

# 1 Le jeu de la vie de Conway

Le jeu de la vie de Conway est un automate cellulaire. C'est à dire que c'est un modèle mathématique basé sur l'automatisation informatique. Les automates cellulaires sont étudiés dans la science de l'automatisation. Le jeu de la vie a été inventé par le scientifique John Horton Conway dans les années 70 et c'est un jeu à 0 joueurs. Cela signifie que son évolution est définie uniquement par son état de départ.

Le jeu de la vie est passionnant car ses règles sont extrêmement simplistes. Cependant il est possible de le voir réaliser des figures extrêmement complexes au fil du temps.

Les règles de ce jeu sont les suivantes :

- Chaque cellule vivante disposant de moins de 2 cellules voisines vivantes meurt d'isolement
- Chaque cellule vivante disposant de 2 ou 3 cellules voisines vivantes reste vivante à la prochaine itération.
- Chaque cellule vivante disposant de plus de 3 cellules voisines vivantes meurt d'étouffement
- Chaque cellule morte disposant exactement 3 cellules voisines vivantes devient vivante grâce au phénomène de reproduction.

Voici un exemple du jeu de la vie



FIGURE 1 – Exemple d'un jeu de la vie

## 1.1 Bonus

Si vous le souhaitez vous pourrez obtenir un bonus si vous implémentez une grille a taille paramétrable.

## 2 Le Snake

Le snake est un jeu vidéo sorti pour la première fois dans les années 70. Le but du jeu est de déplacer un serpent sans qu'il ne se morde la queue tout en le faisant manger des fruits pour le faire grandir. Plus le serpent est grand, plus il devient difficile de le manoeuvrer. Il est possible de déplacer le serpent avec les flèches du clavier et des fruits apparaissent de manière aléatoire à l'écran de jeu. Le serpent ne peut pas s'arrêter, il avance en continu dans une direction jusqu'à ce qu'on change cette direction.

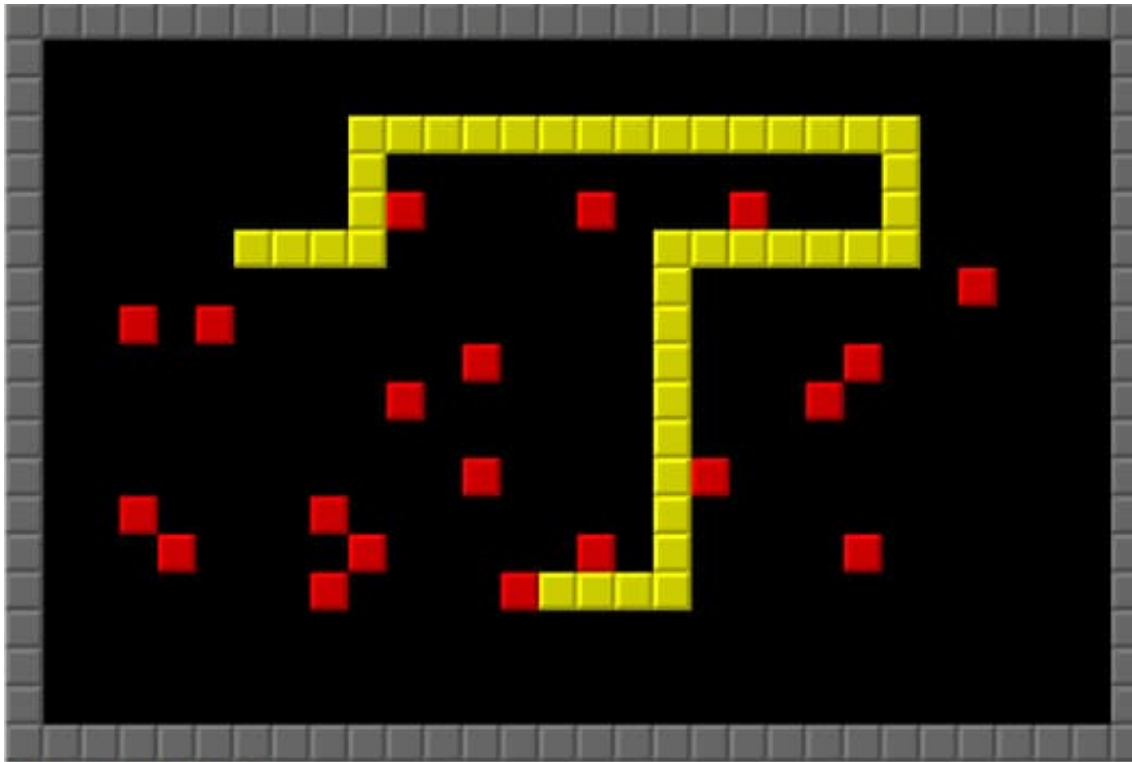


FIGURE 2 – Exemple du jeu de snake

## 2.1 Bonus

Vous pouvez implémenter une fonctionnalité de pause pour obtenir un bonus si vous le souhaitez.

## 3 Votre Projet

### 3.1 Gestion de code source

Pour ce projet, nous allons procéder à une revue de code en temps réel et une gestion de code source comme il serait fait en entreprise. Ainsi, l'enseignant a créé les dépôts git de chacun de vos projets et il ne sera pas possible pour vous de faire des pushes sur la branche main. Le seul moyen d'envoyer du code sur la branche main sera d'être soumis à la revue de code du professeur. Pour demander une revue de code, vous allez devoir faire une merge request, celle-ci sera acceptée uniquement si elle respecte les standards de programmation demandé par l'enseignant. Si votre code ne respecte pas les standards, il vous sera retourné et vous devrez corriger les modifications qui vous seront demandées. Ainsi pour chaque nouvelle feature, vous devrez créer une nouvelle branche portant le nom de la feature et envoyer une merge request une fois que la feature sera terminée. Attention, aucune pénalité ne vous sera accordée si votre code n'est pas accepté. Cela a seulement un but pédagogique. Vous serez évalué sur votre capacité à vous adapter et à améliorer votre processus de programmation tout au long de la session.

Définition d'une merge request acceptable :

- Le code est écrit proprement, en évitant au maximum les syntaxes difficiles à lire
- Le code est compréhensible et des commentaires décrivent toutes les parties
- Le standard de nomenclature C# est respecté ([lien du standard ici](#))
- Les noms de fonctions et variables sont descriptifs, sans abréviations et bien orthographiés.
- La merge request contient une seule feature et la description du commit est bien écrite
- Le code est correct d'un point de vue algorithmique et orienté objet.

### 3.2 Interface graphique

Vous devrez réaliser une interface graphique pour votre projet. Les deux projets sont composés d'une grille, ainsi vous allez devoir utiliser un canevas pour votre interface WPF. Voici des maquettes d'exemples.

**Bonne chance !**

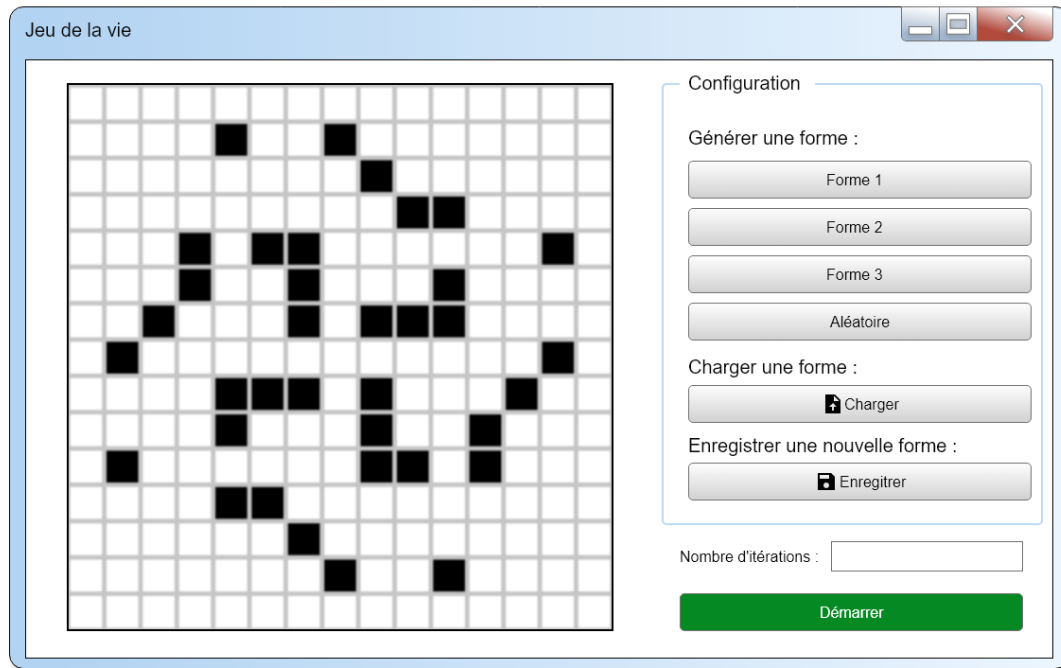


FIGURE 3 – Interface JDV

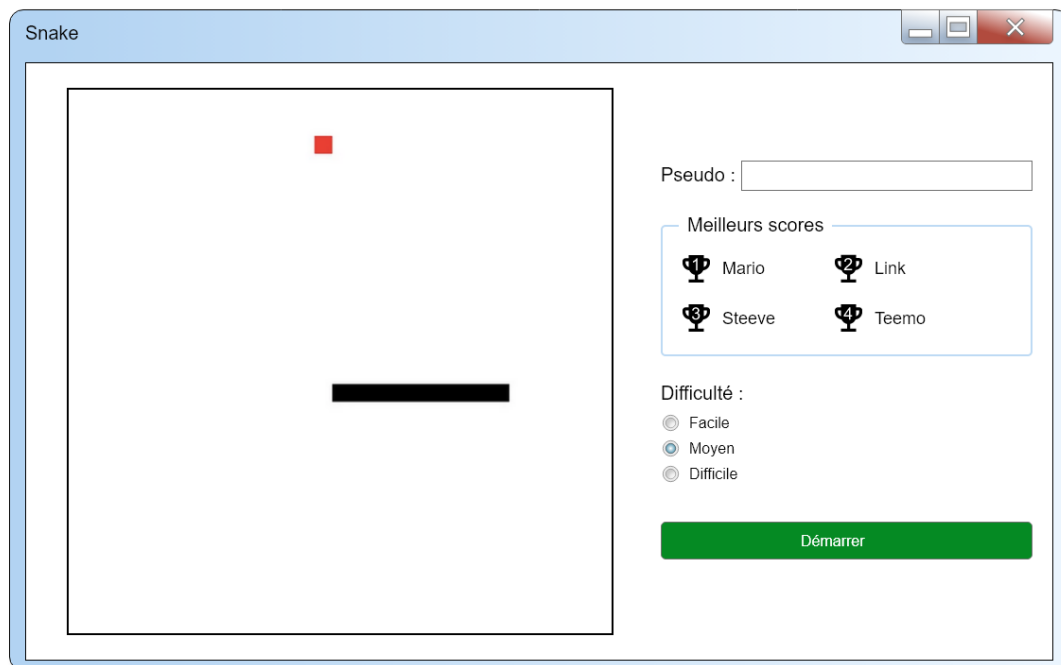


FIGURE 4 – Interface Snake