

Ce « jeu » présente peu d'intérêt mais permet de mettre en place les structures dont vous avez besoin.

## Objectif temps réel

Vous devez passer votre jeu en temps réel, c'est-à-dire que le serpent se déplace perpétuellement sans nécessité de presser une touche. Le jeu devient un peu plus intéressant, mais pas passionnant.

Vous pourrez proposer plusieurs options de vitesse de déplacement afin de rendre le jeu plus ludique.

## Objectif multijoueurs

Dans cette version, au moins deux serpents combattent sur l'écran, gérés par des touches clavier séparées. On peut imaginer 3 ou 4 joueurs simultanés, si tant est qu'on puisse faire jouer autant de joueurs simultanément sur un même clavier.

Ici, le jeu devient pertinent : on doit manger autant de bonus que possible afin de grandir et encercler ou acculer l'adversaire à la faute.

## Objectif ludique

Toute amélioration rendant le jeu plus ludique est la bienvenue. Suggestions non exhaustives :

- bonus augmentant ou diminuant la vitesse de celui qui mange ; les serpents ont donc des vitesses différentes (pendant un temps donné?)
- malus fatal à ne pas manger
- maladie qui fait rétrécir, ou perdre le contrôle...
- chargement d'un plan (fichier) contenant des murs
- sons
- plan circulaire : on sort d'un côté et réapparaît de l'autre
- sauvegarde des parties, des scores
- ...

## Objectif stratégie random (**obligatoire**)

Vous devez mettre un œuvre un algorithme efficace permettant à l'ordinateur de jouer un ou plusieurs joueurs. Pour cet objectif, le joueur géré par l'ordinateur se déplace aléatoirement

## Objectif stratégie tricky

Faire se déplacer le serpent de l'ordinateur à l'aide d'un algorithme plus élaboré, qui lui fera viser des bonus ou tenter de bloquer son adversaire.

# Votre travail

## PENDANT LE DÉROULEMENT DU PROJET

- Code python pas nécessairement objet. Utilisation de tkiteasy (tkinter : à vos risques et périls).
- Travail en **binôme obligatoire** (binôme=trinôme=quadrinôme...).
- Choisissez vos objectifs. La note maximale ira aux projets qui valident tous les objectifs.
- Toutes les fonctions doivent recevoir leurs paramètres et renvoyer une valeur résultat.  
**VARIABLES GLOBALES INTERDITES SAUF EXCEPTION MOTIVÉES.**
- Toutes les fonctions doivent être documentées et commentées :
  - Commentaire en début de fonction : quels paramètres reçoit-elle et que renvoie-t-elle ?
  - Les noms des fonctions et des variables doivent être judicieusement choisis.
  - Un commentaire expliquera les endroits cruciaux du code.

## UNE FOIS LE PROJET TERMINÉ

- Date de rendu des projets : **Samedi 29/3 21h**
- Vous déposerez votre projet **sur moodle**, sous la forme d'une **archive au format ZIP**, nommée **NOM1\_NOM2.zip** où NOM1 et NOM2 sont les noms des deux membres du binôme. L'archive doit être débarrassée des fichiers inutiles : vous n'y placerez que les fichiers .py et ceux dont vous avez besoin (images, sons, plans...), en respectant votre arborescence.  
**ATTENTION : le non-respect précis de ces consignes (mauvais nommage, mauvais format de compression, dépôt en retard) entraîne une diminution de la note finale.**
- Le projet doit être fonctionnel **SUR LES MACHINES DE L'UNIVERSITE**. À vous de tester votre projet sur ces machines avant de le rendre.
- Dans votre archive, vous joindrez un document **README : un fichier texte (txt ou PDF)** indiquant succinctement (une page) le travail accompli, les problèmes rencontrés, les bugs connus, les idées originales...

# Évaluation

**Des soutenances en binôme auront lieu à la fin du module.** Le respect des deux objectifs obligatoires vous garantit une note entre 10 et 12.

Les **compétences** qui seront évaluées sur ce projet seront :

- *Respect du cahier des charges "dépôt" (deadline, README, archive nettoyée et au bon format, programme fonctionne sans erreurs)*
- *Qualité du code (structures, commentaires, nommage, organisation générale)*
- *Validation des différents objectifs*
- *Incorporer des options et solutions techniques innovantes : soyez imaginatifs !*

À l'oral, vous présenterez votre travail en étant capable de montrer que chaque membre du groupe a compris le fonctionnement général du programme. La note finale pourra être modulée en cas d'investissements inégaux des membres du groupe, ou bien de plagiat avéré.

# Et maintenant ?

Vous pouvez commencer votre projet dès maintenant, en respectant les priorités suivantes :

- Trouver un.e binôme
- **Avant de coder, réfléchir ensemble aux structures de données à mettre en œuvre**
- Se répartir le travail en vous mettant d'accord sur les spécifications de ce que chacun réalise

**Un canal dédié au projet a été créé sur Discord.** Bon travail ! Amusez-vous bien...