

# Space Invaders avec Arduino

Colin Dooley (Espagne)

Vous souhaitez combiner votre amour du jeu avec votre passion pour Arduino ? Voici un exemple innovant d'utilisation d'Arduino pour programmer Space Invaders

J'ai grandi en programmant des jeux vidéo dans les années 1980, alors quand j'ai reçu mon premier Arduino, j'ai naturellement voulu en programmer avec. Les Arduino ont les mêmes limitations de mémoire et de processeur que les premiers ordinateurs domestiques, donc je me suis immédiatement senti à l'aise. Les choses ont évolué depuis lors. J'ai maintenant un EDI et un langage de programmation de haut niveau, mais les étapes de base et les techniques d'optimisation pour programmer des jeux sur un Arduino sont les mêmes. Il y a des nombres hexadécimaux, beaucoup de bits à décaler, du temps passé à essayer de tout faire tenir dans une minuscule mémoire vive, etc. Mes anciennes compétences en langage d'assemblage étaient encore très utiles ; j'ai fini par faire beaucoup de désassemblage de code pour voir ce que le compilateur faisait mal (pas mal !) et par adapter mon code pour obtenir les meilleurs résultats.

Le jeu que j'ai choisi est *Space Invaders* (**figure 1**). C'est un jeu emblématique, et pas trop complexe à programmer. L'idée que j'avais en tête était d'utiliser un Arduino pour créer une minuscule machine Space Invaders qui soit aussi proche que possible de l'original (*pixel perfect*).

J'ai trouvé un site Web qui présentait un désassemblage complet de la ROM de Space Invaders et je m'en suis servi comme référence pour la logique du jeu. C'était beaucoup de travail pour obtenir tous les détails corrects, et j'ai beaucoup appris sur le jeu dans le

processus. (Il s'avère qu'il y a des bouts de code vieux/mort dans les ROM qui donnent des indices sur le développement). Le site en question est [www.computerarcheology.com/Arcade/SpaceInvaders/](http://www.computerarcheology.com/Arcade/SpaceInvaders/).

Je peux affirmer que le résultat est aussi proche du jeu original qu'il est possible de l'obtenir sans exécuter les ROM originales sur un émulateur, ce qui ne peut pas être fait sur un Arduino 8 bits ! Toute la logique est là. Si vous êtes un joueur expert, vous pouvez compter vos tirs pour obtenir le maximum de points des soucoupes volantes, etc. Le seul véritable changement par rapport à l'original est que j'ai fait en sorte que les bombes de l'envahisseur tombent un peu plus lentement, car il était difficile de jouer au jeu sur un si petit écran.

## Graphisme

Le jeu original Space Invaders dispose d'un écran de 224 × 256 pixels. Mon premier travail a été de trouver le plus petit écran possible avec cette qualité (il doit être *pixel perfect*, rappelez-vous !). Après quelques recherches sur AliExpress, j'en ai trouvé un approprié et je me suis mis au travail.

Tous les graphiques du jeu ont été faits à l'ancienne, avec un crayon et du papier millimétré (**figure 2**). Ils sont dessinés sous forme de pixels, convertis en nombres hexadécimaux (je peux encore le faire dans ma tête), puis saisis dans le programme. Vous pouvez les trouver dans le fichier `graphics.h` dans le code source ([www.thehighprotondiet.com](http://www.thehighprotondiet.com)).



Figure 1. Le projet Space Invaders.

L'écran communique par SPI, et j'avais peur que cela soit un goulot d'étranglement pour un jeu vidéo qui tourne à 60 fps (**figure 3**). Heureusement, Space Invaders ne change pas beaucoup de pixels entre deux images et la « lenteur » du bus SPI n'était pas un problème. Après quelques week-ends, j'ai réussi à faire fonctionner les graphiques du jeu en n'utilisant qu'environ 10 % du CPU de l'Arduino. Le processeur de l'Arduino est beaucoup plus puissant que ceux des ordinateurs domestiques des années 1980. Il s'est avéré que le son nécessitait beaucoup plus de puissance CPU que les graphiques.

## Son

J'ai trouvé les échantillons sonores de Space Invaders dans le cadre de l'émulateur MAME.

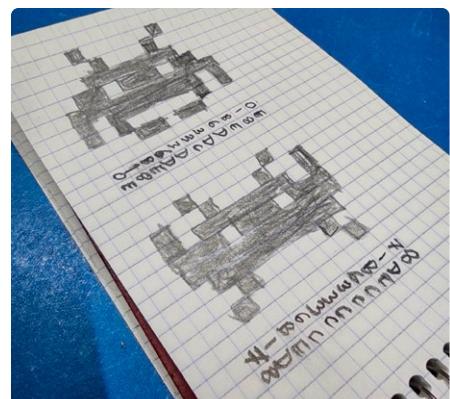


Figure 2. J'ai créé les graphiques avec un crayon et du papier millimétré.

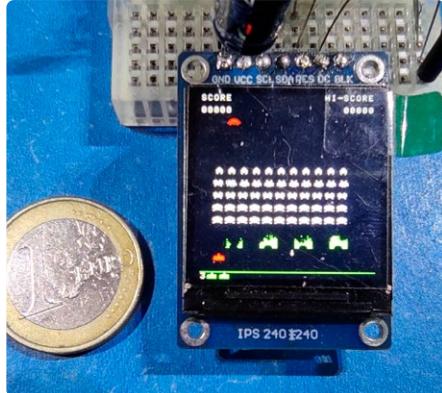


Figure 3. L'écran est compact.

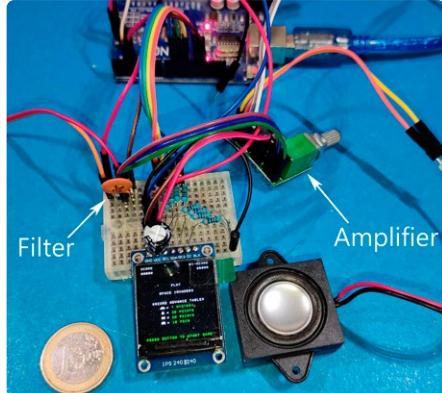
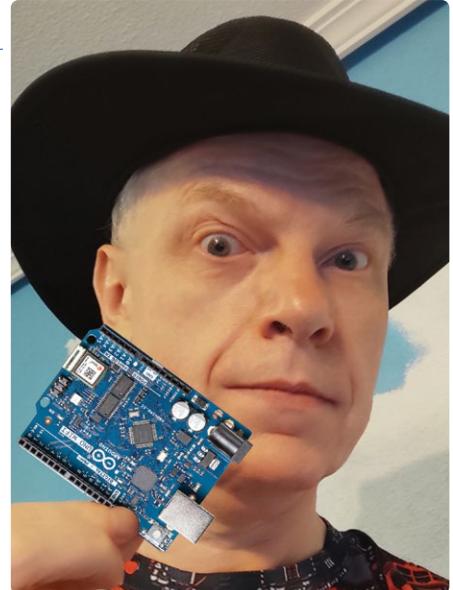


Figure 4. La partie électronique comprend un simple filtre passe-bas, un petit amplificateur audio et un haut-parleur.



Le seul problème était qu'ils faisaient 49 Ko et que l'Arduino n'avait plus qu'environ 20 Ko de mémoire de libre après avoir chargé mon programme. J'ai fini par convertir les sons en échantillons de 6 bits au lieu de 8 bits et par les tasser dans la mémoire. Il y avait un son qui ne tenait tout simplement pas en mémoire (celui de la mort de la soucoupe volante), j'ai donc substitué avec la mort du joueur. J'espère que personne ne le remarquera. Je ne l'avais pas prévu ainsi, mais l'utilisation de sons 6 bits présente un autre avantage considérable : Space Invaders possède quatre canaux sonores, et vous pouvez additionner quatre nombres 6 bits pour obtenir un son 8 bits. Cela signifie que le mixage du son peut être effectué de manière optimale sur un processeur 8 bits comme l'ATmega328 de l'Arduino. Le son 8 bits final sort via un signal MLI généré par le minuteur matériel no. 2 de l'ATmega328. Le signal passe par un simple filtre passe-bas réalisé avec une résistance et un condensateur, puis dans un petit amplificateur audio et un haut-parleur que j'ai trouvé sur AliExpress (**figure 4**). Le résultat est génial ! Le son rappellera certainement beaucoup de souvenirs à tous ceux qui ont connu cette époque.

## Commandes du jeu

Pour les commandes, j'ai trouvé un joystick vraiment minuscule sur AliExpress (**figure 5**). Il était trop petit et le mouvement était très rigide. J'ai donc imprimé en 3D une petite extension. Le résultat final a l'air vraiment bien et est beaucoup plus facile d'usage.

## Le boîtier

J'ai récemment acheté une découpeuse laser pour mon atelier à la maison, donc le boîtier est en MDF découpé au laser à l'aide d'un plan que j'ai trouvé sur Internet (**figure 1**). Je l'ai légèrement modifié pour ajouter des dessins et des découpes pour l'écran et le joystick. Il

constitue une bonne base pour me permettre de trouver comment réaliser les joysticks et leur électronique interne.

## Le futur

Je n'en ai pas encore fini avec ce projet. Le boîtier présenté ici est très générique, ce n'est pas une vraie machine Space Invaders. L'originale utilisait un système à miroirs pour superposer les graphiques des envahisseurs sur un fond coloré. Combiné à d'autres pièces astucieuses, cela donne un effet 3D à plusieurs couches. Je prévois de reproduire ce système prochainement ! Vous pouvez trouver plus d'informations et le code source de ce projet sur mon site web : [www.thehighprotondiet.com](http://www.thehighprotondiet.com).

220557-04 — VF : Maxime Valens

## Des questions, des commentaires ?

Contactez Elektor ([redaction@elektor.fr](mailto:redaction@elektor.fr)).

## À propos de l'auteur

Colin Dooley a grandi en programmant des ordinateurs de salon 8 bits dans les années 1980. Son premier véritable emploi a été dans une société de jeux vidéo à Sheffield, en Angleterre, au milieu des années 80, à l'époque des Sinclair Spectrums, Amstrads et Commodore 64 qui constituent aujourd'hui une grande partie de la scène rétro. Selon Colin, « c'est un honneur pour moi d'être présenté dans une édition spéciale du magazine Elektor. Les magazines comme Elektor étaient l'une des rares ressources dont nous disposions pour apprendre à connaître les ordinateurs et la programmation informatique à l'ère pré-Internet. »



Figure 5. Le minuscule joystick.