



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université des Sciences et de la Technologie Houari Boumediene



Faculté d'Électronique et d'Informatique Département Informatique

Projet Création Du Jeu

Filière : Informatique Spécialité : Informatique Académique

---

## Thème : Jeu Modèle De Droite

---

**Proposé et encadré par :**

**Prof. SMAILI Cherif .**

**Réalisé par :**

**SAADA Samir**

# Table des matières

---

Introduction...	3
Partie I : Analyse Du Jeu .....	3
1. Les Cas D'utilisation .....	3
2. Diagramme De Séquence .....	5
Partie II : Le guide de l'utilisateur.....	5
1. Présentation Générale.....	5
2. Déroulement Du Jeu.....	6
Partie III : Le guide du programmeur.....	8
1. Modules Interfaces De Base.....	8
2. Description Détaillée Sur Les Modules... ..	9
Conclusion .....	11
Référence.....	11

# Introduction

---

## Introduction :

Le jeu « modèle de droite » est un jeu qui consiste à remplir toutes les cellules vides d'une matrice  $6 \times 6$  avec des lignes diagonales. Sur chaque coin d'une cellule est indiqué un chiffre (0, 1, 2, 3, 4) représentant le nombre de lignes se connectant à ce point (voir figure 1).

Le jeu s'arrête dès que toutes les diagonales insérées correspondent aux chiffres indiqués sur chaque coin.

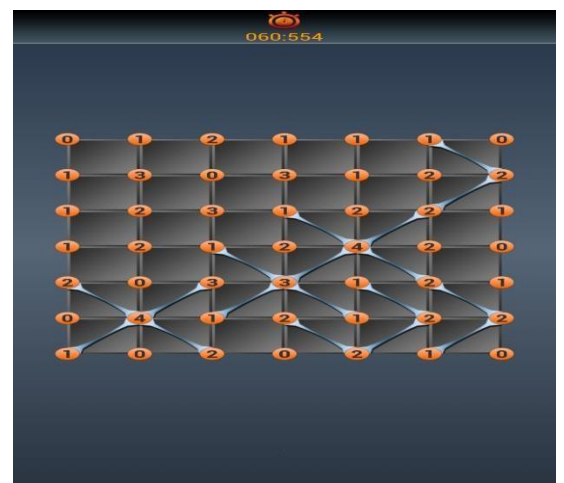


Figure 1 : description générale du design.

## Partie I : Analyse du jeu

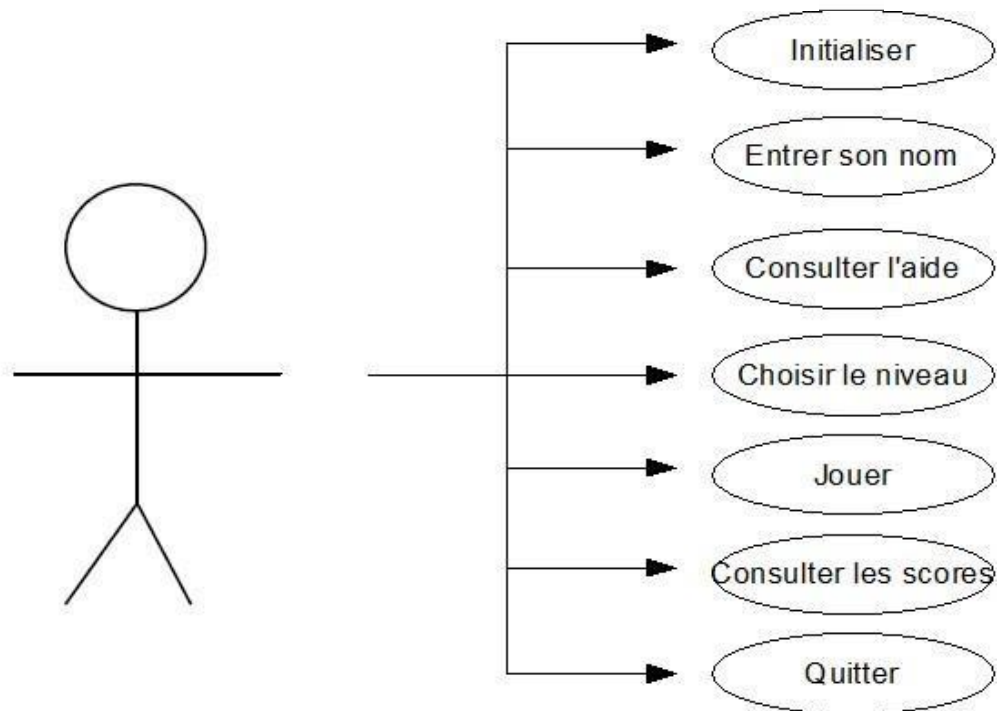
---

### 1. Les cas d'utilisation

- Acteur : L'utilisateur
- Cas d'utilisation pour l'utilisateur :
  - \_ Initialisation (lancement du jeu)
  - \_ Choix mode de jeu
  - \_ Mettre les diagonales

- \_ Consulter le score
- \_ Quitter

➤ Diagramme des cas d'utilisations :



**Figure 2** :diagramme des cas d'utilisation

## 2. Diagramme De Séquence :

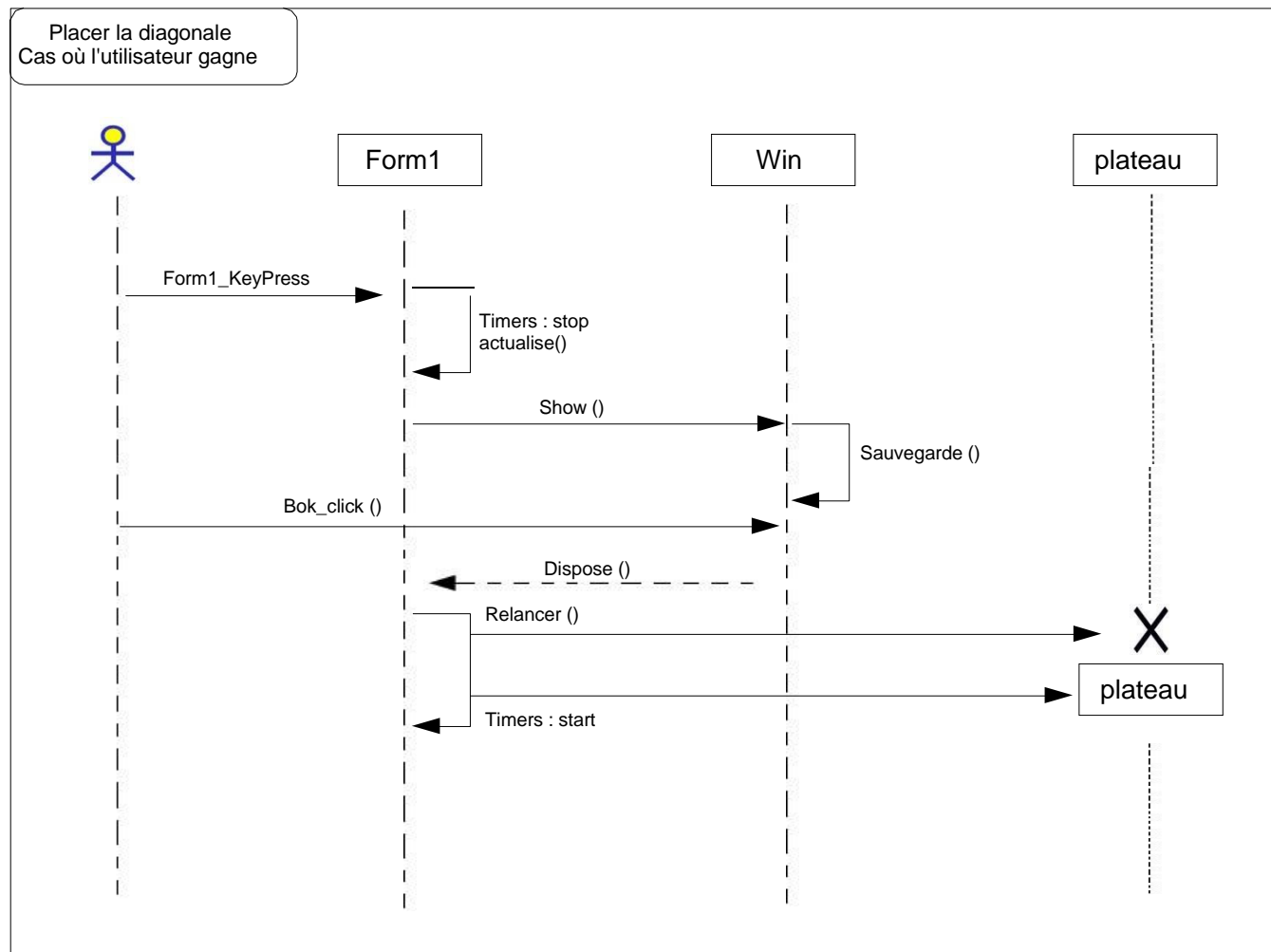


Figure 3 :diagramme de séquence.

# Partie II : Le guide de l'utilisateur

Dans cette partie nous allons expliciter les règles et le fonctionnement de notre jeu modèle de droite .

## 1. Présentation générale :

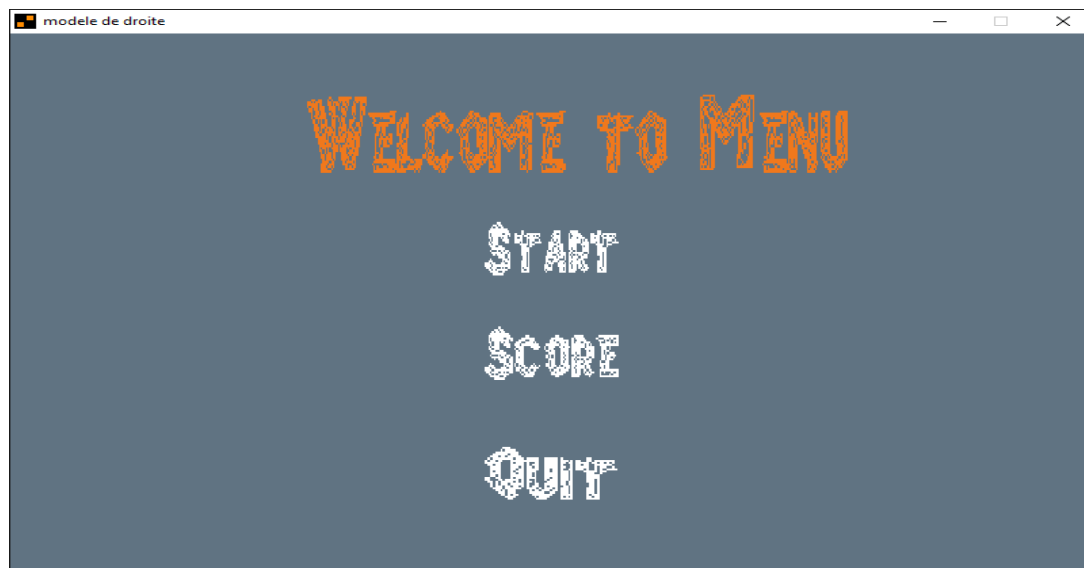
L'idée du jeu est de créer une solution d'une façon aléatoire, autrement dit mettre des 1 et des 2 dans chaque cellule qui représente les diagonales , puis en se basant sur cette solution on met les

chiffres qui correspondent à chaque cellule, pour les différents modes on doit créer des procédures qui les gèrent.

Lorsque le joueur est en train de placer les diagonales on doit vérifier l'état des cellules qu'il a modifié et on affecte à une variable des chiffres 0,1,2 qui correspondent à des diagonales et on vérifie si ces deux variables sont identiques la solution avec ce dernier et on affiche la fenêtre gagnant .

## 2. Déroulement du jeu

Lorsque le jeu débute nous sommes sur la page principale du jeu qui nous permet de choisir le niveau auquel nous souhaitons démarrer notre partie. Si c'est la première partie, seul le niveau 1 sera disponible. Une fois que nous avons entré cette information nous sommes prêt à démarrer le jeu, on clique alors sur le bouton start. Nous avons donc la fenêtre suivante :



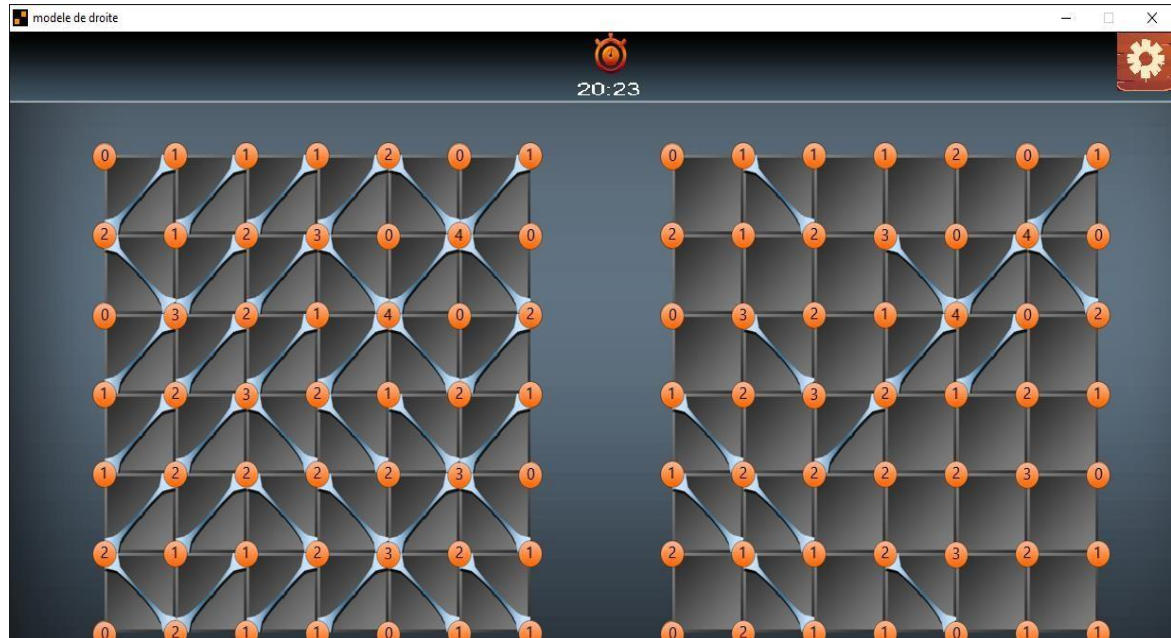
**Figure 4** : chargement du menu.

Lorsque on clique sur le bouton PLAYER ,Nous avons donc la fenêtre suivante : (figure 5)



**Figure 5 :** lorsque le joueur appuie sur le bouton PLAYER.

Lorsque on clique sur le bouton PLAYER VS COMPUTER on va joueur contre la machine (IA)  
Nous avons donc la fenetre suivante :



**Figure 6 :** la machine joue contre l'utilisateur.

Une fois que vous aurez cliqué, vous retrouvez une fenetre similaire à ce que nous avons vu précédemment.

Vous avez alors une fenetre qui s'affiche. Celle-ci est composée du tableau des 10 meilleurs scores. Si votre score le permet vous apparaitrez peut être dans ce tableau.

Pour quitter cette fenetre on doit attendre 5 secondes. Vous pouvez alors recommencer une nouvelle partie.

The image shows a window titled 'modele de droite' displaying a leaderboard. It lists the top 10 players with their names, the date of their best score, and their finish time. The text is in a monospaced font on a dark background.

	NAME:.....	DATE:.....	FINISH TIME:
1-	BEST.....	2021-12-26...(BEST TIME)...	28:994
2-	CHANCE.....	2021-12-26.....	29:115
3-	VOID.....	2021-12-26.....	29:701
4-	MOURAD.....	2021-12-26.....	30:933
5-	GOOD.....	2021-12-26.....	31:957
6-	OKEY.....	2021-12-26.....	33:190
7-	AWOL.....	2021-12-25.....	34:791
8-	THIRD.....	2021-12-26.....	34:944
9-	YOUPI.....	2021-12-25.....	35:490
10-	RACHID.....	2021-12-25.....	37:849

**Figure 7 :**le score des 10 meilleurs joueurs .

# Partie III : Le guide du programmeur

---

Dans cette partie nous allons expliciter certaines parties délicates de notre code.

Afin de gagner du temps nous nous sommes permis de mettre une grande partie de nos attributs en public.

## **1.1 Module Interface De Base :**

### **1.1.1 La fonction jeu( ) :**

Cette fonction gère le premier mode du jeu ( Player) .

- But : Cette fonction permet de déclencher chaque déplacement du diagonale, et le contrôle du chaque cellule.
- Idée : On test l'état de la cellule :
  - On initialise les cellules à des texture qui représente la cellule vide.
  - Puis à chaque clique du souris sur la case choisi on affecte à ces cellules l'état du texture qui est représenté sous forme des diagonales.
  - Si le joueur gagne, autrement dit tout les cellules sont identiques au solution proposée ,on affiche la fenêtre «Win» ,et on demande d'introduire le nom
  - Si le joueur perd, on enregistre le score (s'il fait parti des 10 meilleurs), on affiche la fenêtre « perdu » et on demande d'introduire le nom.

### **1.1.2 La fonction : Jeuordi( )**

Cette fonction gère le deuxième mode du jeu (COMPUTER).

- But : Cette fonction permet la machine de jouer toute seule.
- Idée : On récupère la solution proposé au début du jeu et on la redessiner on changeant les cellules d'une façon aléatoire .

### **1.1.3 La fonction compvsplayer( ) :**

Cette fonction gère le troisième mode du jeu (PLAYER VS COMPUTER) .

- But : Cette fonction permet l'utilisateur de jouer contre la machine.



- Idée : on combine entre les deux fonctions (jeu et jeuordi ),qu'on doit mettre le fonctionnement de la machine dans le bon emplacement dans le mode ( du joueur tout seul ) pour que ce dernier fonctionne.

## **1.2 Description Détaillée Sur Les Modules:**

### **Module Plateau:**

- *void Draw\_Game(SDL\_Renderer \*renderer,SDL\_Rect points[49],SDL\_Rect cells[36]) : une procédure pour dessiner le plateau du jeu.*
- *void Draw\_comps(SDL\_Renderer \*renderer,SDL\_Window \*window,SDL\_Rect points[49],SDL\_Rect cells[36],SDL\_Rect cells2[36],SDL\_Rect points2[49]) : une procédure pour dessiner les chiffres de chaque cellule pour le mode computer vs player.*

### **Module Paramètre:**

- *void menu(SDL\_Renderer\*renderer) :une procédure pour afficher le menu.*
- *void gameMode(SDL\_Renderer\*renderer): une procédure pour afficher le choix des différents modes.*
- *void pauseMode(SDL\_Renderer\*renderer): une procédure pour mettre un choix de quitter ou bien de sauvegarder la partie ou bien de la reprendre .*

### **Module Affichage:**

- *void Blit(SDL\_Renderer \*renderer, SDL\_Texture \*texture,SDL\_Rect dest) :une procédure pour initialiser plateau .*
- *void youWin2(SDL\_Renderer\*renderer, SDL\_Window \* window, unsigned tfinale) :une procédure pour afficher le temps et de demander à l'utilisateur d'introduire son nom(cas win) .*
- *void youLose(SDL\_Renderer \*renderer,SDL\_Window \*window): une procédure pour afficher le temps et de demander à l'utilisateur d'introduire son nom(cas lose) .*
- *void Draw\_Timer(SDL\_Renderer \*renderer, int tiempo) : une procedur pour dessiner le timer pour le mode player et pour le mode computer .*
- *int Draw\_Settings(SDL\_Renderer \* renderer) ) : une fonction pour dessiner les paramètres pour le mode player et pour le mode computer .*
- *void draw\_Model(SDL\_Rect points[49],SDL\_Renderer \* renderer, int grid[7][7]) : une procédure pour dessiner les chiffres du chaque cellule.*
- *void Displayer(SDL\_Renderer\*renderer, char \* str) : une procédure pour afficher le nom .*
- *void Draw\_Timer2(SDL\_Renderer \*renderer, int tiempo) : une procédure pour dessiner le timer pour le mode computer vs player.*

- *int Draw\_Settings2(SDL\_Renderer \*renderer) une fonction pour dessiner les paramètres du mode computer vs player.*
- *void scoreboardDisplay(SDL\_Renderer\*renderer, FILE \*f): : une procédure pour afficher le score .*

### **Module Conversion:**

- *SDL\_Texture \*initialize\_texture\_from\_file(const char\* file\_name, SDL\_Renderer \*renderer) int convertNumber(int oldnumber) : une fonction pour la conversion d'une photo à une texture .*
- *SDL\_Texture \*initialize\_texture\_from\_TTF(const char\* name, SDL\_Renderer \*renderer) : une fonction pour la conversion d'un font à une texture .*

### **Module fonctionnalité :**

- *void grillEtSoluca(int soluce[6][6], int grille[7][7]) :une procédure qui sert à créer une solution et le chiffre de cette solution.*
- *int loadGame(FILE \*f, int Gmode) :une fonction qui gère le mode player vs computer .*
- *int Icon(SDL\_Window \*window): :une fonction qui sert à dessiner l'icone.*
- *void errorSave(SDL\_Renderer\*renderer) :une procédure qui déclare qu'il n'y a pas d'une partie précédente.*
- *void saveOrCont(SDL\_Renderer\*renderer) :une procédure qui sauvegarde le score.*
- *void compFin(SDL\_Renderer\*renderer, unsigned time) :une procédure qui déclare la fin du mode player vs computer .*
- *int jeuFinOrdi(int soluce[6][6]): :une fonction qui déclare la fin du partie .*
- *void jeu(SDL\_Renderer \*renderer,SDL\_Window \* window, int grid[7][7], int sol[6][6], int mod) :une procédure qui gère le mode player.*

### **Module IA:**

- *void compvsplayer(SDL\_Renderer \*renderer,SDL\_Window \* window, int grid[7][7], int sol[6][6], int mod) :une procédure qui gère le mode player vs computer .*
- *void jeuOrdi(SDL\_Renderer \*renderer,SDL\_Window \* window, int grid[7][7], int sol[6][6], int mod) :une procédure qui gère le mode computer*

# Conclusion

---

Ce projet nous a appris d'une part à mettre en œuvre les bases algorithmiques que nous avons appris tout au long de ce semestre et d'autre part, une exploitation de la large documentation existant sur le net pour pouvoir installer et maîtriser la bibliothèque SDL pour une première dans notre cursus universitaire.

# Références

---

<https://www.libsdl.org/>

<https://devopssec.fr/article/bases-fondamentales-de-laffichage-graphique-sdl2>

<https://www.parallelrealities.co.uk/news/20210619/sdl2-widget-tutorial.html>

<https://devopssec.fr/article/cours-introduction-sdl-2>

<https://intelligence-artificielle.developpez.com/actu/306783/Le-code-source-d-une-intelligence-artificielle-qui-joue-au-titre-Fruit-Ninja-pour-Android-iOS-et-Windows-est-disponible-mais-divise-quant-a-l-utilisation-du-terme-IA-pour-sa-designation/>