|  |
| --- |
| CCI Campus |
| Pacman Rezurexion |
| Documentation technique développeur |

|  |
| --- |
|  |

Table des matières

[DESCRIPTION GENERALE 2](#_Toc511727666)

[Fonctionnalités principales 2](#_Toc511727667)

[CONTEXTE D’UTILISATION 2](#_Toc511727668)

[Public visé 2](#_Toc511727669)

[Systèmes d’exploitation 2](#_Toc511727670)

[PRINCIPAUX MODULES / CLASSES / LIBRAIRIES 3](#_Toc511727671)

[Librairies intégrées 3](#_Toc511727672)

[Classes développées 3](#_Toc511727673)

[Héritage 3](#_Toc511727674)

[RESSOURCES 4](#_Toc511727675)

[Sprites et textures 4](#_Toc511727676)

[Cartes 4](#_Toc511727677)

[ENVIRONNEMENT DE TRAVAIL 5](#_Toc511727678)

[Edition du code 5](#_Toc511727679)

[Compilation 5](#_Toc511727680)

[Versioning 5](#_Toc511727681)

[BASE DE DONNEES 6](#_Toc511727682)

[Technologie 6](#_Toc511727683)

[Modèle de données 6](#_Toc511727684)

[Localisation 6](#_Toc511727685)

[DOCUMENTATION DU CODE 6](#_Toc511727686)

[Technologie 6](#_Toc511727687)

[Génération 6](#_Toc511727688)

[Emplacement documentation 6](#_Toc511727689)

[LIVRAISON DE L’EXECUTABLE 6](#_Toc511727690)

[Technologie 6](#_Toc511727691)

[Fichiers inclus 6](#_Toc511727692)

# DESCRIPTION GENERALE

Pacman Rezurexion est, comme son nom l’indique, une implémentation de Pac-Man© avec quelques différences.

## Fonctionnalités principales

* Les fantômes ne bougent que si le joueur est suffisamment proche
* Certains niveaux ne sont accessibles que si le joueur a un score accumulé suffisant

# CONTEXTE D’UTILISATION

## Public visé

Le jeu s’adresse principalement aux retro-gamers.

## Systèmes d’exploitation

Le jeu est uniquement jouable sur Windows et les systèmes de type Unix.

# PRINCIPAUX MODULES / CLASSES / LIBRAIRIES

## Librairies intégrées



Interface de programmation destinée à construire des jeux vidéos



Interface C pour communiquer avec une base de données

## Classes développées

* **Game : Gestion des états du jeu**
* **Database : CRUD pour base de données**
* **Object : Objets plaçables sur l’arène**
* **ObjectCollection : Ensemble d’objets plaçables sur l’arène**
* **Map : Arène du jeu**
* **Menu : Classe réutilisable pour construire des menus**
* **ChooseMapMenu : Le menu qui permet de choisir une arène**
* **ScoreMenu : Menu affichant le score de chaque partie pour chaque joueur**
* **Logger : Aide au débogage**
* **TextBox : Boîte de texte réutilisable retournant le résultat spécifié par l’utilisateur**
* **Player : Représentation du joueur humain**
* **PacMan : Pac-man**
* **PacGum : Pac-gommes mangeables par Pac-man**
* **Wall : Mur**
* **Ghost : Fantôme ennemi de Pac-man**

## Héritage

* Héritent de Object les classes :
  + PacGum
  + PacMan
  + Ghost
  + Wall
* Héritent de Menu les classes :
  + ScoreMenu
  + ChooseMapMenu

# RESSOURCES

## Sprites et textures

Chaque sprite et texture du jeu a été réalisé avec Paint.Net pour le dessin des frames et WinSpriteCreator pour la composition des feuilles.

Il y a trois feuilles de sprites :

* Une pour le Pac-man
* Une pour les fantômes
* Une pour les pac-gommes

Il y a une texture pour le mur.

Tous les sprites et textures sont contenus dans le dossier *resources* quit doit être à la racine de l’exécutable.

## Cartes

Une carte peut être créée avec un éditeur de texte quelconque. Une carte est un fichier à l’extension *.map* qui doit respecter les règles syntaxiques suivantes :

* La première ligne est un en-tête contenant les dimensions de la carte au format ***x,y***
* La deuxième contient le score à partir duquel la carte est accessible
* Un saut de ligne correspond à l’incrémentation de l’axe ***y***

Les types d’objets du jeu sont représentés par les caractères suivants :

* **\*** représente une pac-gomme
* **-** ou **|** ou **+** représente un mur
* **G** représente un fantôme

Exemple :

10,10 <- dimensions de la carte

0 <- score nécessaire pour accéder à la carte

\*\*\*\*\*\*\*\*\*\*

\*--------\*

\*\*\*\*\*\*\*\*|\*

\*|-|G|\*||\*

\*|\*|\*|\*\*\*\* [ METTRE UNE IMAGE REPRESENTANT LA CARTE ]

\*|\*|\*|\*||\*

\*|\*|\*|\*|\*\*

\*\*\*\*\*\*\*||\*

\*--------\*

\*\*\*\*\*\*\*\*\*\*

# ENVIRONNEMENT DE TRAVAIL

## Edition du code

L’édition du code a été réalisée sous vim8 / vim7. Il n’est actuellement porté sur aucun EID mais devrait compiler facilement avec les systèmes de build de ***Code::Blocks*** ou ***CodeLite***.

## Compilation

Un ***Makefile*** a été écrit afin de compiler le code du projet. La compilation est indépendante de la plateforme tant qu’est utilisé le compilateur ***gcc*** pour ***Linux*** et ***MinGW*** pour ***Windows***.

Il y a deux cibles de build : **Debug** et **Release**. Il existe dans le ***Makefile*** les règles correspondantes aux deux cibles.

Compiler les deux versions: [~/pacman-rezurexion]$ make

Compiler la version debug : [~/pacman-rezurexion]$ make debug-target

Compiler la version release : [~/pacman-rezurexion]$ make release-target

La cible **Debug** contient les symboles de débogage et est lancé dans un terminal.

La version **Release** ne contient aucun symbole de débogage et s’exécute en dehors d’un terminal.

Le projet a besoin des librairies statiques de la ***SFML*** et de ***SQLite***. Pour ce faire une compilation « à la main » est très fortement recommandée, notamment à cause des problèmes de compatibilités entre compilateurs et bugs de certaines versions.

Cependant, la compilation des dépendances devrait fonctionner avec :

[~/pacman-rezurexion]$ make dependencies

Si la règle ne fonctionne pas, relatez-vous à l’explication du processus du compilation sur la documentation officielle de SFML.

L’API C++ ***SQLite*** est intégrée au projet via une librairie statique.

## Versioning

Pour suivre l’écriture du code et le récupérer sur plusieurs plateformes le gestionnaire de version ***Git*** a été utilisé.

Le dépôt public se situe à l’URL : <https://bitbucket.org/yodice/pacman-rezurexion>

Afin de cloner le dépôt avec les sous-module, il faut exécuter la commande :

[~]$ git clone --recurse-submodules https://bitbucket.org/yodice/pacman-rezurexion

# BASE DE DONNEES

## Technologie

Un fichier de base de données local ***SQLite3*** est utilisé pour stocker/modifier/récupérer les informations sur les joueurs.

## Modèle de données

## Localisation

Le fichier se nomme *scores.db* et est situé dans le dossier *resources* à la racine de l’exécutable.

# DOCUMENTATION DU CODE

## Technologie

***Doxygen*** est utilisé pour générer la documentation. Le *Doxyfile* se situe dans le dossier racine du projet.

## Génération

Il faut exécuter en ligne de commande dans le dossier contenant le *Doxyfile* :

[./pacman-rezurexion/]$ doxygen Doxyfile

## Emplacement documentation

La documentation aux formats Latex et HTML sont stockés dans le dossier doc situé à la racine du dépôt.

# LIVRAISON DE L’EXECUTABLE

### Technologie

L’installeur est crée à l’aide de ***INNOSetup***.

### Fichiers inclus

Le dossier bin/Release est le dossier où les fichiers déployés vont être cherchés.

Il y a tout le dossier bin/Release/resources ainsi que les librairies partagées de SFML (.dll pour Windows, .so pour Linux) situées à la racine de l’exécutable.