



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

Jeux De Mots

Proposé et encadré par:

-Prof. SMAILI Cherif .

Réalisé par :

-Fekir Nabila Chehla.

-Saidi Fatma Zohra.

-Boutkedjirt Aya.

-Kadri Yousra Ferdaous.

e-mail: nabilachehlafeikir@gmail.com

Table des matières :

• Introduction.....	3
• Schéma du jeu.....	4
• Déroulement du jeu.....	5
• Le guide du programmeur :.....	8
1.La structure.....	8
2.Description générale de la solution.....	9
3.Description détaillée des modules :	10
-Module paramètre.....	10
-Module fonctionnalité.....	10
-Module affichage.....	10
-Module manipulation des fichiers.....	11
• Conclusion.....	11
• Référence.....	11

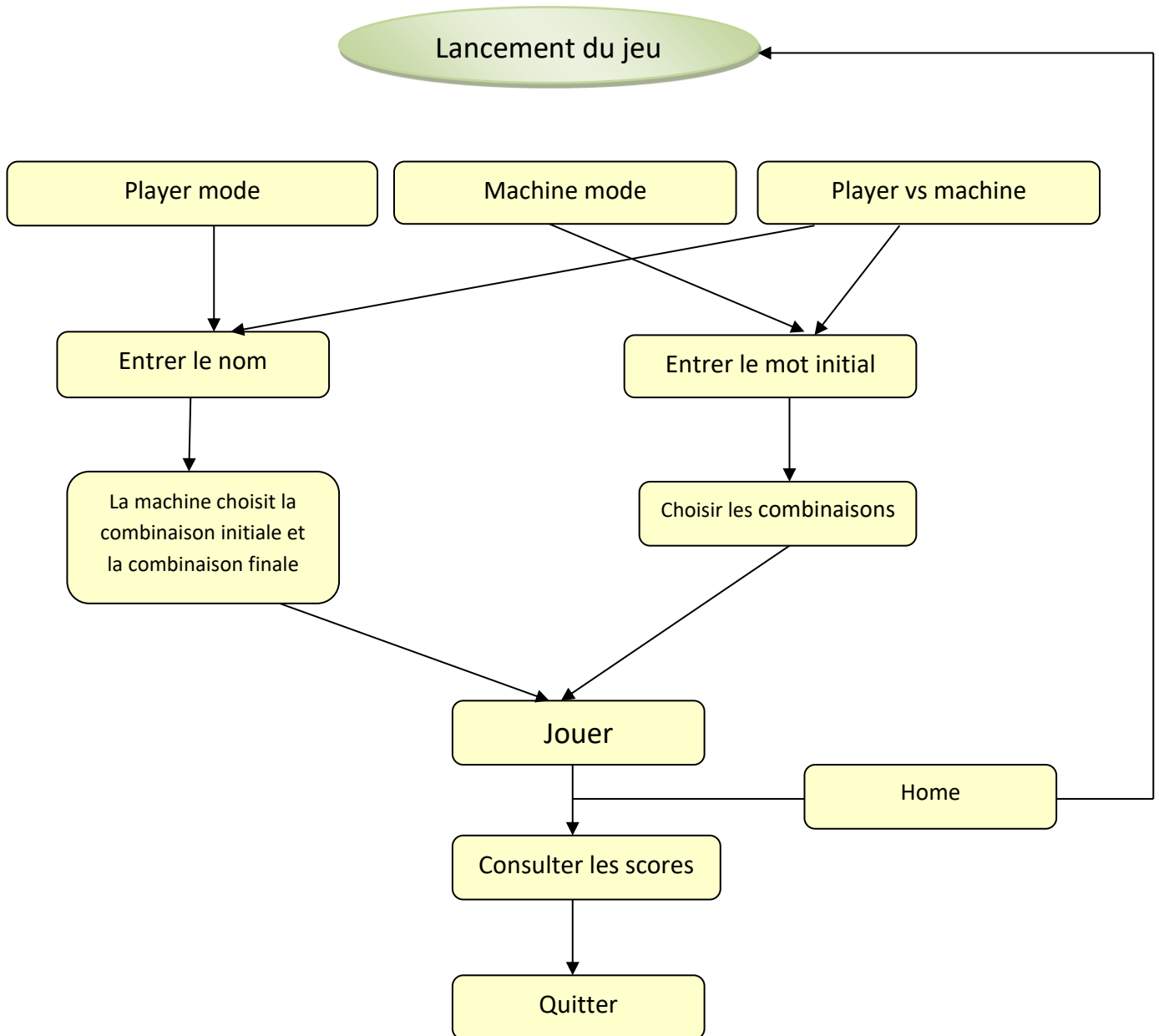
● Introduction :

Le jeu « Jeu de mots » est une initiation en programmation en langage C, en utilisant les différentes structures de l'algorithmique. Ce jeu programmé, représente un parfait moyen de divertissement pour l'utilisateur, ce dernier tentera de trouver le résultat final en un minimum de temps, afin d'avoir un meilleur score.

Le Game consiste à créer un mot à partir de N lettres inscrites. L'utilisateur (ou la machine) présente une nouvelle combinaison et demande à la machine (ou à l'utilisateur) de la trouver durant un temps compté, tout en effectuant des permutations entre des lettres adjacentes qui se renversent à chaque changement de leur position.

Le jeu se termine dès que la combinaison finale sera trouvée. Et le joueur pourra ainsi voir le score obtenu.

- Schéma du jeu :



● Déroulement du jeu :

Lors du lancement du jeu, un menu apparaît et nous permet de choisir un des 3 mode pour jouer :

1. Mode Player (Utilisateur joue seul).
2. Mode machine.
3. Mode Player VS Machine (La machine et l'utilisateur jouent en même temps).

Ou bien, on peut choisir de consulter le nom, la date et le score des 5 meilleurs joueurs

(I.e. les joueurs avec les meilleurs scores).



Figure 01 : menu

D'abord, si vous décidez de jouer en mode Player ou Player Vs machine, le jeu vous demande d'entrer votre nom, afin de le sauvegarder et l'afficher si vous obtenez un bon score.

Ensuite, si vous êtes en mode Player Vs machine ou en mode machine, vous allez entrer le mot initial (soit avec clavier ou souris), et on vous proposera entre 1 et 15 combinaisons parmi lesquelles vous devrez choisir une pour jouer.

ET C'EST PARTIE !



Figure 02: entrer le nom

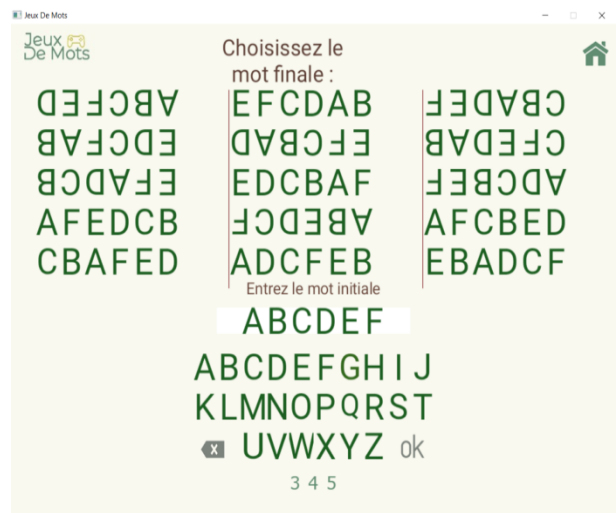


Figure 03: entrer le mot initial et final

Un exemple de mode machine avec mot de départ ABCDEF et mot finale EFCDAB. Permutation avec 4 lettres adjacentes :



Figure 04 : machine joue

Un exemple de mode machine Vs joueur avec mot de départ ABCDEFGH et mot finale ABGDCFEH.

Permutation avec 5 lettres adjacentes :

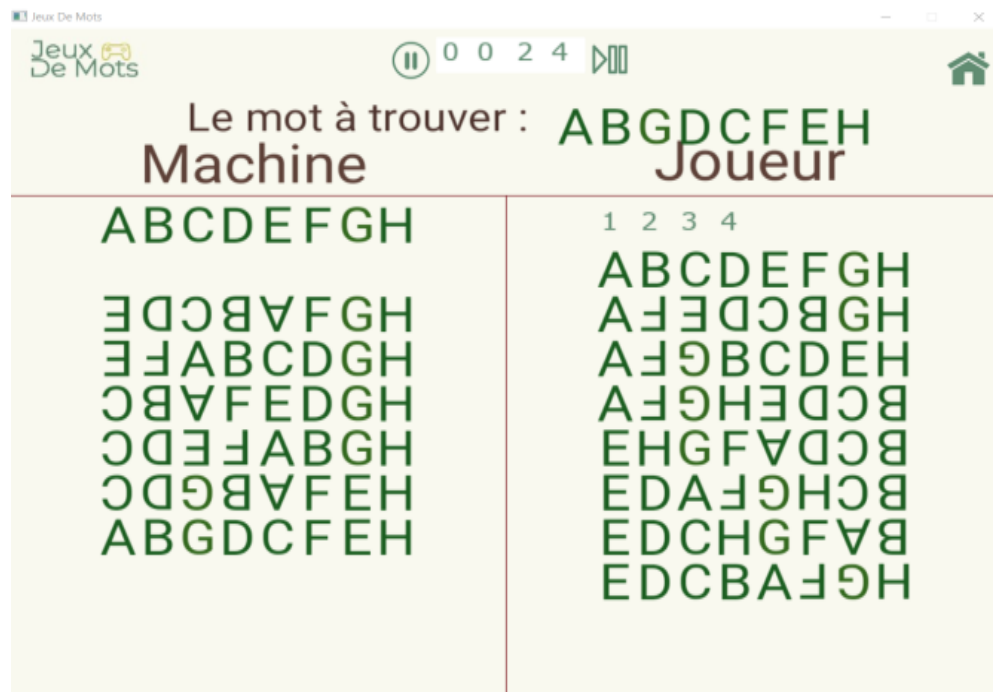


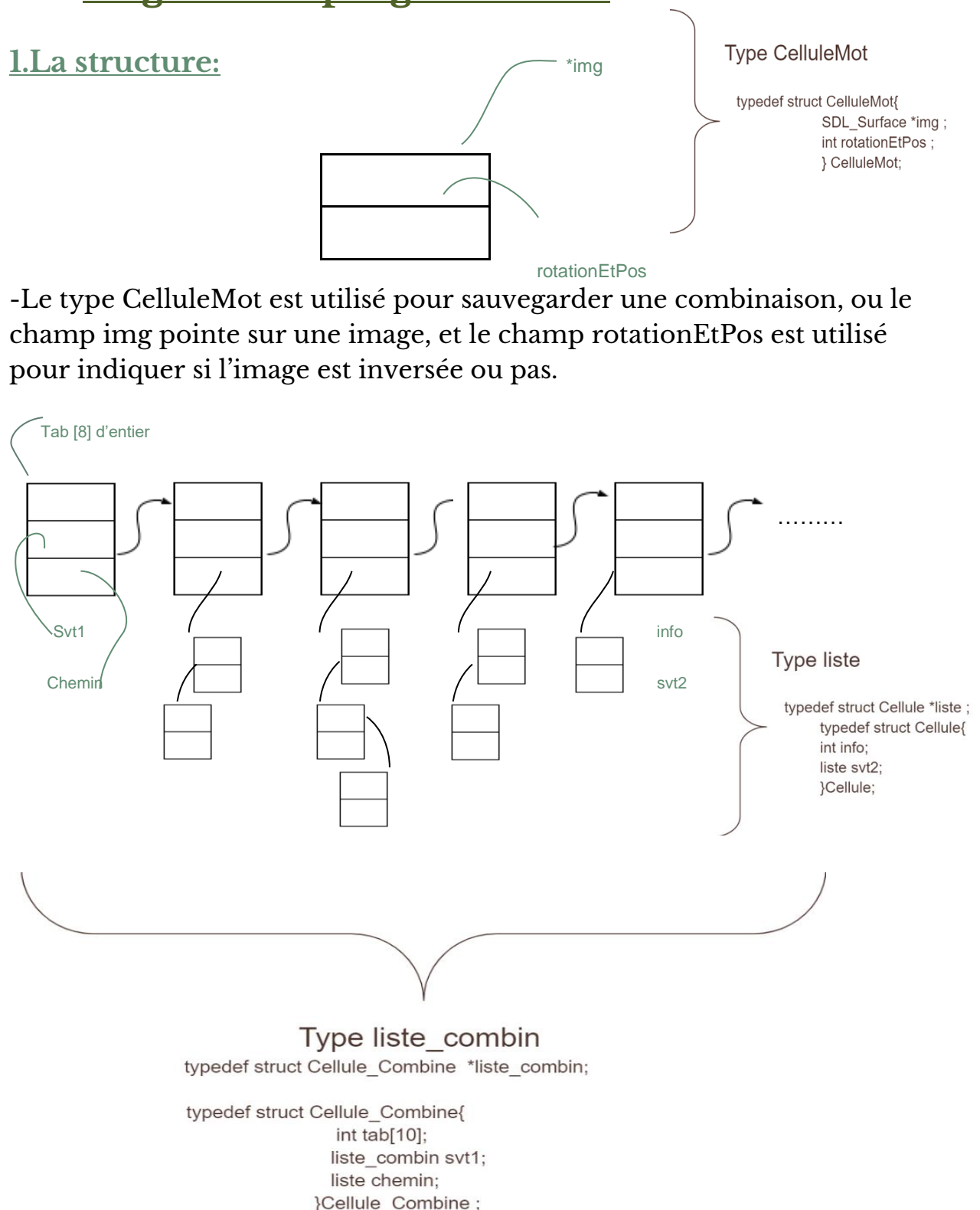
Figure 05 : machine joue contre joueur

Et en fin de partie, vous pouvez consulter si votre nom figure dans la liste des 5 meilleurs joueurs, avec la possibilité de rejouer d'autres parties.



• Le guide du programmeur:

1.La structure:



-le type **liste_combin** est utilisé pour sauvegarder toutes les combinaisons avec les chemins en forme numérique.

2.description générale de la solution :

-à chaque fois que l'on saisit une lettre, on l'associe à un chiffre et on sauvegarde cette combinaison dans un tableau de type CelluleMot.

-Pour trouver toutes les combinaisons, on utilise la fonction:

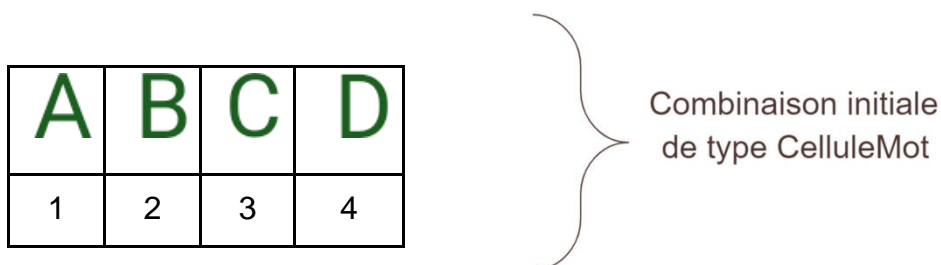
`liste_combin toute_combin(int T[TailleMaxMot],int npermut ,int Taille)`
qui prend le tableau d'entier contenant la combinaison initiale et trouve toutes les combinaisons possibles en permutant les chiffres, cette fonction sauvegarde aussi le chemin(les indices de permutations).

-Pour afficher une combinaison à partir d'une liste liste, on utilise la fonction:

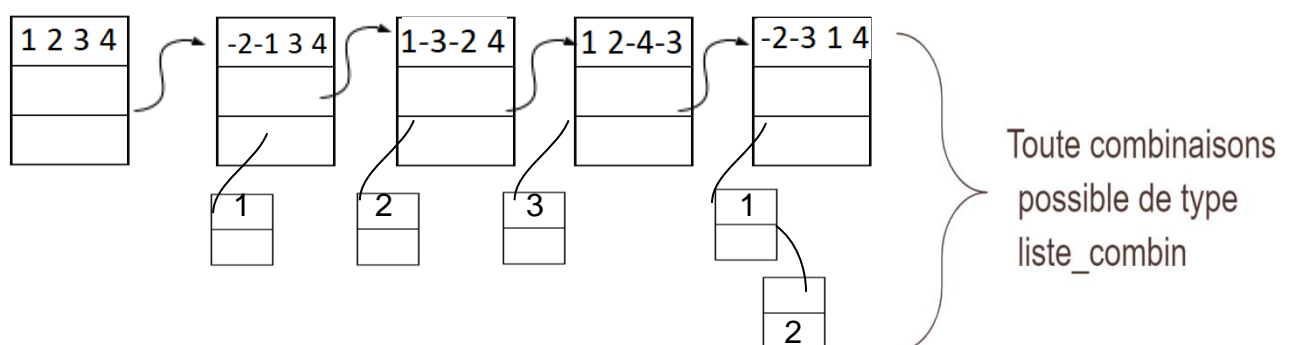
`void transferTabToMot(CelluleMot tabLetChange[TailleMaxMot], CelluleMot tabLet[TailleMaxMot], liste_combin p, int TailleMot)` qui transforme un tableau d'entier en un tableau de CelluleMot.

Par exemple:

-le tableau tab[8] de type CelluleMot:



-les combinaisons possibles:



-Si on veut afficher par exemple la combin 1-3-2 4:

A B C D

3.description détaillée des modules:

Module paramètre :

- *void askContinue*: une procédure qui demande à l'utilisateur s'il veut une nouvelle partie, ou bien continuer depuis la dernière partie non terminée.
- *int getMode*: orienter le joueur vers un mode choisi.
- *void GameBeginMenu* : affiche le menu et met à jour le temps, les fichiers et supprime les données des structures utilisées dans la dernière partie.

Module fonctionnalité :

- *void UserChoixLetClavier* : procédure pour entrer et afficher le nom de l'utilisateur et la combinaison initiale, et les sauvegarder dans les structures et les fichiers.
- *int choixNbPermut*: Une fonction pour choisir le nombre de permutations.
- *void MachineRandomChoixMotbut*: une procédure qui choisit un mot final aléatoirement.
- *void RemplirTabOfCombinPossible*: une procédure utilisée pour le choix pseudo aléatoire du mot finale donné par la machine.
- *void AfficheCombinPossibles* : une procédure pour afficher les combinaisons possibles sur l'écran afin que l'utilisateur puisse choisir une.
- *void UserChoixMotBut*: une procédure qui permet à l'utilisateur de choisir une combinaison.
- *void permuteJoueur*: une procédure qui permute les lettres.
- *void UserPlays*: une procédure qui permet à l'utilisateur de jouer (cliquez sur les boutons de permutation).
- *void MachinePlays*: une procédure qui permet à la machine de jouer.
- *void PlayingBackground*: une procédure qui permet à la machine et l'utilisateur de jouer simultanément.

Module affichage:

- *void afficher*: une procédure pour afficher l'image à partir d'une surface.
- *void afficherClavierWindow*: procédure pour afficher le clavier.
- *void afficherTabWindow*: une procédure pour afficher un mot.
- *void affiche_time*: une procédure pour afficher le temps écoulant.

- *void AfficheBestScore*: une procédure pour afficher les meilleurs scores.
- *void inverserImg*: une procédure pour afficher un mot à l'envers.

Module manipulation des fichiers :

- *void saveCodeMotInitFile*: une procédure pour sauvegarder le mot init dans un fichier.
- *void saveCodeName, saveTimerTxt, saveCodeMotButFil, saveCodeMotInitFile, saveCodeMotCurntFile*: sauvegarder le nom de l'utilisateur, mot initiale, finale et courant ainsi que le timer dans des fichier.
- *recupBestScore*: appelée une fois le jeu est lancée, pour récupérer les informations des joueurs qui ont obtenu les 5 meilleurs scores dans la dernière partie et les stocker dans une structure spécifique.
- *BestScoreMiseAJours*: mis à jour un fichier des 5 meilleurs scores si le joueur a obtenu un meilleur score.

● **Conclusion:**

Le mini projet était une riche expérience qui on espère nous ouvrira les portes pour d'autres réalisations de projets plus complexes dans le futur. D'autre part, la coopération entre les membres du groupe nous a permis l'échange constructifs des idées et l'obtention de meilleurs résultats.

La réalisation du jeu nous a notamment appris la maîtrise du langage C, la bonne exécution des codes écrits, ainsi que l'exploitation du SDL.

● **Référence:**

<https://zestedesavoir.com/tutoriels/1014/utiliser-la-sdl-en-langage-c/la-sdl/>

<https://wiki.libsdl.org/SDL2/Tutorials>

<https://lazyfoo.net/tutorials/SDL/>

<https://www.youtube.com/watch?v=Kddn0MUbn4k>

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