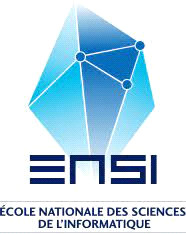
Université de la Manouba  
Ecole Nationale des Sciences de l’informatique  
  
   
  
  
 Sujet :

|  |
| --- |
| Implémentation du Jeux Candy Crush |

Réalisée Par :

***Mahdi Ayedi;***

***Firas Jabeur;***

***Youssef Chraief;***

***Mohamed Chaabouni;***

***Mohamed Aloulou;***

**1-Présentation générale du sujet :**  
En fait, le fonctionnement principal du jeux Candy Crush consiste à l’élimination des bonbons colorés en passant par des combinaisons différentes (combinaisons de trois bonbons ou de quatre ou de cinq ) afin de remplir l’objectif du niveau et de marquer le score nécessaire avec un nombre de click bien déterminés.  
-Lorsqu’une combinaison est effectuée, les bonbons qui le composent sont effacés avec une génération des bonbons aléatoire de haut en bas dans la grille.  
  
  
  
  
  
  
  
**2-presentation générale de la solution :**  
 **2.1-archeticture du programme :**  
**Le jeu est formé principalement par une classe Game et un fichier Main.cpp pour le programme principale;**  
 **2.2-Détaile de la solution :**  
**Pour mieux comprendre la fonctionnalité de la classe déclarés ci-dessus on doit faire une petite récapitulation sur ses attributs et ses méthodes .**

***A-Classe Game :***  
  
***A.1-Les attributs:***  
private:

static sf::RenderWindow window (fenêtre du jeu);

int ts = 74 (taille des candys);

int decalage[8] (necessaire pour la méthode destroy);

***A.2-Les méthodes :***  
void display\_menu(sf::Sprite &back\_gr,sf::Sprite&board):afficher le menu du jeu.  
int levels(): afficher la fenêtre des niveaux et retourner le niveau choisit.  
int credits():afficher la fenêtre du crédits.  
 int options():afficher la fenêtre des paramètres.  
int menu(): gérer la fonctionnalité des buttons du fenêtre Menu.  
void manager():gérer la fonctionnalité de tous les boutons du jeu et l'affichage tant que le jeu est exécuté.  
void destruction(candy grid[][8],int &there\_is\_match,int &isSwap,int &isMoving,int &objectiv) **: l’élimination des bonbons dans la grille(maximum 8 candy ) à travers les positions placés dans la matrice Grid après une vérification que la scène est stable (isMoving==0),qu'il y a une permutation(isSwap==1) valide(there\_is\_match==1)et il y aura une incrémentation du paramétré objectif (le type du bonbon à détruire dans chaque level).**

**void update (candy grid[][8],int &isMoving):corriger les position des candy après chaque permutation .**  
**int run(int level):**

**- initialiser tous les clés du déroulement du jeu.**

**- remplir la matrice par les candy.**

**-affichage du background.**

**-importer les textures des candys et des boutons.**

**- animer les candys lors d'un clic.**

**- afficher le nombre des clics limites et le décrémenter lors de chaque permutation.**

**- afficher le message du victoire ou bien de la défaite.**

***D – main :***  
**Ce fichier contient la bibliothèque SFML pour créer l’interface graphique de notre jeu Candy Crush . Dedans on a utilisé des méthode citons par exemple**   
**setposition() pour retourner la position d’une sprite et modifier loadfromfile(“path”) pour accéder a une image dans le path donnés .**

**Aussi cette bibliothèque permet l’affichage des boutons play, continuer,levels,options,credits et le choix du level lors du jeu avec une position bien déterminés dans la page window .**  
**Aussi ce fichier contient des méthodes bien définie pour manipuler le jeu(void swap(candy a,candy b),void destroy(int a , int b), int main()).**  
**3.Manuel d’utilisation :**  
3.1.Envorinement logiciel :  
**on a utilisé lors de l’implémentation du jeu comme EDI le Visual Studio Code .**   
**parmi les bibliothèque qu’ on a utilisé on cite ci-dessous :   
<iostream>: pour l’affichage des message   
<windows.h>: pour supprimer la version console dans l’exécution du jeu  
et on a utilisé la bibliothèque SFML pour manipuler graphiquement notre jeux Candy Crush**

**3.2.Présentation du travail réalisé :**  
**D’abord on a l’affichage de la fenêtre d’accueil comme illustre la photo ci-dessous :**



**Ensuite vous pouvez lancer un niveau, par exemple 1**



**Dans cette cas, il Ya l’ouverture d’une autre fenêtre et l’affichage d’une matrice de taille 8\*8 spécifique pour le niveau 1.**



**Finalement, vous devez cliquer sur deux bonbons adjacentes pour les permuter.**

**Si vous gagnez, un message s’affiche comme suit :**

**Si non, un message s’affiche comme suit :**



**4.les problèmes rencontrés et leurs solution** :

|  |  |
| --- | --- |
| Les problèmes rencontrés | Les solutions |
| **Lors de l’exécution du programme , on a rencontrés plusieurs erreurs de compilation.** | **Apres beaucoup de test et de recherche , on a arriver finalement a comprendre les sources de notre erreur et les corriger** |
| **Lors de l’affichage de l’interface graphique avec la bibliothèque SFML et la liaison avec le code écrite en console dans le c++** | **Après quelques recherche et modification de quelque méthodes , on a arrivés a faire la liaison entre les deux langage .** |

**6.références :**  
  
**Pour la conception de l’interface graphique SFML:**  
[https://www.sfml-dev.org/tutorials/ HYPERLINK "https://www.sfml-dev.org/tutorials/2.5/?](https://www.sfml-dev.org/tutorials/2.5/?fbclid=IwAR2pIcM8pJ-FC0NhBgCVVEPaeTprHx7Ok3NEr1vWnBJLUO2_-p3j8isxl44)