Rapport de projet :

Hangman

Elaboré par :

-Walid Mouahidi

-Anass Ajja

-Anas Rayk

-Rayane Chihame

Soutenu le 23/01/2023

REMERCIEMENTS

C’est avec un grand plaisir, que nous adressons nos sincères remerciements à l’égard de notre enseignant Mr FAWZI Badr, qui nous a accompagné pour la bonne réussite de ce travail.

Merci aux membres du jury pour l’attention qu’ils ont bien voulu accorder au travail et pour le temps consacré à son évolution.

Merci à tous nos camarades qui nous ont aidé pour collecter certaines données.

Nous ne terminerons pas sans avoir remercié toutes les personnes ayant contribué à la réalisation de ce projet.

Sommaire

1. Introduction

- Description générale du jeu du pendu

-Présentation de l’équipe

- Présentation du sujet et de son objectif

1. Analyse des besoins

- Identification des fonctionnalités requises

- Evaluation des contraintes et des limites

1. Conception

- Réalisation de l’algorigramme

- Mise en place des tâches sur Trello

1. Développement du code

-Bibliothèques utilisées

- Réalisation du jeu de base

- Réalisation des bonus

1. Evaluation

* Test du code

1. Conclusion

* Bilan
* Perspectives d’évolution

Introduction générale

Le jeu du pendu est un jeu de devinettes où un joueur doit deviner un mot ou une phrase en proposant des lettres. Le jeu se joue généralement à deux : un joueur pense à un mot ou une phrase et l'autre joueur doit deviner de quel mot il s'agit en proposant des lettres.

A chaque tour, le joueur qui doit deviner propose une lettre. Si la lettre est contenue dans le mot ou la phrase, elle est révélée à sa place dans le mot ou la phrase. Si la lettre n'est pas contenue dans le mot ou la phrase, un élément d'un dessin d'un pendu est ajouté. Le jeu se termine soit quand le joueur qui doit deviner a trouvé le mot ou la phrase, soit quand le dessin du pendu est complet. Le joueur qui a trouvé le mot ou la phrase gagne, et le joueur qui a fait le dessin du pendu perd.

Il existe des variantes de jeu où le joueur qui doit deviner doit trouver plusieurs mots ou phrases en même temps, ou où il y a plusieurs joueurs qui doivent deviner. Il existe également des versions électroniques du jeu, où un programme informatique gère les tours de jeu et le dessin du pendu.

Voici donc l’objectif de notre projet : créer un programme permettant à l’utilisateur de jouer au jeu du pendu qui gère également les tours de jeu et le dessin du jeu.

Présentation de l’équipe

Chef de projet : Walid Mouahidi

Responsable technique : Anas Ajja

Responsable communication : Anas Rayk

Responsable design : Rayane Chihame

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Taches | Mouahidi | AJJA | Rayk | Chihame |
| Algorigramme | X | X | X | X |
| Communication | X |  | X |  |
| Design |  | X | X |  |
| drawHangman |  |  | X |  |
| Build-In-coloRed |  | X | X |  |
| Build-In-colorGreen |  | X | X |  |
| Build-In-coloRed |  | X | X |  |
| checkWin |  |  |  | X |
| Replay |  |  |  | X |
| gameplayX | X | X |  |  |
| gameover | X |  |  | X |
| Main | X | X | X | X |
| Powerpoint | X | X |  |  |
| Rapport | X |  |  |  |

Présentation du sujet et de ses objectifs

Nous nous retrouvons pour la création d’un jeu du pendu en utilisant le langage de programmation Go. L'objectif de ce projet serait de développer une application fonctionnelle de pendu qui permette aux utilisateurs de deviner des mots en utilisant des lettres proposées. Le but serait de créer une interface utilisateur conviviale et de mettre en œuvre des algorithmes efficaces pour gérer les fonctionnalités du jeu, comme la gestion des erreurs et la détermination de la victoire ou de la défaite.

Analyse des besoins

1 - Fonctionnalités nécessaires :

Génération aléatoire d'un mot ou d'une phrase à deviner : Il est nécessaire de disposer d'un mécanisme pour générer aléatoirement un mot ou une phrase à deviner pour chaque partie.

Stockage des mots et phrases à deviner : Il est nécessaire de disposer d'une base de données ou d'un fichier pour stocker les mots et phrases qui peuvent être utilisés pour chaque partie.

Saisie des lettres proposées par le joueur : Il faut un mécanisme pour permettre au joueur de saisir les lettres qu'il propose comme solutions.

Vérification de la validité des lettres proposées : Il faut un mécanisme pour vérifier si les lettres proposées par le joueur sont valides, c'est à dire si elles appartiennent à l'alphabet.

Comparaison des lettres proposées avec le mot ou la phrase à deviner : Il est nécessaire d'avoir un mécanisme pour comparer les lettres proposées par le joueur avec le mot ou la phrase à deviner, pour déterminer si une lettre est correcte ou non.

Affichage des lettres correctes : Il faut un mécanisme pour afficher les lettres correctes dans le mot ou la phrase à deviner, à leur place respective.

Affichage du dessin du pendu : Il est nécessaire d'avoir un mécanisme pour afficher le dessin du pendu, qui évolue à chaque tour de jeu en fonction du nombre de lettres incorrectes proposées par le joueur.

Compteur des erreurs : Il est nécessaire d'avoir un mécanisme pour compter le nombre d'erreurs commises -par le joueur et afficher le score

Gestion de fin de partie : Il est nécessaire de disposer d'un mécanisme pour détecter les fins de parties, et afficher un message de victoire ou de défaite en conséquence.

2 - Limites et Contraintes

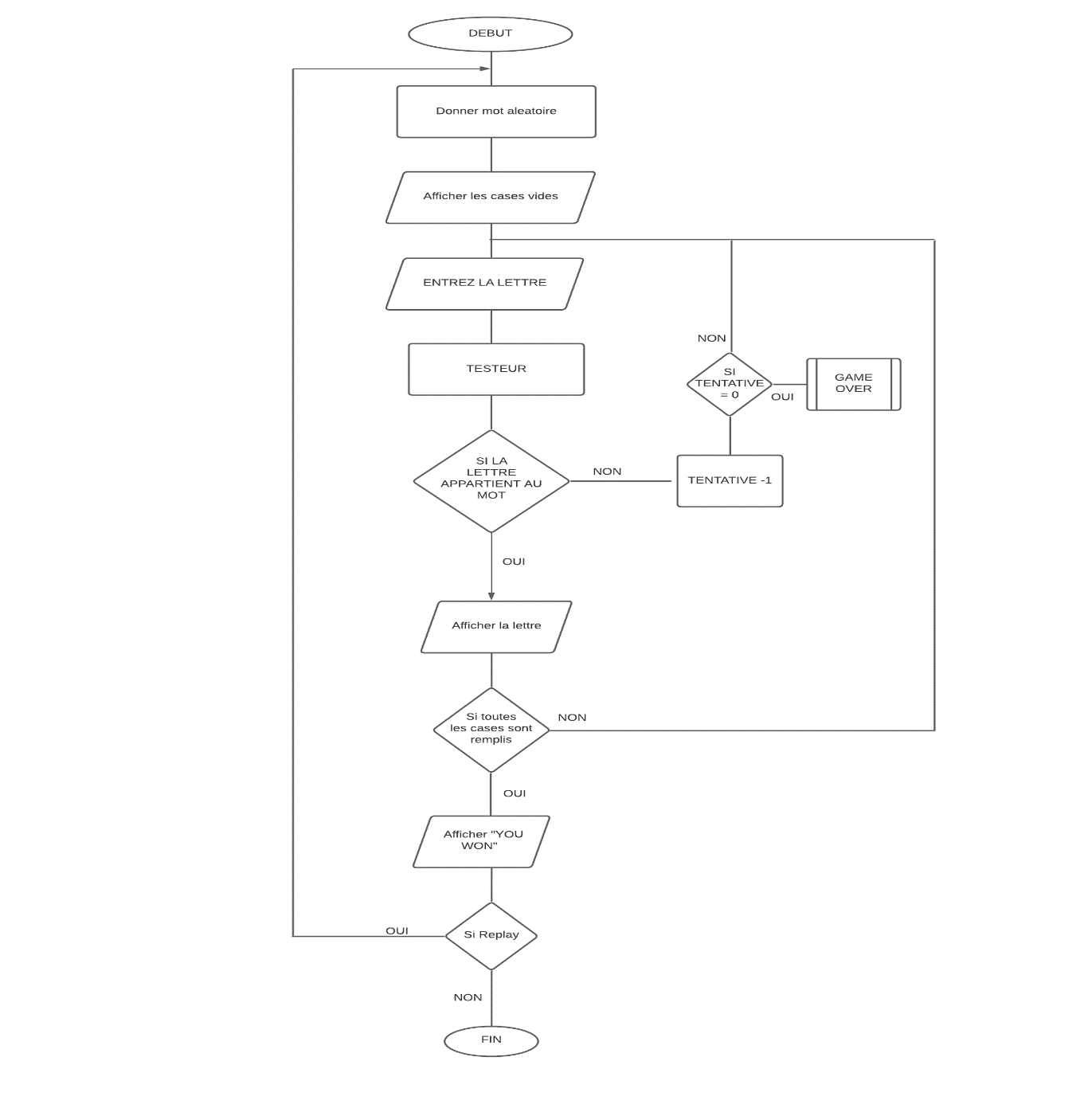
Connaissance du langage Go : Il est nécessaire d'avoir une bonne connaissance de Go pour réaliser ce projet efficacement. Il est possible de devoir consacrer du temps pour approfondir nos compétences en langage Go.

Temps de développement : Le développement d'un jeu peut prendre beaucoup de temps, il est donc important de planifier adéquatement le temps de développement pour être en mesure de respecter les délais impartis.

Dictionnaire de mots : Il serait nécessaire de disposer d'un dictionnaire de mots pour le jeu, cela pourrait prendre du temps pour le construire ou le trouver et pourrait entraîner des coûts supplémentaires.

Stabilité et performance : Il est important que le jeu soit stable et performant pour une bonne expérience utilisateur. Cela pourrait nécessiter des tests et des optimisations pour assurer la qualité du jeu.

Conception

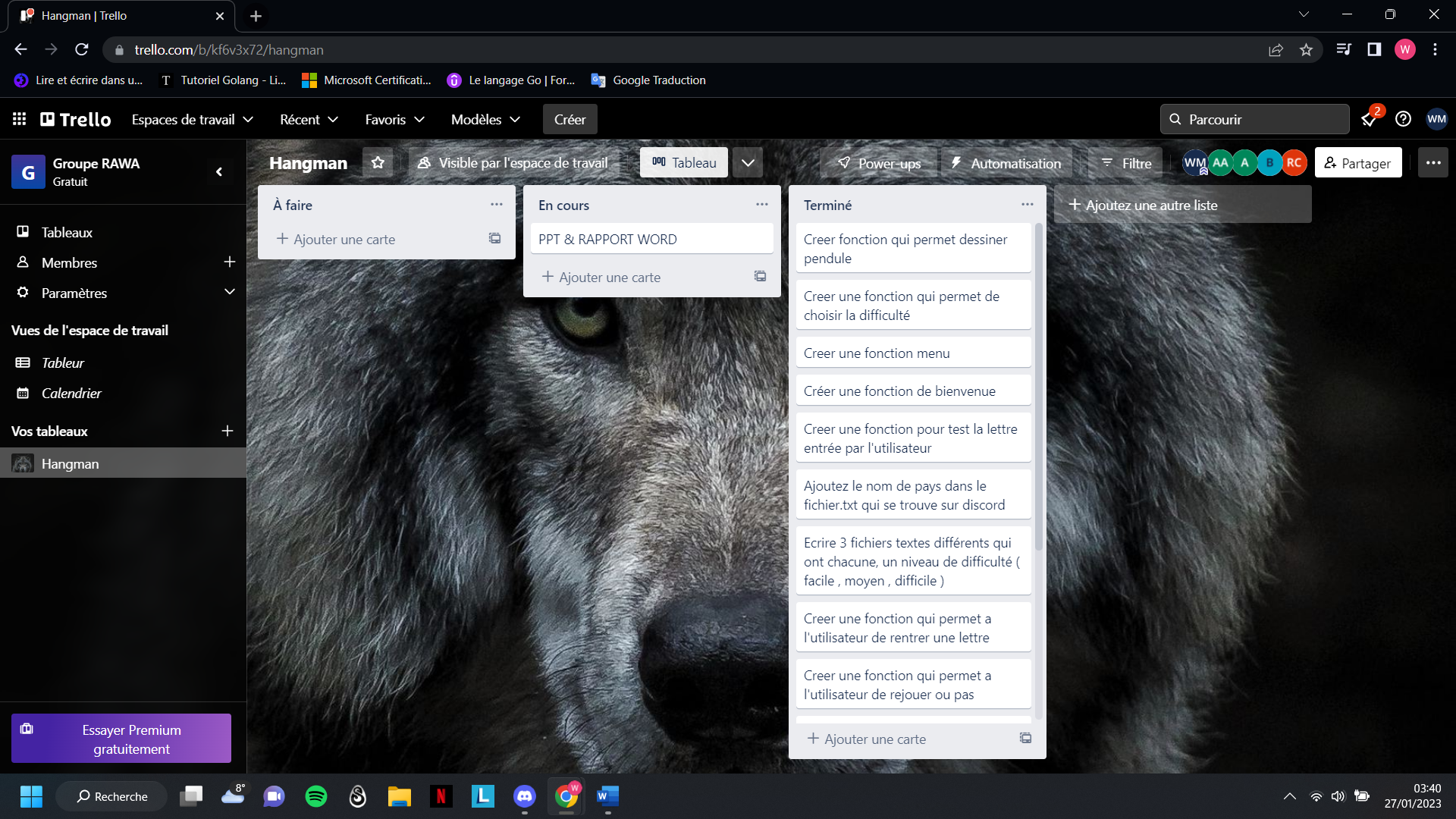


Voici l’algorigramme que nous avons suivi lors de la réalisation du programme.

Une image contenant texte, intérieur, noir, capture d’écran

Description générée automatiquement

Voici une capture d’écran de le plateforme Trello en début de projet



Voici une capture d’écran de le plateforme Trello en fin de projet.

Développement du code

1. Bibliothèques utilisées

Une image contenant texte

Description générée automatiquement

Liste des bibliothèques utilisées :

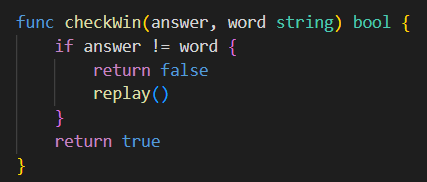
* Bufio : package utilisé pour les E/S tamponées
* Fmt : package utilisé pour l’affichage du pendu et le stockage de variable principalement
* Math/rand : utilisé pour générer des nombres aléatoires
* Os : "os" en Go est utilisée pour accéder aux fonctionnalités système d'exploitation
* Strings : "strings" en Go est utilisée pour manipuler les chaînes de caractères
* Time : "time" en Go est utilisée pour manipuler les dates et les heures

1. Réalisation du jeu de base
2. Voici la fonction qui permet l’affichage du pendu en fonction du score de l’utilisateur grâce à la fonction switch avec « chances » comme paramètres qui représente le nombre de tentatives.

Une image contenant texte, moniteur

Description générée automatiquement

1. Voici la fonction qui permet de vérifier si l’utilisateur à compléter le mot ou pas à l’aide de if avec comme paramètres d’entrées answer et word. En retour, le programme nous répond true si le mot est complet et false si le mot n’est pas le complet.



1. Voici la fonction qui permet à l’utilisateur de choisir la difficulté. Dépendamment de la réponse, le programme choisira ensuite l’un des fichiers texte dans lesquels nous avons préalablement noter plusieurs mots en fonction de leur longueur. En effet, la difficulté sera relative à la longueur du mot. Ensuite, le programme piochera un mot aléatoirement grâce à la fonction rand.seed. Ensuite le programme demandera à l’utilisateur de rentrer une lettre. Puis, il va tester la lettre et en fonction de la réponse, il va afficher le mot caché avec les lettres trouvées. Pour finir, il va aussi tester si l’utilisateur a encore des tentatives ou pas. S’il n’en a plus, le programme appelle une autre fonction «gameover ».

Une image contenant texte

Description générée automatiquement

Une image contenant texte, écran, capture d’écran

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

1. Voici la fonction qui permet à l’utilisateur de choisir si il rejoue ou pas. Le programme nous repondra oui si l’utilisateur veut rejouer et false si il ne veut pas rejouer. Tout depend de ce qu’il a stocké dans la variable response.

Une image contenant texte

Description générée automatiquement

1. Voici la fonction qui permet l’affichage du message de fin de partie lorsque l’utilisateur perd la partie.

Une image contenant texte

Description générée automatiquement

1. Voici la fonction qui nous permet l’affichage du menu et qui affiche le message de bienvenue à l’utilisateur. Puis il lui demande de choisir la difficulté puis le thème pour que l’utilisateur puisse recevoir le mot.

Une image contenant texte

Description générée automatiquement

Une image contenant texte, périphérique, mètre, jauge

Description générée automatiquement

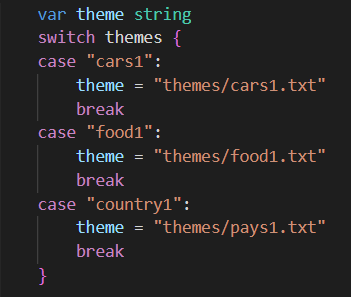
1. Réalisation des bonus

Personnalisation des couleurs : Nous avons changé la couleur du pendu lorsqu’il s’affiche et lorsque l’utilisateur perd(rouge) ou gagne(vert).

Une image contenant texte

Description générée automatiquement

Personnalisation des thèmes : Il est possible de permettre aux joueurs de choisir différents thèmes avec le switch qui a pour paramètres « themes » pour les images et les mots ou phrases à deviner.



Ajout d'un système de difficulté : Il est possible d'ajouter des niveaux de difficultés pour les mots ou phrases à deviner.Une image contenant texte

Description générée automatiquement Une image contenant texte

Description générée automatiquement

Ajout d’un score variable pour l’utilisateur : L’utilisateur commence a avoir un compteur de points de l’exécution du programme à sa fin où il gagne 20 points pour chaque lettre trouvé.

Une image contenant texte

Description générée automatiquement

Voici les tous les bonus que nous avons rajouté lors de notre projet.

Evaluation

1. Test du code

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Voici les deux cas dans lesquels l’utilisateur gagne et perd la partie.

Conclusion

1. Bilan

* Objectifs :

- Développer nos skills en programmation

- Mieux travailler en groupe

- Avoir une meilleure gestion de projet

* Méthodologie :

- Mise en place de l’algorigramme

- Mise en place des fonctions et des tests

* Résultats :

- Bon fonctionnement du jeu

- Ajout de bonus : Build-In-ColoRed, Build-In-ColoReset, Build-In-colorGreen

* Difficultés rencontrées :

- Découverte de nouvelles bibliothèques :bufio,math/rand

- Respecter le temps

- Utilisation de plusieurs fonctions

- Affichage du score en temps réel : écrire le mot avec les lettres cachées pour chaque étape de la partie

* Conclusion :

- Programme efficace.

- Pas de problèmes ou de bug au niveau de l’exécution du programme.

1. Perspectives d’évolution
   * Ajout de fonctionnalités : Il est possible d'ajouter des fonctionnalités supplémentaires au jeu, telles que la possibilité de sauvegarder et de charger des parties, de jouer en ligne contre d'autres joueurs, etc.
   * Amélioration de l'interface utilisateur : Il est possible d'améliorer l'expérience utilisateur en utilisant des bibliothèques graphiques pour créer une interface plus attrayante et facile à utiliser.
   * Ajout de la traduction : Il est possible de traduire le jeu dans plusieurs langues pour rendre le jeu accessible à un public plus large.
   * Mode multijoueur : Il est possible de rajouter la possibilité de jouer avec un autre joueur pour rendre le jeu encore mieux.