

# Jeux de Nim

---

Projet informatique 1  
L1 Mathématiques et Informatique  
Année 2021-2022 - Semestre 1

Fait par Amal Abddalah, Nicolas Seban et Adam Souiou

---

## Table des matières

---

### Guide d'installation et d'utilisation du programme

1. Installation
2. Règles de jeu
3. Comment jouer ?
4. Utilisation des menus

### Fonctionnement de l'équipe et organisation du travail

1. Répartition du travail entre les membres
2. Modalités d'organisation, fréquence des rencontres
3. Difficultés rencontrées

### Avancement du projet

1. État d'avancement dans les tâches obligatoires
2. Bugs connus éventuels
3. Améliorations apportées
4. Améliorations éventuelles
5. Informations intéressantes sur le code
6. Ce qui nous a plu ou déçu dans le projet

### Crédits

---

# Guide d'installation et d'utilisation du programme

---

## Installation :

Pour installer notre jeu, rendez-vous sur : <https://github.com/Nicolas-93/PR1-Abdallah-Souiou-Seban.git>

Vous obtiendrez un dossier compressé, décompressez le et placez le où vous le voulez.

Installez la police Biometric Joe si elle ne l'est pas déjà par défaut.

Il faudra aussi vous fournir de Pillow

- Pour plus d'explications sur comment installer Pillow, rendez-vous: <https://blog.finxter.com/python-install-pil/>

L'installation de Pygame, bien qu'optionnel, est recommandée pour pouvoir profiter des sons et de l'ambiance sonore.

- Pour plus d'explications sur comment installer Pygame, rendez-vous: <https://www.pygame.org/wiki/GettingStarted>

Une fois fais, à l'aide du terminal déplacez vous dans le répertoire où vous avez placé le dossier et exécuter nim.py dans votre terminal !

## Règles de jeu :

Le jeu de Nim peut se jouer de deux façons:

- Le jeu classique, où vous n'avez qu'une ligne d'allumettes et une liste de coups possibles prédéfinies.
- Le jeu de Marienbad, où vous avez plusieurs lignes d'allumettes mais vous pouvez sélectionner autant d'allumettes sur une rangée que vous voulez.

Chacun de ces jeux peut se jouer avec deux règles (voir Utilisation des menus):

- Le mode normal, où le gagnant de la partie est celui qui récolte la dernière allumette.
- Le mode misère, où le perdant de la partie est celui qui prend la dernière allumette.

## Comment jouer ?

Dans ce programme, tout se fait à la souris. Que ce soit la navigation à travers les menus, que la sélection d'allumettes.

Pour la sélection d'allumettes:

- Le clic gauche sert à augmenter votre sélection
- Le clic droit à diminuer

L'augmentation et la diminution est par défaut de un, mais dans le cas où vous jouer au mode classique avec une liste de coups discontinue, les choix impossibles à faire seront simplement ignorés.

Une fois votre sélection faite, pressez le bouton "Fin de tour" afin de valider votre tour.

- Dans le cas où vous jouez contre l'ordinateur, ce bouton sert aussi à valider le tour de votre adversaire, vous laissant donc le temps de regarder ce que l'ordinateur compte faire.

## Utilisation des menus :

En exécutant le programme, vous vous trouvez face à trois boutons, les deux premiers sont assez explicites car ils vous amènent aux jeux expliqués plus haut. Et le troisième bouton, le bouton "option".

Le menu d'options vous permet de changer de nombreux paramètres de jeux:

- Mode 2 joueurs
  - Qui une fois cliqué dessus alterne avec un mode solo, avec un choix de difficulté.
- Un choix de nombre d'allumettes pour le jeu classique.
- Changement entre mode misère et mode normal
- Activation des animation d'arrière plan ainsi que le son de jeu.
  - A noter que si vous n'avez pas pygame, il est impossible d'activer le son.

Cependant si vous voulez changer la liste de coups possibles pour le jeu classique, les séries d'allumettes pour le jeu de marienbad, la taille de la fenêtre ainsi que les options par défauts, amusez vous à modifier la première moitié du fichier `cfg.py` !

Cependant faites attention à bien garder la même syntaxe des listes.

---

# Fonctionnement de l'équipe et organisation du travail

---

## Répartition du travail entre les membres :

La répartition était assez simple.

Au début du projet, il n'y avait pas réellement d'organisation prévue, nous faisons tous un peu de tout histoire d'être productif.

Ce n'est qu'une fois la structure du jeu classique et du jeu de marienbad faite que nous avons décider d'avoir une organisation plus soignée et moins improvisée.

- Nicolas Seban s'est penché sur le polissage de l'architecture du programme :
  - Correctif de nombreux bugs
  - Finition du jeu de Marienbad
  - Optimisation globale du programme
  - Amélioration de fltk
  - Création d'une façon simple de gérer les boutons

- Adam Souiou a travaillé sur des aspects plus ciblés du jeu :
  - Correctif de nombreux bugs
  - Gestion des tours des joueurs
  - Implémentation d'un mode solo face aux ordinateurs
  - Ajout d'un mode impossible, intitulé *Hardcore* dans les deux jeux et modes de jeu
  - Élaboration d'un menu d'options
- Amal a décidé de prendre en charge tout le côté accessibilité utilisateur :
  - Aide à la création des boutons
  - Travail sur l'ambiance sonore et musicale des parties
  - Création d'animations d'arrière-plan
  - Amélioration du gameplay des joueurs
  - Création d'un écran de fin de partie

## Modalités d'organisation, fréquence des rencontres :

L'organisation était des plus sommaire, nous avons utilisé la première séance pour établir ce que nous voulions faire, quoi faire et comment faire.

Cette séance a servi surtout de base pour la suite, car maintenant que nous savions dans quelle direction partir, nous avons chacun de notre côté implémenter quelque chose dans le code dès que nous le pouvions, tout en étant sûr que tout fonctionnait.

Pour les rencontres, nous avons principalement utilisé les séances en université pour travailler et pour planifier ce que chacun allait tenter de faire jusqu'à la prochaine séance.

Puis nous avons augmenté les fréquences des réunions en faisant des appels sur Discord pendant la dernière semaine, comme nous étions sur la dernière ligne droite, nous devions être sûrs de ne pas avoir oublié quoi que ce soit pendant la finalisation du projet.

## Difficultés rencontrées :

Nous avons eu surtout deux grandes difficultés pendant ce projet:

- La simplification bien trop importante de `fltk` concernant les images. Rendait impossible des choses pourtant essentielles comme la redimension d'image mais également la rotation de ces dites images.
  - Problème qui a été réglé par une légère modification de la librairie
- De nombreux bugs autour des ordinateurs.
  - Étant de la programmation assez complexe, il nous a fallu beaucoup de détermination et de persévérance pour aboutir à un résultat satisfaisant.

---

# Avancement du projet

---

## État d'avancement dans les tâches obligatoires :

Les tâches obligatoires qui ont été réalisées sont les suivantes :

- Structures de données et fonctions de base
  - Fonction pour déterminer si un coup est possible
  - Fonction pour jouer un coup et modifier l'état du jeu
  - Fonction pour détecter la victoire d'un des joueurs
- Interface visuelle et jouabilité stable
- Mode de saisie des actions des joueurs à la souris
- Grande majorité de paramétrage du jeu dans le menu options
- Stratégies de l'ordinateur

## Bugs connus éventuels :

A l'heure où j'écris ce texte un seul bug est connu :

- L'ordinateur *Hardcore* semble ne pas comprendre la stratégie parfaite en jeu de Marienbad en mode misère

Nous continuons de travailler sur ce point.

## Améliorations apportées :

Plusieurs améliorations sont notés:

- Différents menus pour naviguer à travers les différentes parties du programme
- Deux ordinateurs de difficultés différentes, une basée sur l'aléatoire, l'autre sur la stratégie parfaite peut importe l'ensemble des coups possibles
- Création d'animations pour un meilleur côté esthétique
- Effets sonores lors d'interactions avec le programme
- Musique d'accueil, de duel, ainsi qu'une musique de duel et de fin de jeu spécial contre un des deux ordinateurs

## Améliorations éventuelles :

Voici les quelques améliorations que nous pourrions apporter dans un futur plus ou moins proche :

- Correctif du bug mentionné plus haut
- Possibilité de changer la taille de la fenêtre dans le menu option
- Possibilité de changer la liste des coups possibles dans le menu option
- Créer de meilleures marqueurs distinctifs entre les tours de chaque joueur
- Amélioration de l'identité des deux bots pour leur donner plus de caractère
- Un curseur personnalisé
- Support des champs de texte

- Supporter les textes sur plusieurs lignes dans bouton.py
- Ajuster/Zoomer les allumettes de manière à optimiser l'espace disponible
- Animation de disparition des allumettes: les allumer!!

## Ce qui nous a plus ou déplu dans le projet :

Ce qui nous a plu :

- L'entente et la productivité inattendues du groupe
- La création des ordinateurs
  - Dont un qui humilie le joueur
- Tout le côté esthétique apporté par les animations et les sons
- Faire les boutons

Ce qui nous a déplu :

- La rigidité de fltk

---

## Crédits

---

Allumette de base fournie par pixabay : <https://pixabay.com/fr/vectors/correspondre-allumettes-feu-1717377/>  
Allumette brûlée était l'allumette de base modifiée par Amal.

Les effets sonores viennent de la bibliothèque de sons de Sonic Mania

Les musiques:

- "Battle Against an Unfathomable Enemy" par Robert Blaker
- "It Be Like That Sometimes" par James Roach
- "VALID END" par James Roach
- "WORST END" par James Roach