

Environnement pour Windows 🖥️ (approche suggérée)	1
Environnement pour Ubuntu 🐧 (approche suggérée si vous préférez Ubuntu)	2
Détails pour Ubuntu 🐧 (avancé)	3
macOS 🍏 (approche partielle)	3

Environnement pour Windows 🖥️ (approche suggérée)

Afin de travailler sur Windows, suivez ces étapes:

- Vous n’avez plus à utiliser la machine virtuelle ARM;
- Récupérez [cette archive](#) 📁 et décompressez-la sur votre ordinateur;
- Compilez le programme `devoir5.asm` en double-cliquant sur l’exécutable `make`, puis exécutez-le en double-cliquant sur l’exécutable `exec`;
- Il est possible que Microsoft Defender tente de bloquer l’exécution des scripts (`.bat`) que j’ai écrits moi-même. Dans ce cas, il suffit de cliquer sur « Informations complémentaires » puis « Exécuter quand même » (il n’y aura plus d’avertissement dans le futur);
- Utilisez un éditeur de votre choix afin d’écrire votre code, par exemple:
 - Visual Studio Code avec une extension comme [celle-ci](#) 📁;
 - Atom: pour installer la coloration syntaxique du NES: appuyez sur `Ctrl+Shift+P`; tapez « `install package` »; puis recherchez et installez « `language-65asm` ». Vous pourrez maintenant choisir la coloration « `6502 Assembly (cc65)` » à l’aide du raccourci clavier `Ctrl+Shift+L`;
 - Notepad++ avec [cette extension](#) 📁.
- Les boutons `A`, `B`, `select` et `start` sont respectivement assignés aux touches `f`, `d`, `s` et Entrée du clavier. La croix directionnelle est assignée aux flèches du clavier. Vous pouvez les reconfigurer au besoin dans le menu du haut sous « Options > GamePad Config » ou « Config > Gamepad Config »;
- Pour lancer l’exécutable de la solution ou du bonus du devoir 5, double-cliquez sur le script `solution` ou `bonus`.

Vous pouvez aussi compiler le programme `exemple.asm`, décrit dans les notes de cours, en double-cliquant sur l’exécutable `make_exemple`, puis l’exécuter en double-cliquant sur l’exécutable `exec_exemple`.

Environnement pour Ubuntu (approche suggérée si vous préférez Ubuntu)

Afin de travailler sur Ubuntu, suivez ces étapes:

- Installez l'émulateur `fceux` en tapant cette commande dans un terminal:

```
sudo apt install fceux
```

- Récupérez [cette archive](#)  et décompressez-la sur votre ordinateur;
- Dans le dossier obtenu, placez le fichier que vous désirez compiler, par ex. « `devoir5.asm` » (s'y trouve déjà).
- Dans un terminal, déplacez-vous dans ce dossier avec la commande « `cd` »;
- Entrez la commande

```
./nesasm devoir5.asm && fceux devoir5.nes
```

La première partie de la commande compile votre programme et la seconde l'exécute dans un émulateur. Il s'agirait de l'équivalent de « `make && ./devoir5` » sur ARMv8. Si vous préférez, vous pouvez les exécuter l'une à la suite de l'autre.

- Pour une expérience plus agréable, effectuez ces configurations dans l'émulateur:
 - Appuyez sur la touche virgule (,) de votre clavier afin d'afficher l'état de la manette;
 - Sous « `Option > Video Config` » ou « `Config > Video Config` », cochez « `Clip sides` » et assignez 4,0 à « `X scaling factor` » et à « `Y scaling factor` » (redémarrez l'émulateur au besoin).
- Les boutons `A`, `B`, `select` et `start` devraient être assignés aux touches `f`, `d`, `s` et Entrée du clavier, et la croix directionnelle aux flèches du clavier. Vous pouvez les reconfigurer au besoin dans le menu du haut sous « `Options > GamePad Config` » ou « `Config > Gamepad Config` ».
- Utilisez un éditeur de votre choix afin d'écrire votre code. Par exemple, vous pouvez utiliser Visual Studio Code avec une extension [comme celle-ci](#) , ou bien [Atom](#)  en installant la coloration syntaxique du NES en tapant cette commande (une seule fois) dans un terminal:

```
apm install language-65asm
```

Vous pourrez choisir la coloration « `6502 Assembly (cc65)` » à l'aide du raccourci clavier `Ctrl+Shift+L`;

- Pour lancer l'exécutable de la solution ou du bonus du devoir 5, entrez l'une de ces commandes:

```
fceux solution.nes  
fceux bonus.nes
```

Détails pour Ubuntu (avancé)

— Afin d'exécuter `nesasm` dans n'importe quel dossier de votre ordinateur et sans le préfixe « `./` », déplacez l'exécutable `nesasm` dans votre répertoire « `/usr/local/bin/` ».

— Alternativement, vous pouvez compiler l'assembleur vous-même:

- téléchargez [ce code source](#) ;
- compilez-le avec la commande

```
cd source && make && sudo make install
```

- vous pourrez invoquer `nesasm` dans n'importe quel dossier sans le préfixe « `./` » (`make install` s'occupe du déplacement dans « `/usr/local/bin/` »).

macOS (approche partielle)

Je n'ai pas accès à un ordinateur avec macOS. Vous pouvez obtenir une machine virtuelle Ubuntu sur le réseau public du département, puis suivre les [instructions pour Ubuntu](#).

Sinon, vous devez obtenir un assembleur et un émulateur:

— Vous devriez pouvoir obtenir l'assembleur `nesasm` en téléchargeant [ce code source](#) , puis en le compilant vous-même avec la commande

```
cd source && make && sudo make install
```

— Tentez d'installer l'émulateur `fceux`. Par exemple, si vous avez le gestionnaire de paquets Homebrew, vous devriez y arriver en entrant cette commande dans un terminal:

```
brew install fceux
```

Alternativement, consultez [cette page](#) .

— Ensuite, vous devriez pouvoir compiler et exécuter vos programmes comme [sur Ubuntu](#).