



# v1.1 **ProjetCoconut partyY**

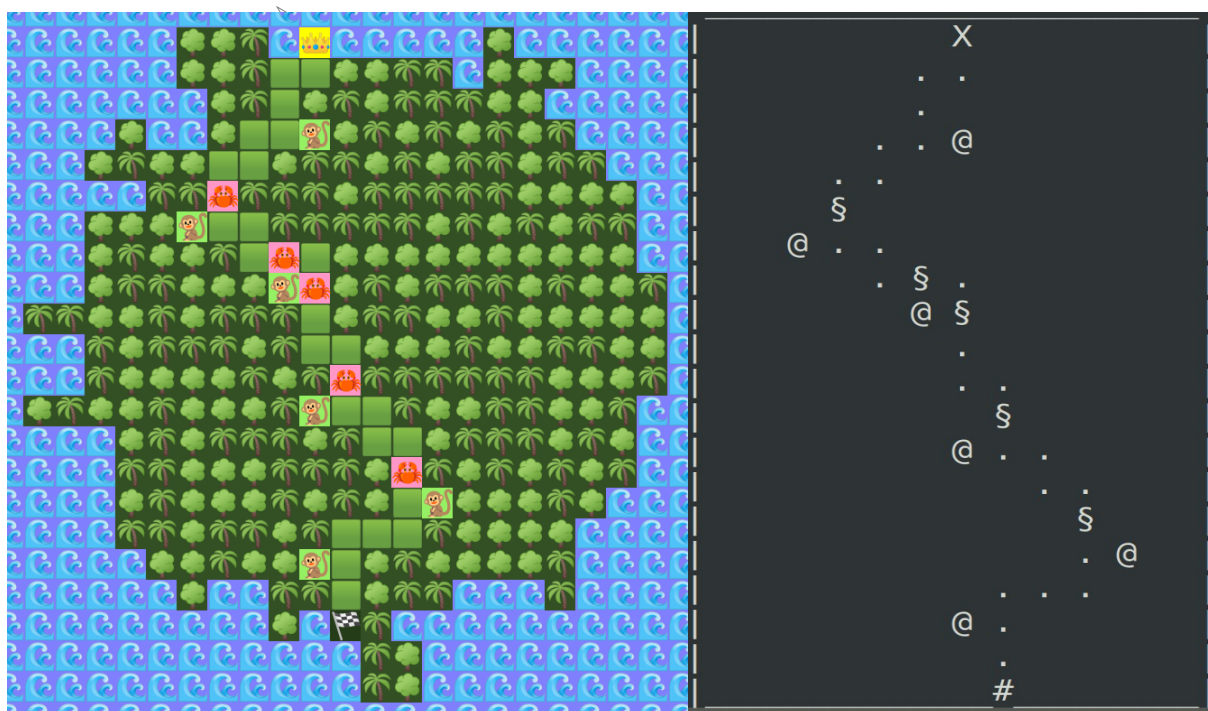
FILIERE préING1 • 2024-2025

AUTEURS E.ANSERMIN – R. GRIGNON

E-MAILS [eva.ansermin@cyu.fr](mailto:eva.ansermin@cyu.fr) – [romuald.grignon@cyu.fr](mailto:romuald.grignon@cyu.fr)

## DESCRIPTION DU PROJET

- Ce projet est un petit jeu vidéo du type 'Tower-Defense' dans lequel des singes défendent la couronne de leur roi contre des crabes cupides qui cherchent à s'en emparer.
- Le jeu se déroule dans une matrice à 2 dimensions, dans laquelle chaque case représente un seul élément du jeu à la fois (case de départ, crabe, singe, chemin des crabes, couronne, case vide, ...).
- Voici deux exemples d'affichages possibles de ce type de jeu **dans un terminal** : (stylisé avec emojis à gauche / ASCII simple à droite).  
Les **cases** du chemin vert ('.') mènent du point de **départ** ('#') vers la **couronne** ('X') en passant à côté des singes ('@').  
Des **crabes** ('\$') sont représentés **sur** certaines cases du chemin.



- Le but est donc de positionner des singes (et de les améliorer) pour lutter contre des vagues de crabes successives. Le critère de fin de jeu peut être de faire le maximum de score en résistant aux différentes vagues d'attaques des crabes.

## INFORMATIONS GENERALES

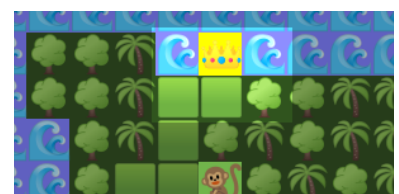
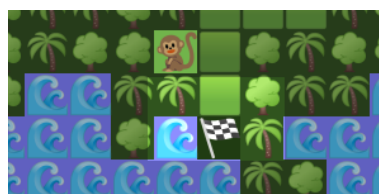
- **Taille de l'équipe**  
Ce projet est un travail d'équipe. Il est autorisé de se réunir en groupe de 3 au maximum. Si le nombre total d'étudiants n'est pas un multiple de 3 et/ou si des étudiants n'arrivent pas à constituer des groupes, c'est au chargé de TD de statuer sur la formation des groupes. Pensez donc à anticiper la constitution de vos groupes pour éviter des décisions malheureuses.
- **Démarrage du projet**  
Vous obtiendrez de plus amples informations quant aux dates précises de rendu, de soutenance, les critères d'évaluation, le contenu du livrable, ..., quand le projet démarrera officiellement.
- **Dépôt de code**  
Vous devrez déposer la totalité des fichiers de votre projet sur un dépôt central Git. Il en existe plusieurs disponibles gratuitement sur des sites web comme [github.com](https://github.com) ou [gitlab.com](https://gitlab.com).
- **Rapport du projet**  
Un rapport écrit est requis, contenant une brève description de l'équipe et du sujet. Il décrira les différents problèmes rencontrés, les solutions apportées et les résultats. L'idée principale est de montrer comment l'équipe s'est organisée, et quel était le flux de travail appliqué pour atteindre les objectifs du cahier des charges. Le rapport du projet peut être rédigé en français.
- **Démonstration**  
Le jour de la présentation de votre projet, votre code sera exécuté sur la machine de votre chargé(e) de TD. La version utilisée sera la **dernière** fournie sur le dépôt Git **avant** la date de rendu. Même si vous avez une nouvelle version qui corrige des erreurs ou implémente de nouvelles fonctionnalités le jour de la démonstration, c'est bien la version du rendu qui sera utilisée.
- **Variantes**  
Lors de votre choix de projet, votre chargé(e) de TD vous donnera la ou les variantes que vous devrez implémenter.
- **Organisation**  
Votre projet complet devrait (**dans l'idéal**) être stocké sur un **dépôt git** (ou un outil similaire) tout au long du projet pour au moins trois raisons : éviter de perdre du travail tout au long du développement de votre application, être capable de travailler efficacement en équipe, et partager vos progrès de développement facilement avec votre chargé de projet. De plus il est **recommandé** de mettre en place un **environnement** de travail en **équipe** en utilisant divers outils pour cela (Slack, Trello, Discord, ...)

## CRITERES GENERAUX

- Le **but principal** du projet est de fournir une **application fonctionnelle** pour l'utilisateur. Le programme doit correspondre à la description en début de document et implémenter toutes les fonctionnalités listées.
- Votre code sera généreusement **commenté**.
- Tous les éléments de **votre code** (variables, fonctions, commentaires) seront écrits dans la **même langue**. Langue anglaise conseillée mais pas obligatoire.
- Votre application ne doit jamais s'interrompre de manière intempestive (crash), ou tourner en boucle indéfiniment, quelle que soit la raison. Toutes les erreurs doivent être gérées correctement. Il est préférable de d'avoir une application stable avec moins de fonctionnalités plutôt qu'une application contenant toutes les exigences du cahier des charges mais qui plante trop souvent. Une application qui se stoppe de manière imprévue à cause d'une erreur de segmentation ou d'une exception, par exemple, sera un événement très pénalisant.
- Votre application devra être **modulée** afin de ne pas avoir l'ensemble du code dans un seul et même fichier par exemple. Apportez du soin à la conception de votre projet avant de vous lancer dans le code.
- Le livrable fourni à votre chargé(e) de TD sera simplement l'**URL** de votre **dépôt Git** accessible **publiquement**. Même si vous n'avez pas utilisé ce dépôt régulièrement au cours du projet, le code final sera livré dessus.

## FONCTIONNALITES DU PROJET

- Ce jeu se déroule sur une matrice 2D qui nécessite d'être remplie avec les différents éléments du jeu. Il est possible de créer cette carte depuis **un fichier** qui contiendrait tous les éléments, ou bien développer un code qui va **générer** de manière **procédurale** le chemin, les positions des singes, etc.
- La carte doit être de taille variable (soit entrée par l'utilisateur, soit fixée aléatoirement au lancement du programme). Elle doit contenir un chemin passant au travers de la carte (soit verticalement, soit horizontalement, ou encore en diagonale, ...)
- Dans un premier temps le chemin peut être rectiligne pour simplifier le développement, mais il faut tenter de produire un chemin serpentant un minimum pour le livrable final.
- Plusieurs phases sont donc nécessaires pour ce jeu :
  - **(#0) menu de démarrage** demandant de démarrer une nouvelle partie, de reprendre une ancienne partie sauvegardée (si elle existe), ou de quitter le jeu.
  - **(#1) Génération de la carte** de jeu avec les différents éléments de base positionnés (chemin, départ, fin, positions possibles des singes). Aucun singe n'est présent au début.





Les phases de déplacement (#3a) et de tir (#3b) se déroulent en boucle tant qu'il reste des crabes ou que la condition de fin (#4) n'a pas encore été atteinte.

- **(#3c) Affichage.** Après les phases (#3a) et (#3b), le programme réaffiche la carte du jeu pour que l'on puisse visualiser l'avancement, et la destruction des crabes, case par case.

Cet affichage n'est pas obligatoirement dynamique, vous pouvez afficher les étapes les unes en dessous des autres sur le terminal. Néanmoins, il est possible d'effacer le terminal et de réafficher par dessus ce qui est déjà présent à l'écran. Un lien est donné en ce sens pour piloter le terminal.

- **(#4) Gestion de fin de jeu :** si un crabe atteint la couronne, il vous faut déterminer la mécanique de jeu voulue : fin du jeu, perte d'un point de vie du roi des singes, ... .

Si votre mécanique détermine que le jeu est perdu, alors la vague de crabes est stoppée et un message affiche la fin du jeu avec un score ou un nombre de crabes éliminés, ou le nombre de vagues parcourues, ... . Ensuite le jeu repasse au menu de départ.

Si la fin du jeu n'est pas encore prononcée (ex : plusieurs points de vie pour le roi et il n'en n'a pas encore perdu la totalité) et qu'il n'y a plus de crabes sur le chemin alors le jeu repasse sur la phase (#2) et ce, jusqu'à la fin du jeu.

- **(#5) Sauvegarde.** Lors de la phase (#2), il est possible de stopper la partie, la sauvegarder en l'état, pour la reprendre plus tard. Si l'utilisateur choisit cette option, il faudra sauvegarder en mémoire l'ensemble du modèle de données représentant votre jeu (contenu de la carte, position et points de vies des différents éléments, ...) et ensuite revenir au menu de départ.

L'option du menu de départ « restaurer » devra alors pouvoir restaurer en mémoire la partie et la reprendre sans bug.

Si vous pensez ne pas avoir le temps d'implémenter la fonctionnalité de sauvegarde + restauration, il est préférable de ne faire la sauvegarde qu'au format ASCII, plus verbeuse en terme de code, mais qui permettra de montrer que toutes les données nécessaires à la reconstitution du jeu sont bien sauvegardées dans le fichier et ainsi marquer des points même si la restauration ne fonctionne pas.

.....

## VARIANTES

- **Direction** de déplacement :
  - variante **DIR\_UP** : la case de départ est située en bas, et la couronne en haut de la carte.
  - variante **DIR\_DOWN** : la case de départ est située en haut, et la couronne en bas de la carte.
  - variante **DIR\_LEFT** : la case de départ est située à droite, et la couronne à gauche de la carte.
  - variante **DIR\_RIGHT** : la case de départ est située à gauche, et la couronne à droite de la carte.
- **Dimensions** de la carte (pour le cas d'une génération procédurale de la carte. Si la carte provient d'un fichier contenant la carte entière, cette variante servira à choisir le nom/numéro du fichier plutôt que les dimensions) :
  - variante **DIM\_USER** : les dimensions initiales de la carte sont entrées par l'utilisateur.
  - variante **DIM\_RAND** : les dimensions initiales de la carte sont des valeurs générées aléatoirement.
- **Conditions de fin** :
  - variante **END\_ONE** : la partie se termine lorsqu'un crabe atteint la couronne du roi des singes.
  - variante **END\_RAND\_MULTI** : la partie se termine lorsque le roi des singes n'a plus de points de vie. Chaque crabe qui atteint la couronne fait perdre 1 point de vie. Le nombre de points de vie est fixé aléatoirement en début de partie.
  - variante **END\_USER\_MULTI** : la partie se termine lorsque le roi des singes n'a plus de points de vie. Chaque crabe qui atteint la couronne fait perdre 1 point de vie. Le nombre de points de vie est fixé par l'utilisateur en début de partie.

## RESSOURCES UTILES

- **Github**  
<https://www.github.com>  
<https://docs.github.com/en/get-started/quickstart/hello-world>
- **Couleurs dans le terminal**  
<http://sdz.tdct.org/sdz/des-couleurs-dans-la-console-linux.html>
- **Emojis dans le terminal**  
<https://unicode.org/emoji/charts/full-emoji-list.html>
- **Symboles de bordures dans le terminal**  
<https://www.w3.org/TR/xml-entity-names/025.html>
- **Pilotage du Terminal**  
[https://en.wikipedia.org/wiki/ANSI\\_escape\\_code#Control\\_Sequence\\_Introducer\\_commands](https://en.wikipedia.org/wiki/ANSI_escape_code#Control_Sequence_Introducer_commands)