étant bugguée.

Forcer la désactivation des contrôleurs dans un jeu

Si votre jeu avait un support des contrôleurs, même mauvais, vous devrez **désactiver** le support des contrôleurs dans votre jeu. En effet, si le jeu réagit à certains événements des contrôleurs, ça pourrait entrer en conflit avec les commandes relayées par Padmapper ou Antimicro, et perturber le bon fonctionnement du jeu.

Si votre jeu ne permet pas de désactiver le support des contrôleurs, suivez la solution suivante.

La solution proposée ici consiste à exécuter le jeu par le biais d'un autre utilisateur que l'utilisateur *pi*, cet utilisateur ne sera pas présent dans le groupe *input* et ne sera donc pas capable de lire les entrées des contrôleurs.

On commence par ajouter un nouvel utilisateur qui sera dédié à l'exécution de ces jeux, on nomme cette utilisateur *user_noctrl*:

```

1      sudo useradd -m -d /home/pi/RetroPie/roms/pc/user_noctrl -G video,
          audio user_noctrl -s /bin/bash

```

Pour chaque jeu concerné, changez les permissions sur ses fichiers:

```

1      sudo chown -R user_noctrl /home/pi/RetroPie/roms/pc/<nom_du_jeu>/<
          fichiers_du_jeu>

```

et modifiez le script de lancement du jeu pour qu'il soit exécuté de cette façon:

```

1      [...]

```

```
2      # Lancement du jeu
3      chmod o+rwx ~/Xauthority
4      sudo -u user_noctrl "${WORKING_DIR}/<executable>"
5      [...]
```

Pour que l'utilisateur *user_noctrl* ait accès à l'audio sans problèmes, éditez */etc/pulse/default.pa* et ajoutez-y à la fin:

```
1      load-module module-native-protocol-unix auth-anonymous=1 socket=/
      tmp/pulse-socket
```

ce qui permettra à *user_noctrl* d'accéder au démon audio via un socket UNIX.

Il faut ensuite modifier la configuration Pulse de *user_noctrl* pour qu'il utilise ce socket. Connectez-vous au shell en tant que *user_noctrl*:

```
1      sudo -u user_noctrl -s
```

et exécutez:

```
1      mkdir -p .config/pulse
```

puis éditez *~/.config/pulse/client.conf* pour y insérer:

```
1      default-server = unix:/tmp/pulse-socket
```

et faites *CTRL+D* pour sortir du shell de *user_noctrl*, rebootez, et les utilisateurs *pi* et *user_noctrl* auront accès à l'audio simultanément.

Voilà, le jeu n'a plus accès aux contrôleurs de jeux, seulement au clavier et à la souris.

D'autres méthodes sont listées [sur le wiki d'Antimicro](#).

Ajouter un jeu Steam

Dans la procédure ci-dessous, on considère qu'il n'y a pas Steam sur le PC RetroPie, et que l'on copie les jeux souhaités depuis une autre machine sur laquelle Steam est installé. En effet, nous ne souhaitons pas ici passer par l'interface de Steam pour lancer notre jeu, nous souhaitons y accéder par notre interface EmulationStation.

L'installation de Steam sur la machine n'est pas nécessairement requise, si le jeu souhaité n'utilise pas de fonctionnalités de l'API Steam. Certains jeux ne fonctionneront donc pas sans Steam.

Seuls les jeux Steam compatibles Linux sont supportés.



Attention: Cette procédure ne vise pas à aider au piratage de jeux, vous devez avoir acquis le jeu sur la plateforme Steam pour l'utiliser ensuite sur votre PC de jeu RetroPie.

Copier le jeu Steam vers le PC RetroPie

Sur le PC Steam, les jeux se situent dans votre répertoire:

```
1 /home/\<nom\_\utilisateur\>/.\local/share/Steam/SteamApps/common
```

Copiez simplement le dossier de votre jeu sur votre machine RetroPie, à l'endroit de votre choix.

Ajouter le jeu dans la collection PC d'EmulationStation

Suivez la procédure [Ajouter un jeu PC](#) détaillée précédemment. L'exécutable du jeu se situe dans le dossier précédemment copié, et est spécifique à chaque jeu, à vous de l'identifier.

Dépendances

Certains jeux Steam peuvent dépendre de certaines librairies que l'on trouve habituellement sur un système Ubuntu, pour lequel est prévu Steam à l'origine.

Vous pouvez installer les librairies suivantes, qui pourraient s'avérer utiles:

```
1 sudo apt install libgtk2.0 libxtst6:i386 libxrandr2:i386 libglib2
      .0-0:i386 \
2 libgtk2.0-0:i386 libpulse0:i386 libgdk-pixbuf2.0-0:i386 \
3 libcurl4-openssl-dev:i386 libopenal1:i386 libusb-1.0-0:i386 \
4 libdbus-glib-1-2:i386 libnm-glib4:i386 libnm-util2:i386
```

Avancé

Cette section s'adresse à des utilisateurs d'un niveau plus avancé, qui veulent tenter de faire fonctionner sans Steam des jeux plus récalcitrants.

Si l'un de vos jeux ne démarre pas correctement, vous pouvez commencer par vérifier s'il ne lui manque aucune librairie:

```
1 ldd executable_du_jeu
```

LDD vous informera si une librairie est manquante. Vous pouvez ensuite chercher si la librairie est fournie par un paquet Debian en la recherchant sur <https://packages.debian.org>, section **Search the contents of packages**, et installer l'éventuel paquet.

D'autres librairies peuvent-être dynamiquement chargées pendant l'exécution du jeu. Vous pourrez les identifier avec la commande suivante:

```
1      strace -f executable\du\jeu
```

Cette commande donne énormément d'informations peu digestes au sujet des appels système que le jeu effectue, elle peut vous permettre d'identifier un fichier manquant au jeu.