

Documentation :

Pour implémenter une librairie graphique :

- Elle doit pouvoir recevoir des events et les renvoyer à travers la méthode "registerEvents" qui renvoi un "vector<strings>"
- Gérer le display de la carte, des joueurs et/ou des différents IA de la fenêtre dans une méthode "display" prenant en paramètre un "vector<vector<strings>>" et un "int"
- Ouvrir une simple fenêtre, à travers la méthode "createWindow"
- Contenir deux méthodes dites externes, create prenant en paramètre un "vector<vector<strings>>" contenant le nom de librairie, le nom de l'utilisateur... La seconde méthode, "destroy", prends en paramètre l'objet précédemment crée afin de le supprimer

La librairie graphique devra être placée dans un dossier comportant le nom de la librairie, qui se trouvera dans le dossier lib. Voici un exemple: "lib/NCURSES/Ncurses.cpp"

Pour implémenter une librairie de jeu :

- Elle doit gérer le déplacement des différentes entités du jeu
- Vérifier si le joueur a gagné ou non et le renvoyer au programme principal
- Renvoyer différentes informations tel que : le score, l'état du jeu (gagné, en cours ou perdu) et la carte de jeu. Ces méthodes devront s'appeler respectivement : "getScore" (renvoyant un "size_t"), "getState" (renvoyant "game::state") et "get Map" (renvoyant un "vector<strings>")
- Elle doit pouvoir déterminer aussi le statut du jeu (gagné, en cours ou perdu) avec une méthode s'appelant "setState" prenant en paramètre l'état du jeu ("game::state")
- Elle doit contenir aussi deux méthodes dites externes, create et destroy. La méthode externe destroy prend en paramètre l'objet précédemment crée.
- Le jeu devra avoir une carte préalablement crée, au format .txt, et mise dans le dossier (voir ci-dessous) avec les autres fichiers

La librairie de jeu devra être placée dans un dossier comportant le nom du jeu, qui se trouvera dans le dossier "games". Voici un exemple: "game/Pacman/Pacman.cpp"