CEDRIC GARAND  
Technologies Émergentes  
420-5A3-VI gr. 01

PROJET DE SESSION  
Démineur

Travail présenté à  
Étienne Rivard

Département des Techniques de l’informatique  
Cégep de Victoriaville  
Le 20 décembre 2023

Table des matières

[Buts du projet 3](#_Toc153208908)

[Liste de concepts 3](#_Toc153208909)

[Références 3](#_Toc153208910)

[Cartographie 3](#_Toc153208911)

[Tableau comparatif 4](#_Toc153208912)

[Journal des enjeux et solutions 5](#_Toc153208913)

[Fiche 1 : 5](#_Toc153208914)

[Fiche 2 : 5](#_Toc153208915)

[Fiche 3 : 7](#_Toc153208916)

[Conclusion 7](#_Toc153208917)

# Buts du projet

Le but principal de mon projet était de faire un jeu de démineur fonctionnel en python en utilisant la bibliothèque pygame. Le but était de faire en sorte que les mécaniques de mon jeu ressemblent le plus possible aux mécaniques du vrai jeu de démineur. Ensuite, je voulais que le jeu puisse être recommencé un fois la partie terminée. J’ai réussi à terminer ce premier but. Je me suis donc lancé un deuxième but puisqu’il me restait encore du temps pour avancer mon projet. Ce deuxième but était de faire une intelligence artificielle qui reçoit les mêmes informations à propos de la grille de jeu qu’un humain et qui suit une série de règles prédéfini pour faire un coup.

# Liste de concepts

Python : Le jeu est fait dans le langage python

Pygame : C’est la bibliothèque utilisée pour créer le jeu

Algorithmie : L’algorithmie est très présente dans la conception de ce jeu

Affichage visuel : L’affichage de la grille et des icones est aussi un concept important

Entrés utilisateurs : Les entrés effectués par l’utilisateur pour interagir avec le jeu

# Références

1. Documentation officielle de la bibliothèque pygame : <https://www.pygame.org/docs/>
2. Un exemple de projet avec pygame pour comprendre le fonctionnement général de la bibliothèque : <https://realpython.com/pygame-a-primer/>
3. Comment créer des boutons dans pygame : <https://www.geeksforgeeks.org/how-to-create-buttons-in-a-game-using-pygame/>
4. Comment mettre des images (pour les drapeaux et les bombes) : <https://www.geeksforgeeks.org/python-display-images-with-pygame/>
5. Comment avoir plus qu’un USEREVENT dans pygame : <https://stackoverflow.com/questions/23571956/pygame-way-to-create-more-userevent-type-events>

# Cartographie

Voici une cartographie des principaux éléments dans un jeu de démineur.

Une image contenant texte, capture d’écran

Description générée automatiquement

# Tableau comparatif

Une image contenant texte, ligne, Police, nombre

Description générée automatiquement

Comme on peut le voir avec cette matrice de décision, la meilleure bibliothèque python pour créer un jeu de démineur est la bibliothèque pygame puisqu’elle à obtenu le plus haut total.

# Journal des enjeux et solutions

## Fiche 1 :

Lors de la programmation de différentes grandeurs de grille de jeu, j’ai rencontré un problème pour trouver les index autour d’une case. Code initial : 

Ce tableau contient les additions à faire pour obtenir les index des cases autour d’une case. J’ai bâti la mécanique de jeu sur une grille de jeu 10x10 donc ce tableau fonctionnait pour cette taille de grille.

Cependant, lorsque j’ai essayé de faire des grilles de jeu de différentes tailles, le tableau d’addition ne contenait pas les bonnes additions pour obtenir les index des cases autour d’une case. Cela est dû au fait que si je veux avoir l’index de la case à la droite de la case à l’index 34, je fais juste ajouter 10 pour l’obtenir dans une grille 10x10. Mais, dans une grille 14x14, il faut que j’ajoute 14 au lieu de 10 pour avoir l’index de la case à la droite.

Pour résoudre ce problème, j’ai fait le même tableau qu’au début, mais j’ai fait en sorte de mettre 10 (la longueur de la grille) dans le calcul des additions.



Ensuite, j’ai remplacé le 10 par une variable qui contient la longueur de la grille. De cette façon, peu importe la taille de la grille, les additions donnent les index des cases autour d’une case principale.



## Fiche 2 :

Un autre problème que j’ai rencontré lorsque j’essayais de créer différentes tailles de grille était de définir le nombre de cases en longueur et en largeur et la taille des cases. Si je prenais la même taille de cases pour une grille plus grande, je me retrouvais avec une grille coupée puisque le nombre de cases était trop grand pour être contenu dans l’écran.

Une image contenant texte, capture d’écran, carré

Description générée automatiquement

Pour résoudre ce problème, j’ai créé une classe *Difficulty* qui contient les informations nécessaires pour chaque niveau de difficulté. Avec cela, je crée les trois niveaux de difficulté au début du programme et lorsque le joueur choisi une difficulté, j’envoie les informations de difficulté au constructeur de la grille de jeu. De cette façon, toutes les informations pour la taille de la grille et des cases est bonne. Classe *Difficulty*:

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

## Fiche 3 :

Lors de la programmation de la troisième règle pour l’intelligence artificielle, j’ai rencontré un problème d’index non présent dans le tableau. Code initial :Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

array\_index\_square\_left\_main: Contient l’index des cases possibles autour de la case principale

array\_index\_square\_left\_neighboor: Contient l’index des cases possibles autour de la case voisine à la case principale

Dans cette règle, il faut enlever les deux index présents dans le tableau principal du tableau de la case voisines, mais parfois, je recevais cette erreur : Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

Cette erreur veut dire que les index présents dans le tableau principal ne sont pas présents dans le tableau de la case voisine.

J’ai ajouté des « print() » au code pour voir le contenu des deux tableaux au moment de l’erreur et j’ai vérifié ce qui causait l’erreur avec où se trouvait l’erreur dans l’interface graphique. Cela m’a permis de voir que la règle que j’avais implémenté avait deux configurations possibles dans la grille de jeu.

J’ai donc modifié mon code pour ne pas retourner de cases si la configuration n’était pas la configuration désirée :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

# Conclusion

En conclusion, malgré les quelques difficultés rencontrées pendant le développement de mon jeu et avec les fonctionnalités de la bibliothèque pygame, j’ai réussi à atteindre mon premier but (faire un jeu de démineur fonctionnel) assez facilement. Puis, j’ai aussi réussi à faire une très grande partie de mon deuxième but (ajouter une intelligence artificielle qui résout la grille de jeu). Le langage Python, utilisé dans le projet, est intuitif et très facile d’apprentissage. Il peut aussi être utilisé pour la programmation orienté objet ce qui lui permet de faire beaucoup de choses. Pour ce qui est de la bibliothèque de jeu pygame, j’ai trouvé ça assez facile à utiliser et ça ne m’a pas pris beaucoup de temps pour comprendre les différentes fonctions intégrées dans la bibliothèque. Cette bibliothèque à une très grand potentiel. Elle pourrait facilement être utilisée pour programmer des jeux 2D beaucoup plus complexes comme des jeux à plateformes ou des jeux de tirs. Puisque pygame est développé avec le langage Python, il serait aussi possible de faire des jeux multi-joueurs.