

Programmation événementielle :
Visual Basic .NET
Le jeu du Démineur



Table des Matières

Introduction	3
Schéma d'ordonnancement des formulaires	7
Fonctionnalités abouties	8
Conclusion	8

Introduction

Notre Projet consiste à programmer une variante du jeu du “Démineur”, le langage utilisé est Visual Basic .NET, ce projet s’inscrit dans le module Développement d’application avec IHM. Notre programme devra permettre à un utilisateur d’interagir avec un ensemble de formulaire WinForm pour jouer au jeu, avec les fonctionnalités techniques sur des modules. Il faudra donner de l’importance au respect de l’encapsulation et au respect du cahier des charges. Le fichier texte doit être au même niveau que l’exécutable

Illustration :

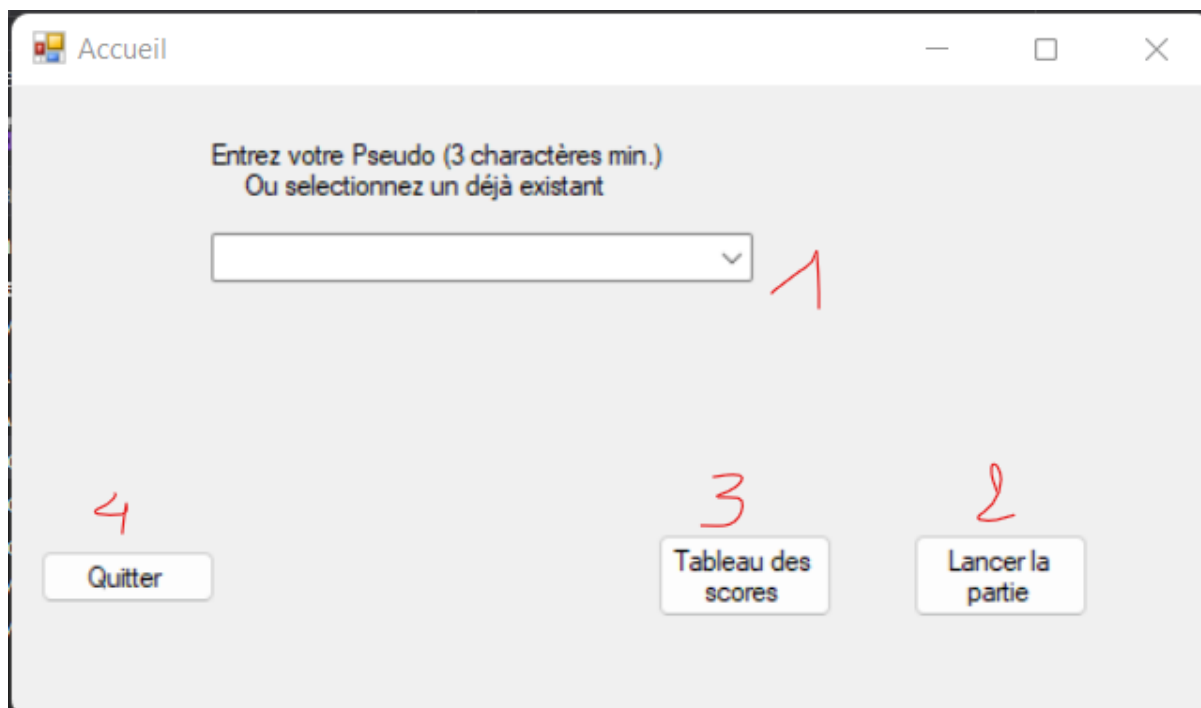


Figure 1 : accueil

Le joueur sera accueilli par le formulaire “Accueil” ci-dessus (Fig1), il pourra entrer un pseudo ou sélectionner un pseudo déjà connu pour s’identifier avec la ComboBox(1), nous avons décidé d’autoriser les lettres et chiffres (ex : BigBoss8), il faut 3 caractères minimum, l’enregistrement dans la ComboBox est triée par ordre alphabétique.

Une fois l’identification valide, il pourra appuyer sur le bouton “Lancer la partie”(2) qui lui renverra au formulaire de jeu “Partie de Démineur”.

Avec le bouton “Tableau des scores”(3) il aurait dû pouvoir voir les scores (nombre de cases démasqué, meilleur temps, etc) des anciens joueurs qui ont déjà joué par lecture de fichier texte, notre application ne le permet pas. Mais il ne pourra voir que les noms des anciens joueurs qui ont déjà joué une partie.

Le bouton “quitter”(4) lui permet de quitter l’application après une vérification par MsgBox.

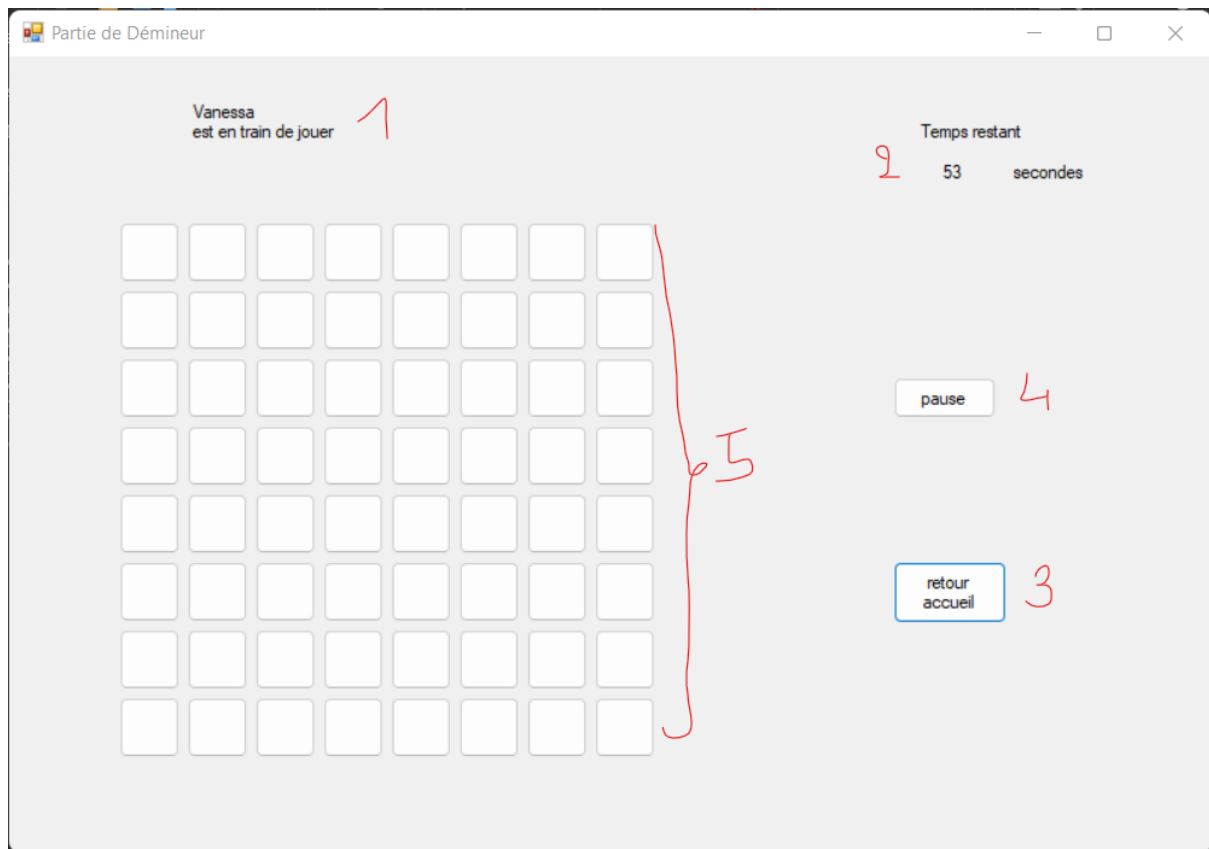


Figure 2 : Partie de Démineur

Le bouton “Lancer la partie” renvoie au formulaire “Partie de Démineur” (Fig2), un Label (1) récupère le pseudo du joueur et indique au joueur qui joue (lui).

Il y a un Label (2) correspondant à un chronomètre donnant le temps qu’il reste au joueur pour jouer sa partie, il est lié avec un outil Timer, l’indication est en seconde, on lui donne 60 pour jouer.

Le bouton “retour accueil” (3) permet d’arrêter une partie et de retourner au formulaire d’accueil avec une MsgBox pour prévenir le joueur.

Le bouton “pause” (4) permet de suspendre la partie sans la quitter, elle permet de stopper le chronomètre et de cacher la grille, ce qui permet de ne pas pouvoir jouer dessus mais également de permettre au joueur de ne pas pouvoir prévoir un coup après une pause de réflexion, non chronométré. Un nouveau bouton “reprendre” apparaît avec un label pour indiquer au joueur sa situation de pause. Sur l’appuie du bouton reprendre la grille reviendra et le chronomètre reprendra (voir Figure 2-2).

Le clique sur n’importe quel bouton provoque un événement de libération des cases et indication numérique de combien de mines entourent une case neutre. Cependant notre grille ne fonctionne pas correctement (voir Figure 2-3)

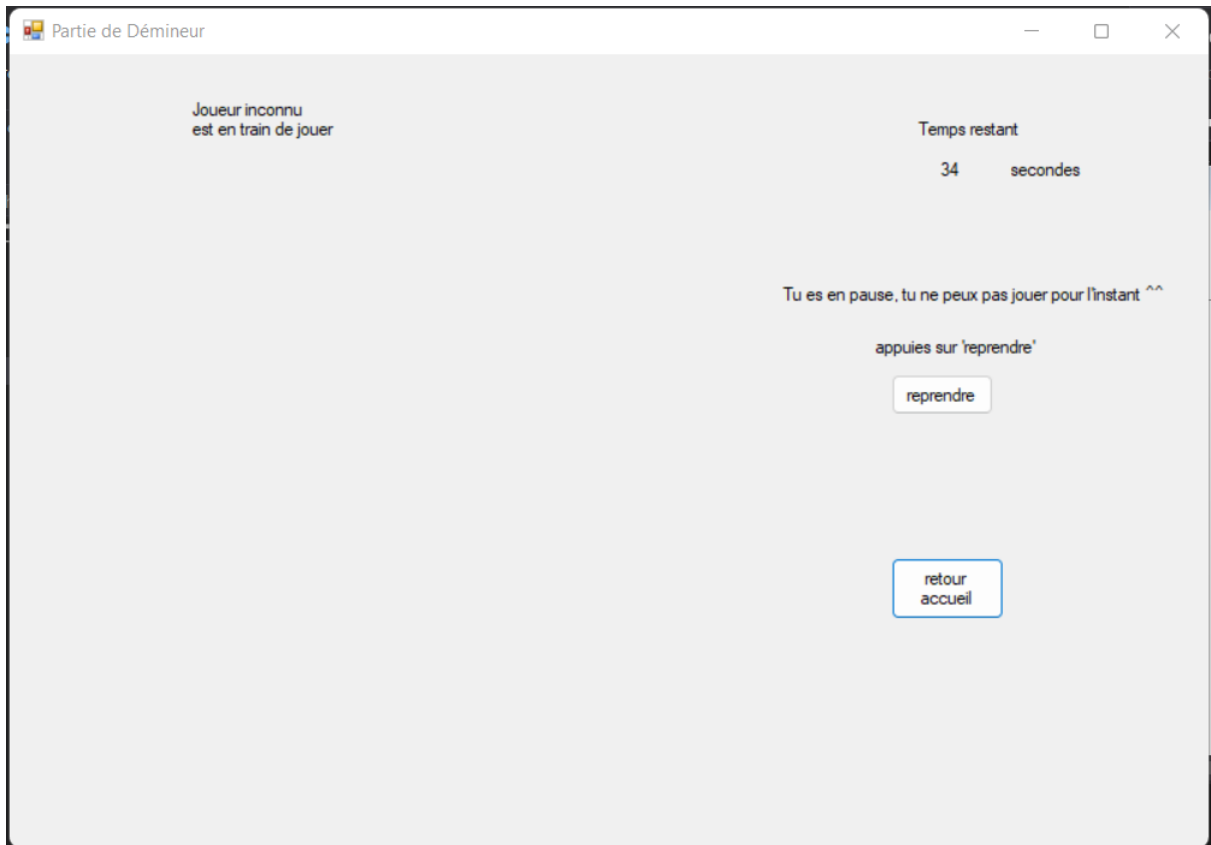


Figure 2-2 : Démineur en pause

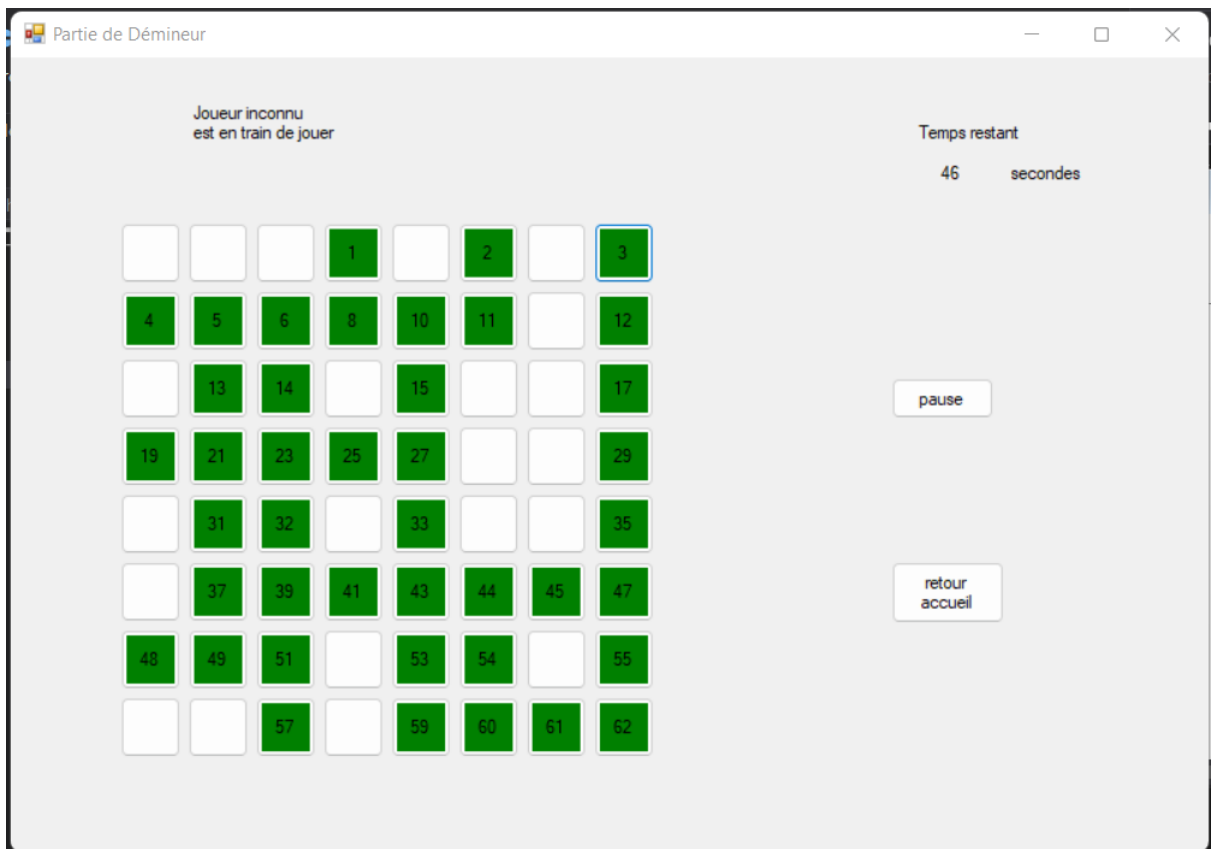


Figure 2-3 : Démineur, événement grille

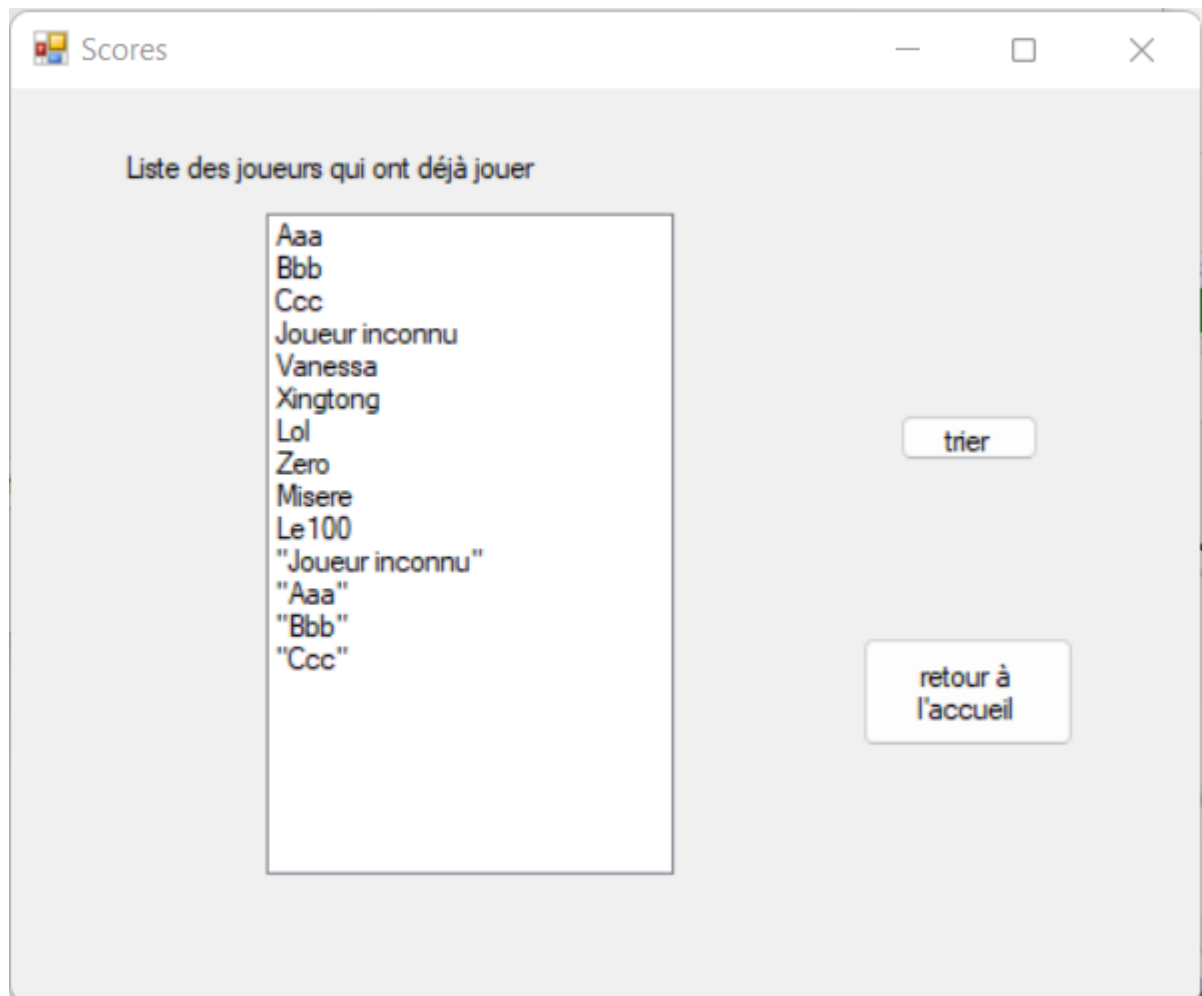
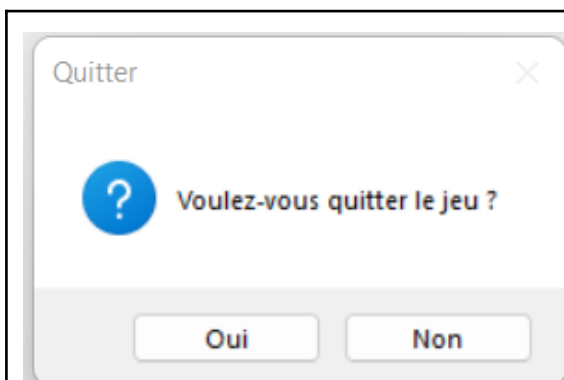


Figure 3 Scores

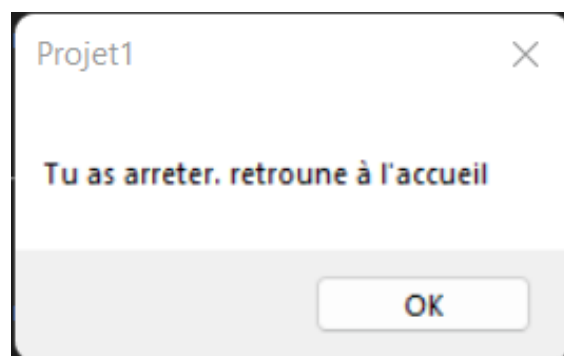
Le bouton "Tableau des scores" renvoie au formulaire "Scores", elle affiche dans une ListBox le nom des anciens joueurs qui ont déjà démarré au moins une partie. Ses éléments viennent de la ComboBox et de la lecture d'un fichier texte.

Le bouton "retour à l'accueil" fait revenir au formulaire "Accueil".

Le bouton "trier" aurait dû trier par ordre inverse de l'alphabétique (incorrect).

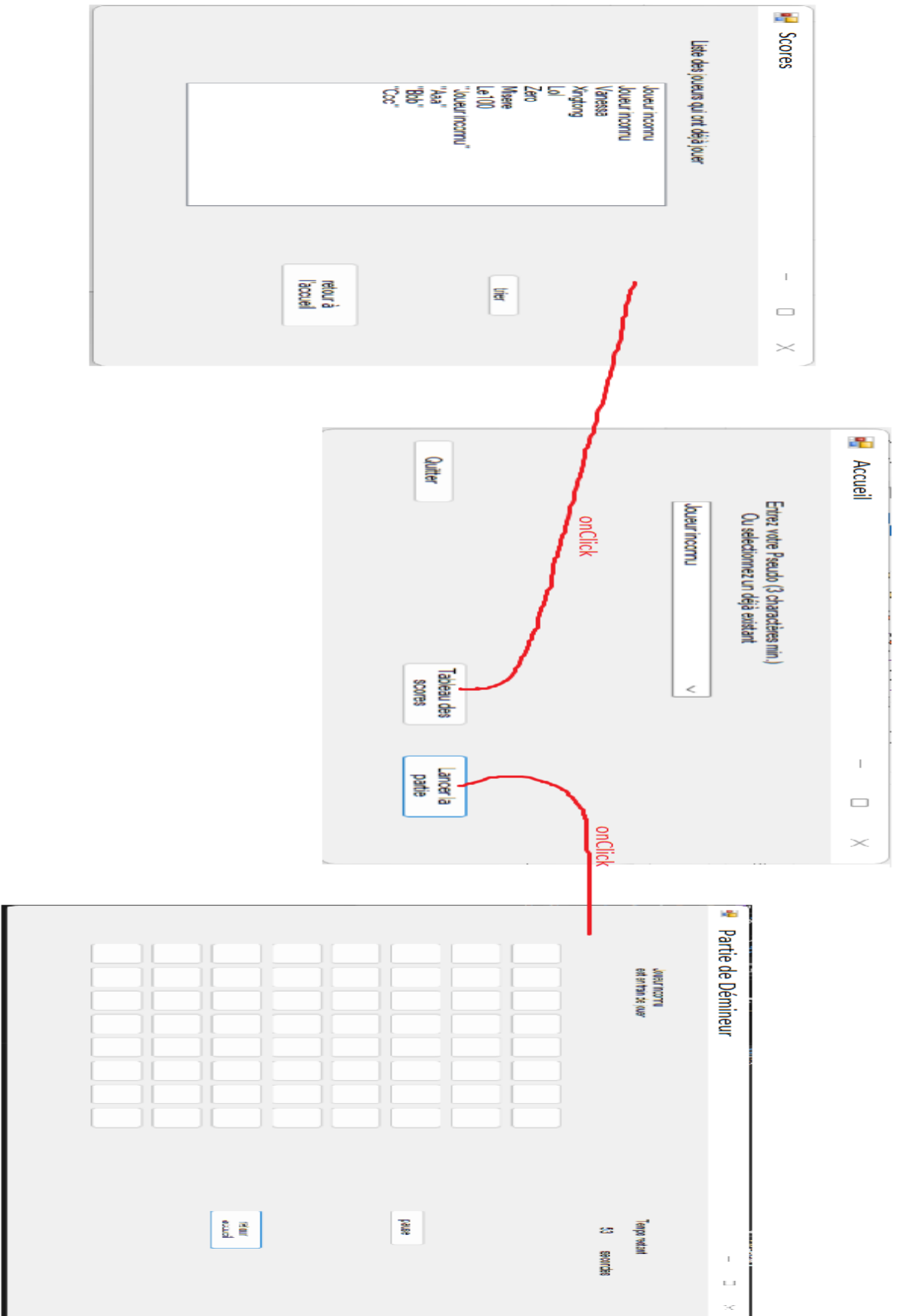


MsgBox du bouton "Quitter" (Fig1-4)



MsgBox du bouton "retour à l'accueil" (Fig3)

Schéma d'ordonnancement des formulaires



Fonctionnalités abouties

Fonctionnalités abouties concernant la mise en place du jeu :

Nous avons une fonctionnalité qui crée automatiquement du grille de boutons, espacés avec un intervalle régulier.

Une fonctionnalité s'occupe de générer 10 chiffres entre 0 et 63 aléatoirement, puis de les faire correspondre au bouton avec le même numéro d'identification. Cette correspondance sert à placer les mines au hasard sur la grille.

Puis après la création des boutons et des mines, une fonction est appelée par un événement cliquez, elle est développée dans le module ModuleJouer. Si j'appuie sur n'importe quel bouton créer alors il va se passer un évènement. Cet événement correspond au démasquage d'une première case, si elle est vide elle libère les autres autour d'elle et indique combien de mine une case adjacente (fonction incorrect/marche pas).

Il y a une fonctionnalité qui permet de convertir le timer en seconde et une autre pour stopper le Chronomètre, quand le chronomètre est arrêté la grille est masquée et on ne peut pas la toucher. Et une fonctionnalité pour redémarrer le chronomètre et montrer la grille.

Conclusion

Comme nous n'avons pas abouti à présenter une fonctionnalité qui permet de jouer au Démineur. Pour améliorer le code, je pense qu'il faut créer des propriétés pour les boutons, comme une énumération des états que les boutons peuvent avoir pour faciliter le démasquage. Il faudrait appliquer correctement le concept de la célèbre fonction flood fill pour jouer. Puis créer une classe Joueur si possible, avec des attributs qu'on pourrait retrouver dans le tableau de scores. Ainsi qu'un compteur de case démasqué. Puisque c'est supposé être une interface Homme Machine, il aurait fallu faire une interface plus attractive peut-être