Projet pyCrate

# Modalité du projet

Durée

* 5 semaines (jusqu’à la semaine 14 inclue)

Groupes

* Max 2 personnes

Évaluation :

L’évaluation sera décomposée en deux parties : qualité et algorithmes.

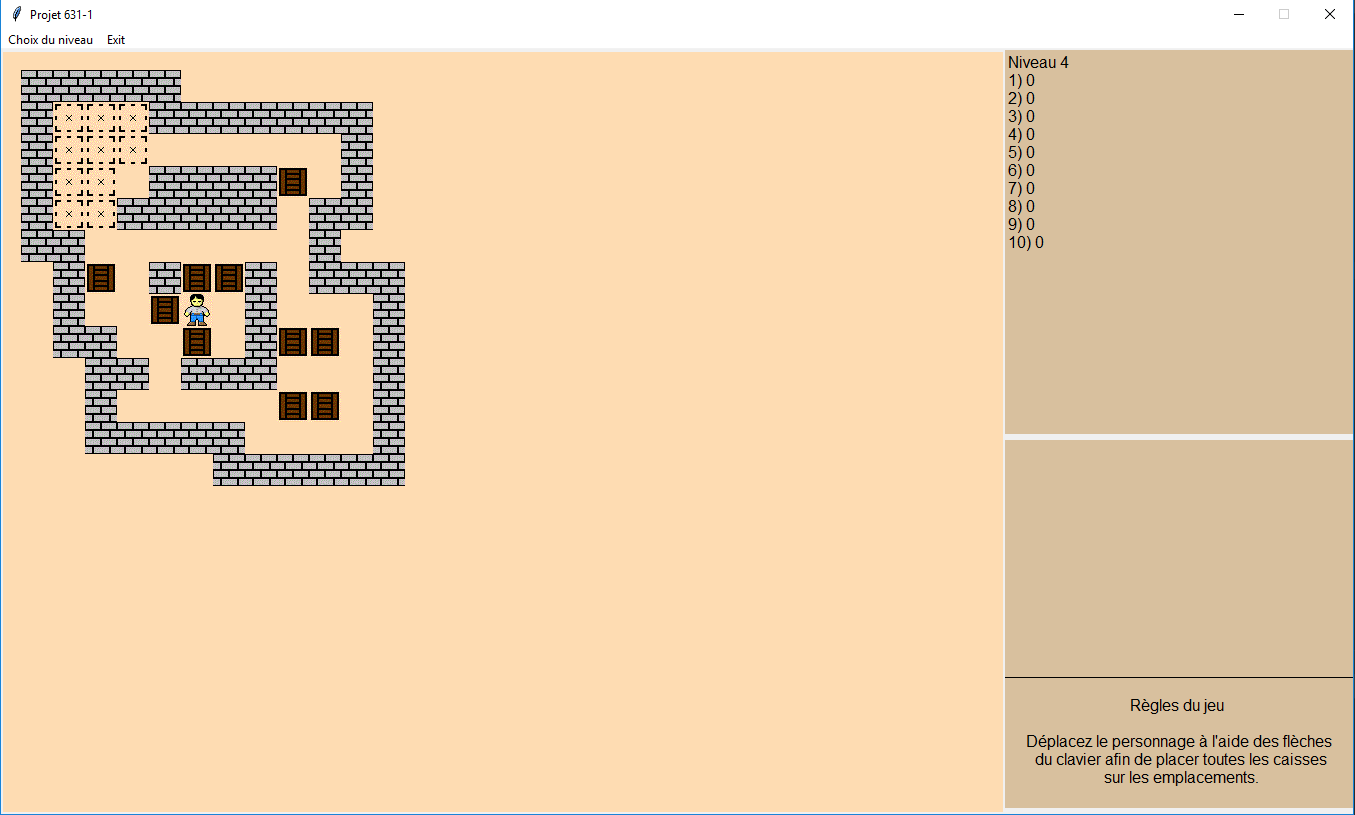
Voici les sous-parties et le poids de chacune :

* Qualité (25%)
  + Typage (variables, paramètres et types de retour) (5%)
  + Documentation du code complexe et docstring (10%)
  + Vérification de paramètres (10%)
* Algorithmes (75%)
  + Fonctionnement correct (35%)
  + Algorithme approprié (25%)
  + Décomposition de fonctions (15%)

## Temps à disposition

* Heures de laboratoire

## Description du projet

L’objectif de ce projet est de mettre à profit toutes les notions apprises lors des cours du module 631-1. Le projet en question est un petit jeu qui se présente sous la forme suivante : 

## L’objectif du jeu

Le but du jeu est de placer toutes les caisses sur les emplacements prévus à cet effet. Cependant, le personnage ne peut pas tirer une caisse, seulement la pousser. Il faudra donc réfléchir à une façon d’atteindre cet objectif sans bloquer les caisses.



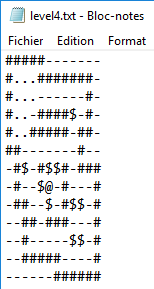
### Quelques règles

* Le joueur ne peut pas tirer une caisse, seulement la pousser
* Le joueur ne peut pas pousser une caisse si un mur est derrière la caisse
* Le joueur ne peut pas pousser une caisse si une autre caisse est derrière la première caisse
* Le joueur ne peut pas traverser les murs
* Le joueur peut marcher sur les cibles.
* Une caisse se trouvant sur une cible peut être déplacée si besoin
* Lorsque toutes les caisses sont sur des cibles, la partie est terminée

## Instructions

### Semaine 10 : Charger la carte et Condition de jeu

Dans le dossier niveaux se trouve 4 fichiers texte. Chaque fichier texte contient la description d’un niveau détaillé comme ceci :



C:\Users\jerome.humbert\AppData\Local\Microsoft\Windows\INetCache\Content.Word\wall.gif

# : représentant un mur

C:\Users\jerome.humbert\AppData\Local\Microsoft\Windows\INetCache\Content.Word\floor.gif

- : représentant du vide

C:\Users\jerome.humbert\AppData\Local\Microsoft\Windows\INetCache\Content.Word\dock.gif

. : représentant des cibles

C:\Users\jerome.humbert\AppData\Local\Microsoft\Windows\INetCache\Content.Word\box.gif

$ : représentant des caisses

C:\Users\jerome.humbert\AppData\Local\Microsoft\Windows\INetCache\Content.Word\worker.gif

@ : représentant le joueur

Pour charger la carte, il faudra donc lire le fichier sélectionné caractère par caractère et remplir les structures de données permettant ensuite l’affichage de la carte.

Il est donc demandé dans cette période de développer les fonctions charger\_niveau et jeu\_en\_cours

### Semaine 11-12 : Mouvement

Vous serez amenés à développer la fonction mouvement et effectuer\_mouvement permettant de déplacer le personnage.

Vous devez prendre en compte les « Quelques règles » listées ci-dessus lors du développement des mouvements.

### Semaine 13 : Scores

Finalement, dans le répertoire scores, fichier scores.txt, se trouvent les scores pour chaque niveau. Ceux-ci sont stockés sous cette forme :

<numNiveau>;score1;score2;score3;score4;score5;score6;score7;score8;score9;score10;

Exemple pour le niveau 1 : 1;7939;7889;7837;7837;7638;7638;6490;500;350;234;

4 fonctions seront à développer pour la gestion des scores :

1. chargement\_score  
   Fonction chargeant les scores depuis un fichier.txt et les stockent dans un dictionnaire.
2. maj\_score  
   Fonction mettant à jour l'affichage des scores en stockant dans un str l'affichage visible sur la droite du jeu.

*("Niveau x*

* 1. *7699*
  2. *... ").*

1. sauvegarde\_score  
   Fonction enregistrant un nouveau score réalisé par le joueur. Le calcul de score est le suivant :

*score\_base - (temps actuel - temps initial) - (nombre de coups \* valeur d'un coup)*

Ce score est arrondi sans virgule et stocké en tant que int.

1. update\_score\_file  
   Fonction qui va écrire les scores dans le fichier scores.txt. Cette fonction est appelée lors de la fermeture de l’application

## Fonctions et outils fournis

Dans le script outils.py vous trouverez toutes les fonctions à votre disposition. Voici la liste des fonctions que vous trouverez :

* creer\_image()
* creer\_mur()
* creer\_caisse()
* creer\_cible()
* creer\_personnage()
* creer\_case\_vide()
* coordonnee\_x()
* coordonnee\_y()
* est\_egal\_a()

Pour comprendre ce qu’elles font, référez-vous à leur docstring. Pour les utiliser, vous pouvez directement les appeler dans votre code (en faisant attention de bien utiliser leurs paramètres dans le bon ordre).

liste\_image : [] Vous l’utiliserez uniquement lorsque vous avez besoin de la fonction creer\_image(). Chacun de ses index fait référence à une image :

[0] : image du mur

[1] : image de la cible

[2] : image de la caisse

[3] : image de la caisse sur une cible

[4] : image du joueur

[5] : image du joueur sur la cible

[6] : image du sol

Donc si je souhaite remplacer l’image par l’image du joueur -> liste\_image[4]

dict\_score qui est le dictionnaire stockant les scores. Celui-ci est structuré de cette manière : key = numéro du niveau, value = tableau de scores ordonnés

ainsi : { 1 : [6049, 3493, 2322,1982,0,0,0,0,0,0]}