

# DOCUMENTATION ARCADE

La documentation qui suit est un ensemble d'instruction du programme « ARCADE ». Ce document recense donc l'ensemble des bonnes pratiques d'utilisation et des manipulations techniques. Ce projet a été développé par Simon MEYER, Léo FORNES et Kevin SPEGT. Au moindre soucis rencontrés, reportez-vous à cette documentation, sinon, contactez-nous.

<a href="mailto:simon.meyer@epitech.eu">simon.meyer@epitech.eu</a>	<a href="mailto:leo.fornes@epitech.eu">leo.fornes@epitech.eu</a>	<a href="mailto:kevin.spegt@epitech.eu">kevin.spegt@epitech.eu</a>
--	--	--

---

Projet réalisé en CPP dans le cadre de la deuxième année du Coursus EPITECH - 2019

# I. Documentation d'utilisation

---

## A. Installation

Procédez premièrement au téléchargement du programme sur votre environnement de travail. Afin de pouvoir correctement utiliser le projet dans son état original, veuillez à posséder une installation des librairies : «NCURSES » et « SFML ». Ces librairies sont très facilement accessibles par le biais de gestionnaire de paquets sur des systèmes d'exploitation Linux.

## B. Build

Différents moyens existent pour construire le projet dans sa globalité ou non. Ces commandes sont à entrer dans le terminal de commande, dans le dossier dans lequel se trouve l'intégralité des fichiers du programme.

- Pour simplement compiler le programme :

```
$ make
```

- Pour compiler uniquement le « Core » :

```
$ make core
```

- Pour compiler uniquement les « Libs » :

```
$ make graphicals
```

- Pour compiler uniquement les « Games » :

```
$ make games
```

## C. Utilisation

Afin d'exécuter le programme, il suffit d'entrer cette commande dans le terminal :

```
$ ./arcade ./lib/lib_arcade_XXXXXXX.so
```

Remplacer les « XXXXXX » par le nom d'une des librairies présentes dans le dossier « libs », parmi lesquelles se trouvent par défaut : ncurses, libcaca, sfml.

### 1. Menu

Le Menu sert à sélectionner le jeu à démarrer ainsi qu'à renseigner son pseudonyme, utile à l'enregistrement des scores bien qu'optionnel.

- |                    |   |
|--------------------|---|
| <b>[Echap] :</b>   | Permet de quitter le programme.   |
| <b>[F1-F?] :</b>   | Permet de changer de libre graphique, le numéro FX dépend du nombre de librairie chargée, par défaut, F1, F2 ou F3. |
| <b>[ ^   V ] :</b> | Permet de naviguer parmi les jeux proposés.   |
| <b>[Entrée] :</b>  | Permet de sélectionner le jeu souhaité.   |

**[Espace] :** Permet de recharger les librairies présentes.  
**[0-9] & [A-Z] :** Appuyer sur ces touches afin d'écrire votre pseudonyme.

## 2. Minesweeper

Le but du jeu est de trouver toutes les bombes cachées à l'aide des numéros indiqués sur les cases. Une case ayant le chiffre 1 signifiera qu'il y a 1 bombe cachée autour de celle-ci. Afin de démasquer les bombes, il suffit de poser un drapeau sur chacune d'entre elles.

**[Echap] :** Permet de revenir au menu.  
**[F1-F?] :** Permet de changer de libre graphique, le numéro FX dépend du nombre de librairie chargée, par défaut, F1, F2 ou F3.  
**[Espace] :** Permet de relancer une partie.  
**[E-A-H] :** Permet de choisir le niveau de difficulté.  
**[Clique droit] :** Permet de poser un drapeau.  
**[Clique gauche] :** Permet de découvrir la case sélectionnée.

## 3. Nibbler

Le but du jeu est de faire survivre son serpent dans ses déplacements afin de manger les pommes qui apparaissent aléatoirement dans la zone de jeu. A chaque pomme ingurgiter, la taille du serpent augmente. Attention ! Le serpent ne doit pas toucher un mur, ni sortir de la zone de jeu, ni se manger la queue !

**[Echap] :** Permet de revenir au menu.  
**[F1-F?] :** Permet de changer de libre graphique, le numéro FX dépend du nombre de librairie chargée, par défaut, F1, F2 ou F3.  
Menu - **[Espace] :** Permet de recharger les zones de jeu disponibles.  
Menu - **[ ^ | v ] :** Permet de parcourir les zones de jeu disponibles.  
Menu - **[Entrée] :** Permet de sélectionner le jeu souhaité.  
Jeu - **[Espace] :** Permet de mettre le jeu en pause.  
Jeu - **[ ^ | v | < | > ] :** Permet de se déplacer.

## 4. Pacman

Le but du jeu est survivre à l'attaque des fantômes tout en essayant de manger l'intégralité des points présents sur la map.

**[Echap] :** Permet de revenir au menu.  
**[F1-F?] :** Permet de changer de libre graphique, le numéro FX dépend du nombre de librairie chargée, par défaut, F1, F2 ou F3.  
Menu - **[Espace] :** Permet de recharger les zones de jeu disponibles.  
Menu - **[ ^ | v ] :** Permet de parcourir les zones de jeu disponibles.  
Menu - **[Entrée] :** Permet de sélectionner le jeu souhaité.  
Jeu - **[Espace] :** Permet de mettre le jeu en pause.  
Jeu - **[ ^ | v | < | > ] :** Permet de se déplacer.

## II. Documentation technique

---

### A. Implémenter une nouvelle librairie graphique

Pour implémenter une nouvelle librairie graphique, vous devez implémenter des fonctions génériques nécessaires à la compabilité de l'ARCADE. Il sera donc nécessaire de la compiler de sorte à obtenir une librairie partagée (.so). Les fonctions génériques à respecter découle de l'interface **IGraphic** :

- Event **getEvent()** ;
  - Récupération de l'événement en cours.
- void **clearWindow**(Color color) ;
  - Réinitialise la fenêtre.
- void **displayWindow()** ;
  - Affiche le contenu de la fenêtre.
- void **printPixel**(Vector pos, Color color) ;
  - Affiche un pixel de couleur à une position donnée.
- void **printRectangle**(Vector pos, Vector size, Color color) ;
  - Affiche un rectangle de couleur à une position donnée.
- void **printText**(Vector pos, const std::string &text, Color color, Color background) ;
  - Affiche un texte d'une couleur de police et de fond donnée à une position renseignée.

La position devant être relative et proportionnelle à la librairie graphique utilisé. La SFML par exemple compte en pixel tandis que la NCURSES en caractère. Il est important de noter que toutes les couleurs ne sont pas disponibles dans tous les langages.

### B. Implémenter une nouvelle librairie de jeu

Pour implémenter une nouvelle librairie de jeu, vous devez implémenter des fonctions génériques nécessaires à la compabilité de l'ARCADE. Il sera donc nécessaire de la compiler de sorte à obtenir une librairie partagée (.so). Les fonctions génériques à respecter découle de l'interface **IGame** :

- void **setUsername**(const std::string &name) ;
  - Enregistre le pseudonyme de l'utilisateur.
- void **manageEvent**(IGraphic &graphic) ;
  - Récupère et traite les événements (claviers, souris, pad).
- void **updateGame()** ;
  - Actualise les données et le contenu du jeu.
- void **displayWindow**(IGraphic &graphic) ;
  - Affiche le contenu du jeu.

### **C. Panels d'outils**

Quelques outils sont à disposition pour la création de nouvelles librairies afin d'assurer la cohésion des librairies. Des classes **Color**, **Vector**, **Event** sont disponibles. Ces classes permettent d'avoir des couleurs et des events génériques pour assurer une bonne communication entre les différents pôles du programme. La classe Vector quant à elle permet une utilisation simple de vector de position et de mouvement entre les librairies.