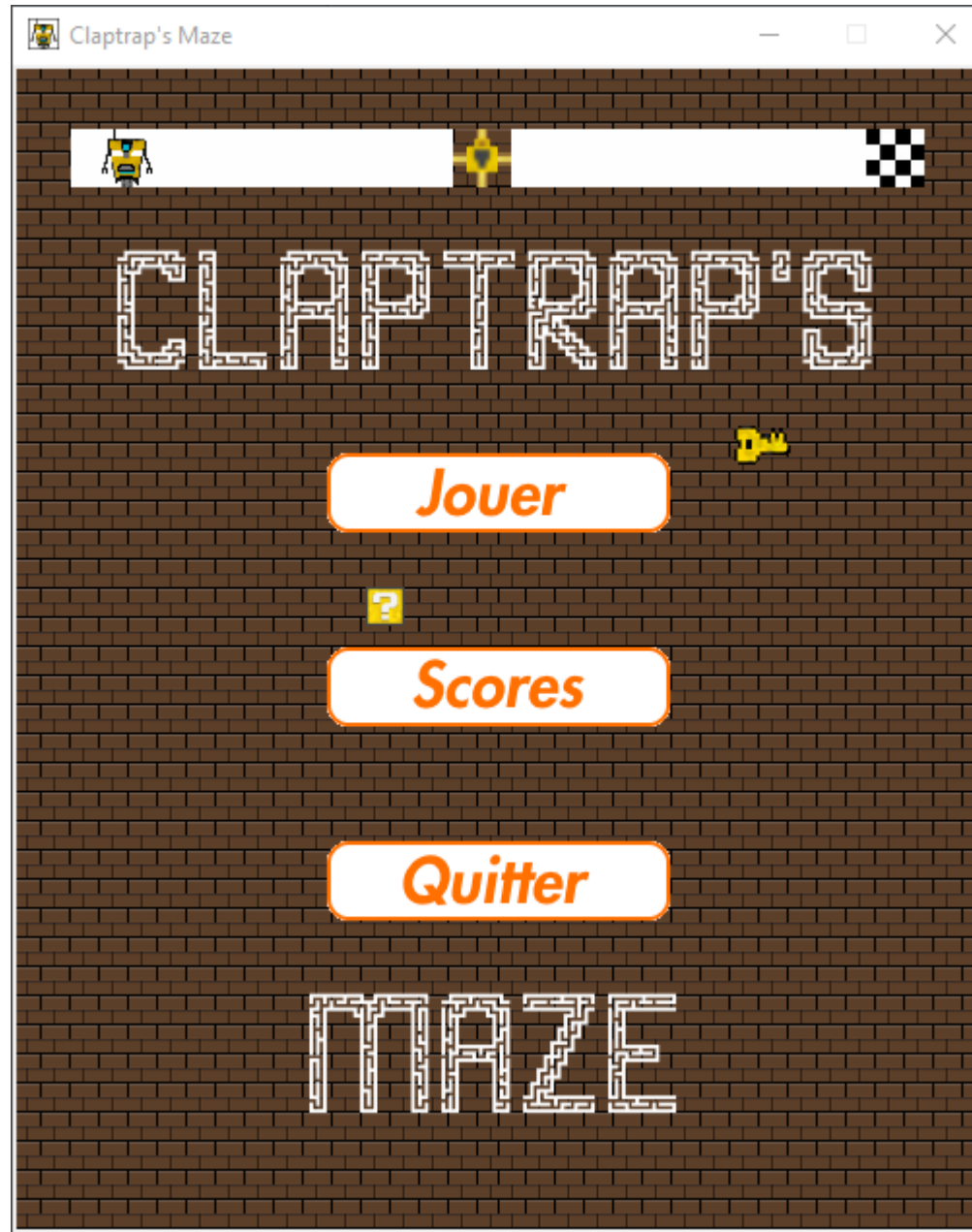


# Projet de Damien Clauzon et Julien Fleury



# Objectifs

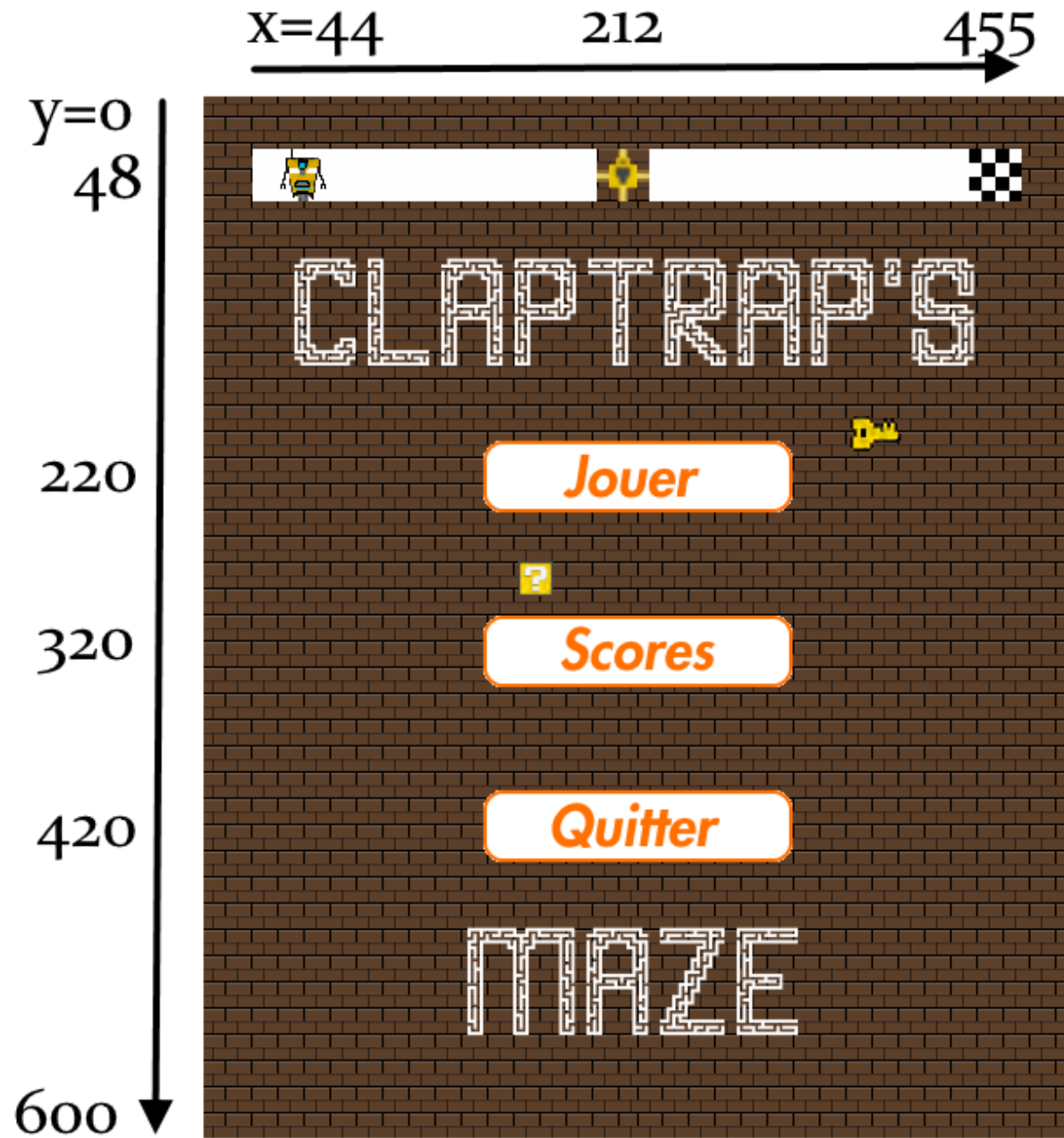
- Respecter l'idée de labyrinthe
- Ajouts par rapport à un labyrinthe classique
- Système de classement

# Planning de la programmation

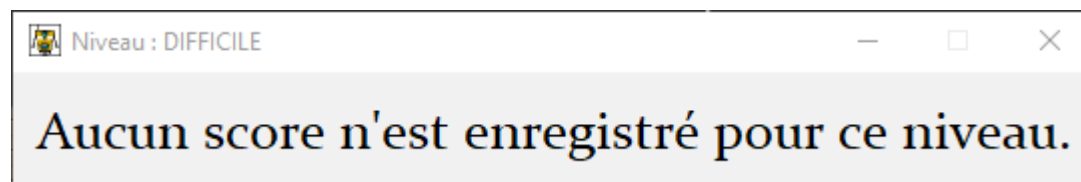
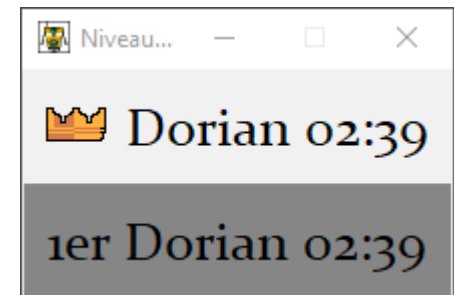
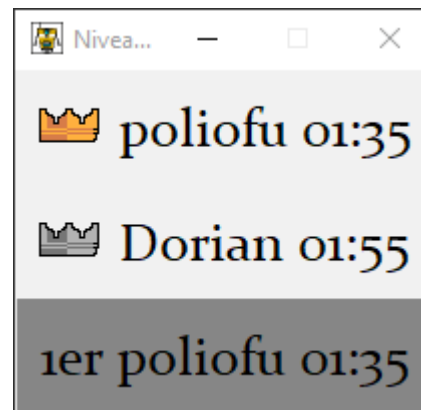
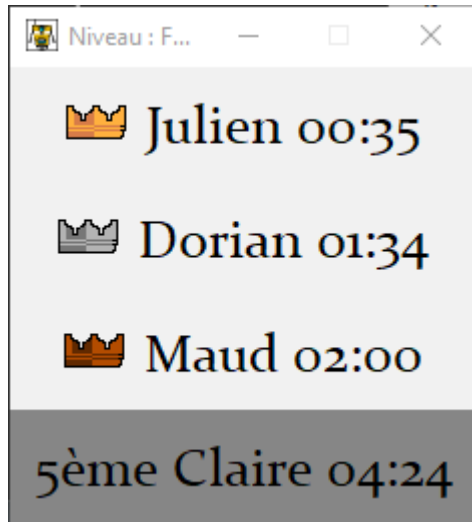
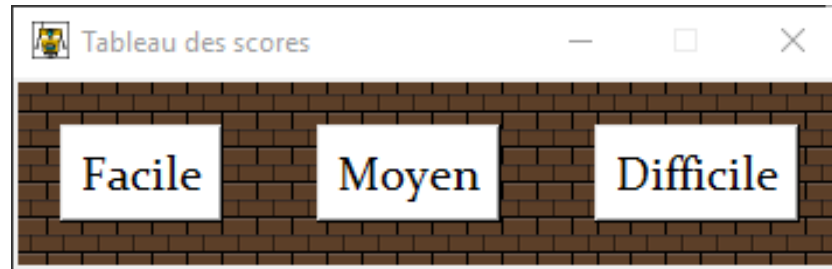
- Noyau du programme (déplacement, génération de labyrinthe)
- Éléments supplémentaires
- Tableau des scores
- Interface graphique et menu interactif

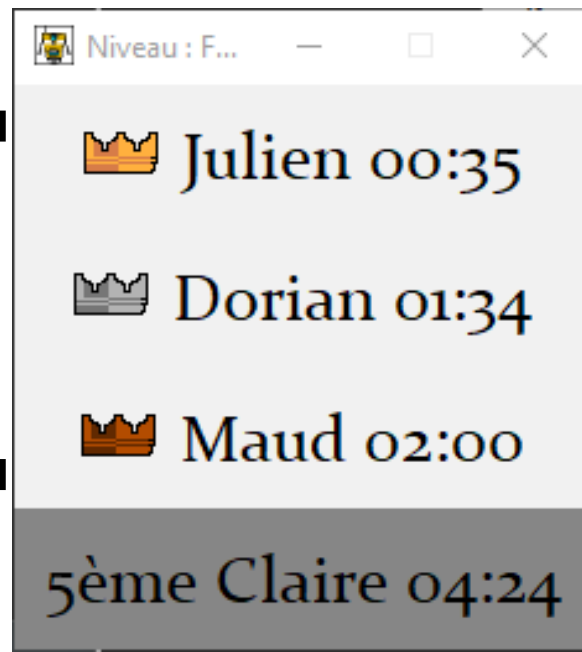
# Menu interactif





# Tableau des scores





facile - Bloc-notes

Fichier Edition Format Affichage ?

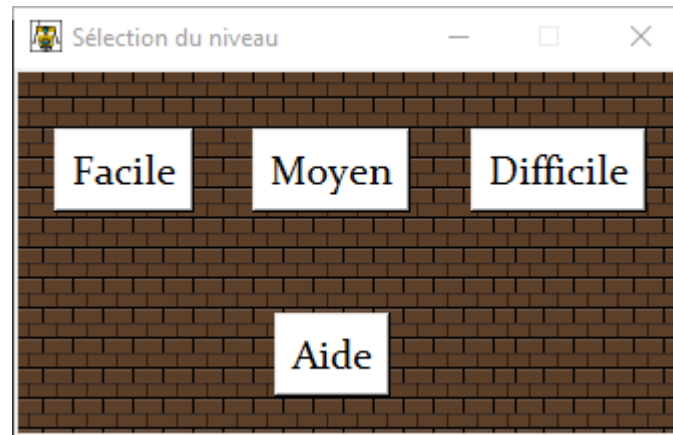
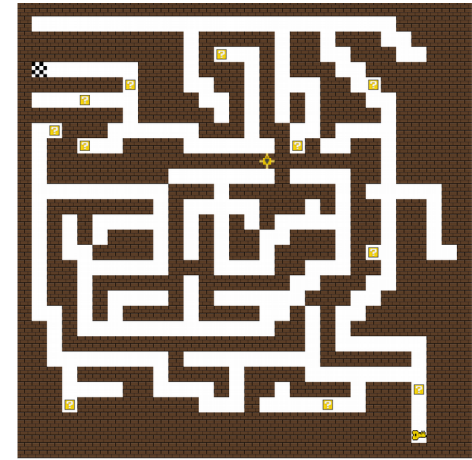
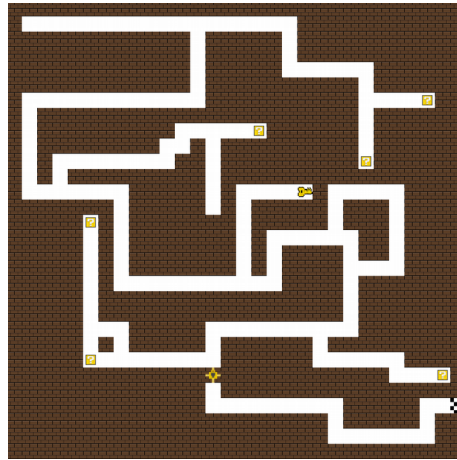
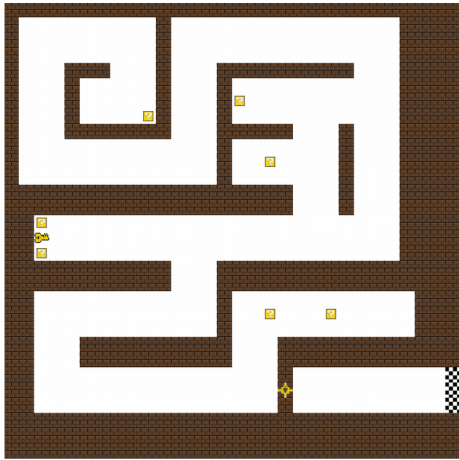
Dorian  
01:34  
Julien  
00:35  
Jonathan  
02:22  
Maud  
02:00  
Claire  
04:24

dernier-facile - Bloc-notes

Fichier Edition Format Affichage ?

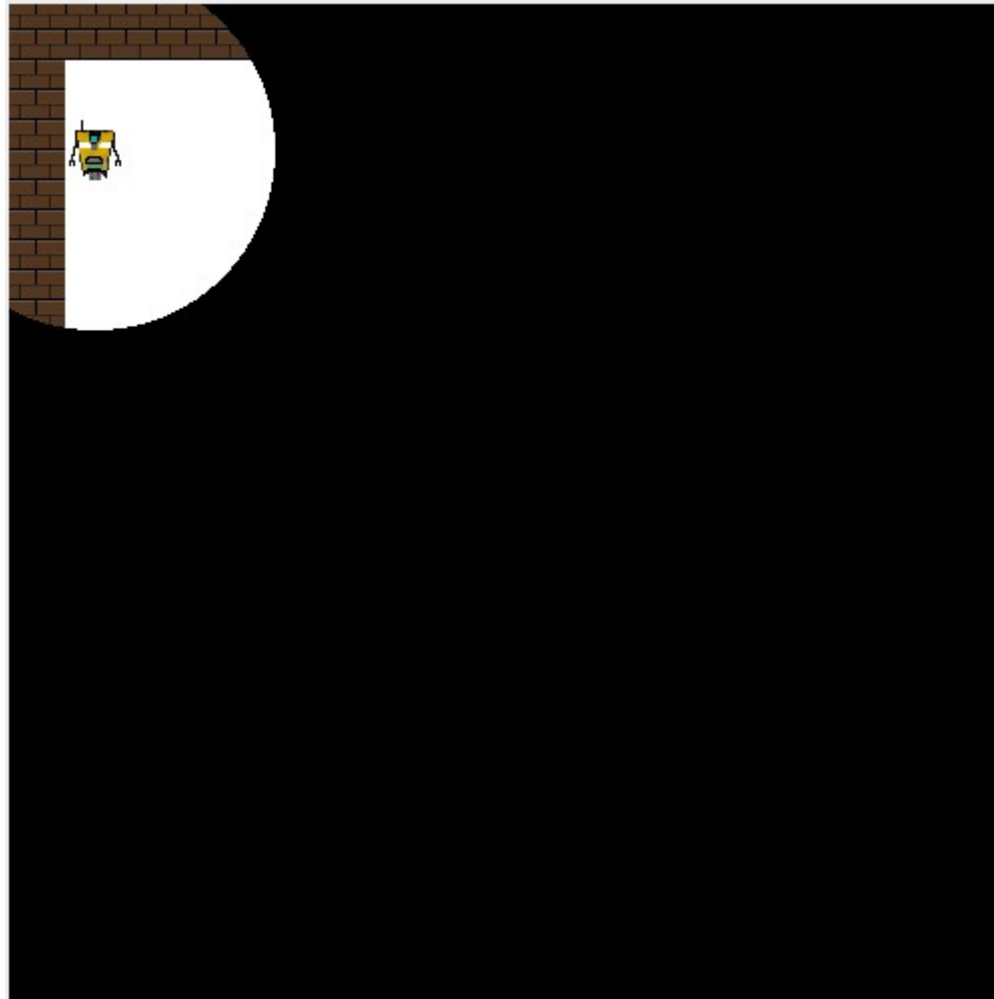
Claire  
04:24  
5

# Labyrinthes

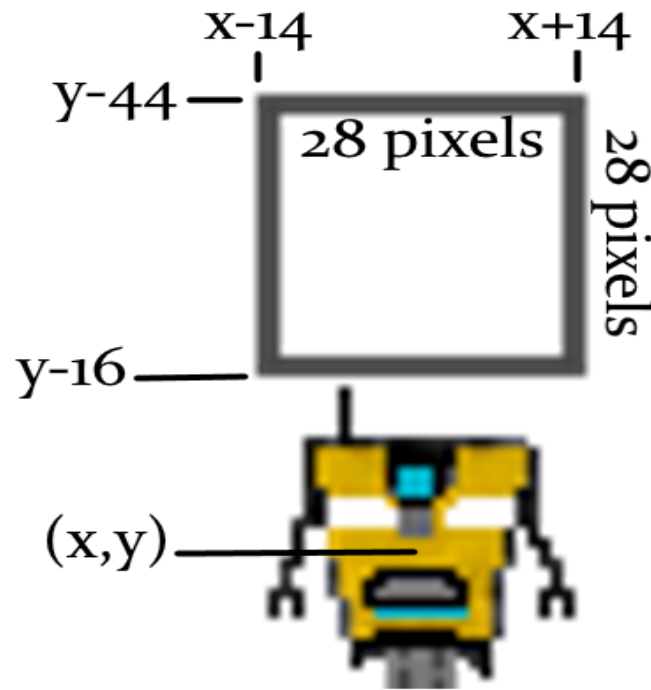




# Exemple de labyrinthe



## Exemple : un mur dans le rectangle



`C.find_overlapping(...)` → (124,425)

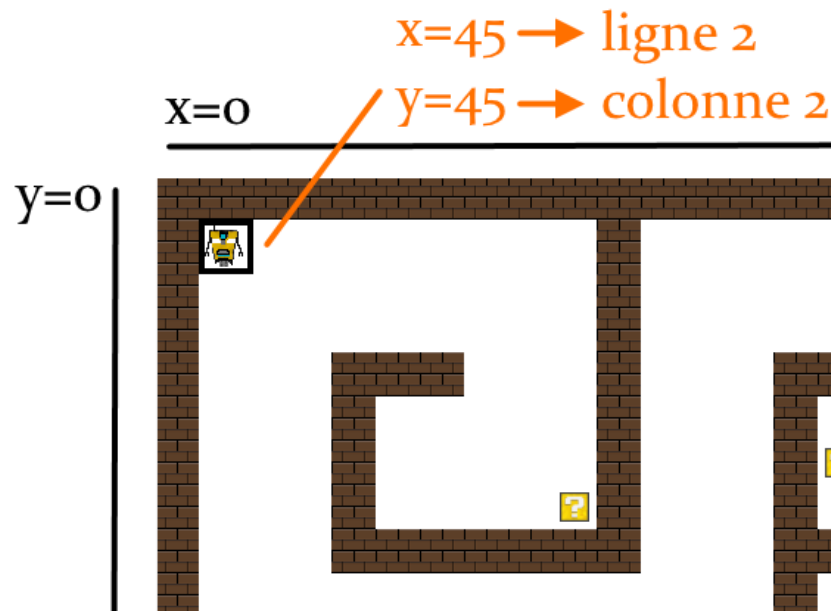
`C.gettags(124)` → ('mur',)

`C.gettags(425)` → ('Personnage',)

`C.gettags(collisionU[o])` → ('mur',)

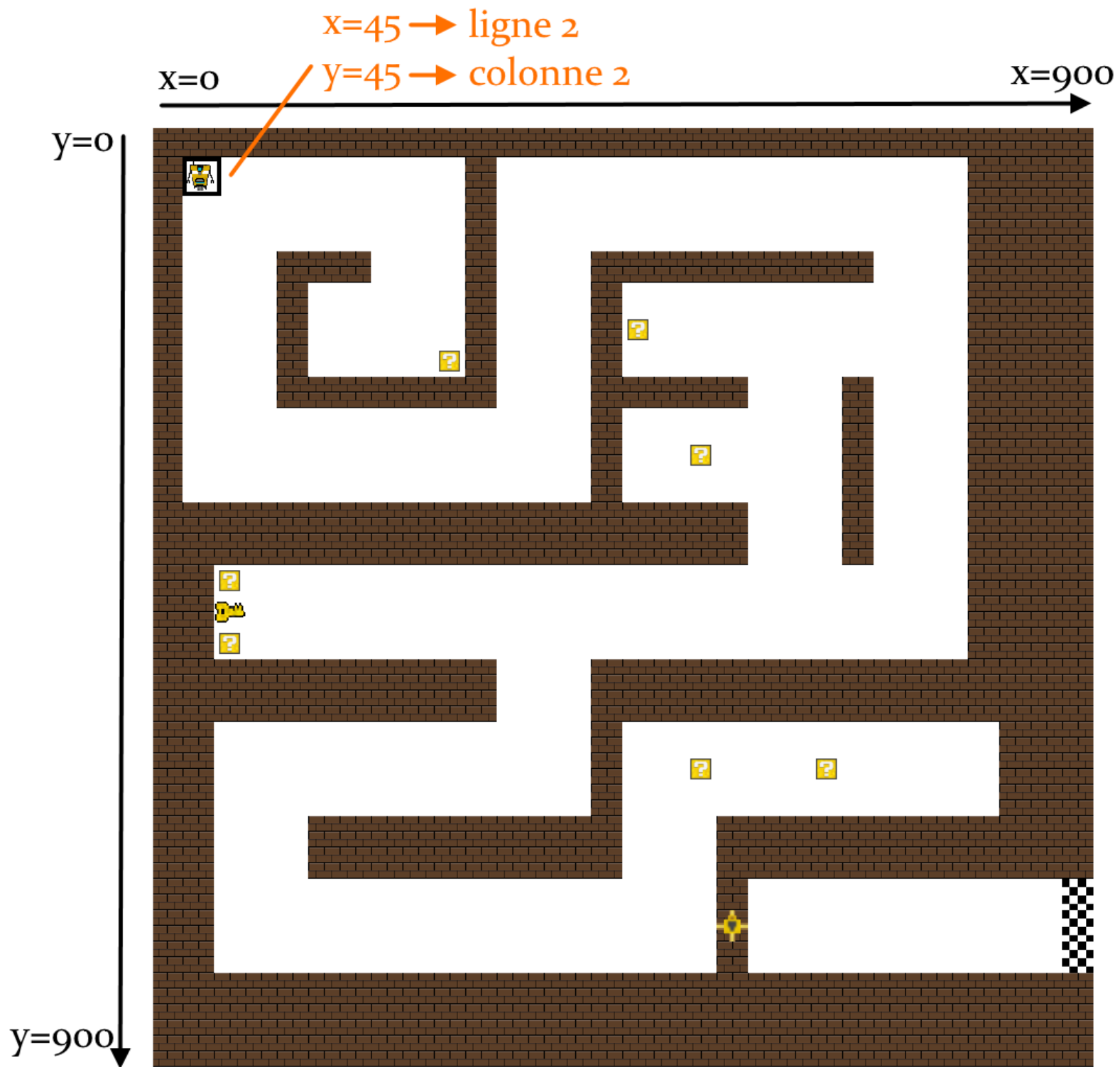
```
def move(event):
    global x,y,BonusDestroy
    if event.keysym=='Up':

        collisionU=C.find_overlapping(x+14,y-16,x-14,y-44)
        bloc0=C.gettags(collisionU[0])
        if "mur" not in bloc0 and "porte" not in bloc0:
            y=y-30
            C.move(perso,0,-30)
            if "goal" in bloc0:
                Lab.destroy()
                Lab.quit()
                goal()
```



```
def Labyrinthe():
    [...]
    for y in range(0, hauteur):
        for x in range(0, largeur):
            goal="+" + str(x) + "+"
            bonus="/" + str(x) + "/"
            piege="-" + str(x) + "-"
            vide="," + str(x) + ","
            clef="!" + str(x) + "!"
            porte=" " + str(x) + " "
            # Cree un bloc d'arrivee lorsque "+x+" est ecrit
            if goal in ligne[y]:
                C.create_image(15+30*x, 15+30*y, image=Arrivee, tags="goal")
            # Cree un bloc de bonus lorsque "/x/" est ecrit
            elif bonus in ligne[y]:
                BonusNombre+=1
                C.create_image(15+30*x, 15+30*y, image=Bonus, tags="bonus%s"%BonusNombre)
            # Cree un bloc de piege lorsque "-x-" est ecrit
            elif piege in ligne[y]:
                C.create_image(15+30*x, 15+30*y, image=Bonus, tags="piege")
            # Cree une clef lorsque "!x!" est ecrit
            elif clef in ligne[y]:
                C.create_image(15+30*x, 15+30*y, image=Clef, tags="clef")
            # Cree une porte lorsque " x " est ecrit
            elif porte in ligne[y]:
                C.create_image(15+30*x, 15+30*y, image=Porte, tags="porte")
            # Cree un carre lorsque ",x," n'est pas ecrit
            elif vide not in ligne[y]:
                C.create_image(15+30*x, 15+30*y, image=Mur, tags="mur")
```

```
#1# (premiere ligne)
#2# ,1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,
#3# ,1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,
#4# ,1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,
#5# ,1,2,3,7,8,9,11,12,13,23,24,25,
#6# ,1,2,3,5,6,7,8,9,11,12,13,15,16,17,18,19,20,21,22,23,24,25,
#7# ,1,2,3,5,6,7,8,9,11,12,13,15,16,17,18,19,20,21,22,23,24,25,/15/
#8# ,1,2,3,5,6,7,8,9,11,12,13,15,16,17,18,19,20,21,22,23,24,25,/9/
#9# ,1,2,3,11,12,13,19,20,21,23,24,25,
```

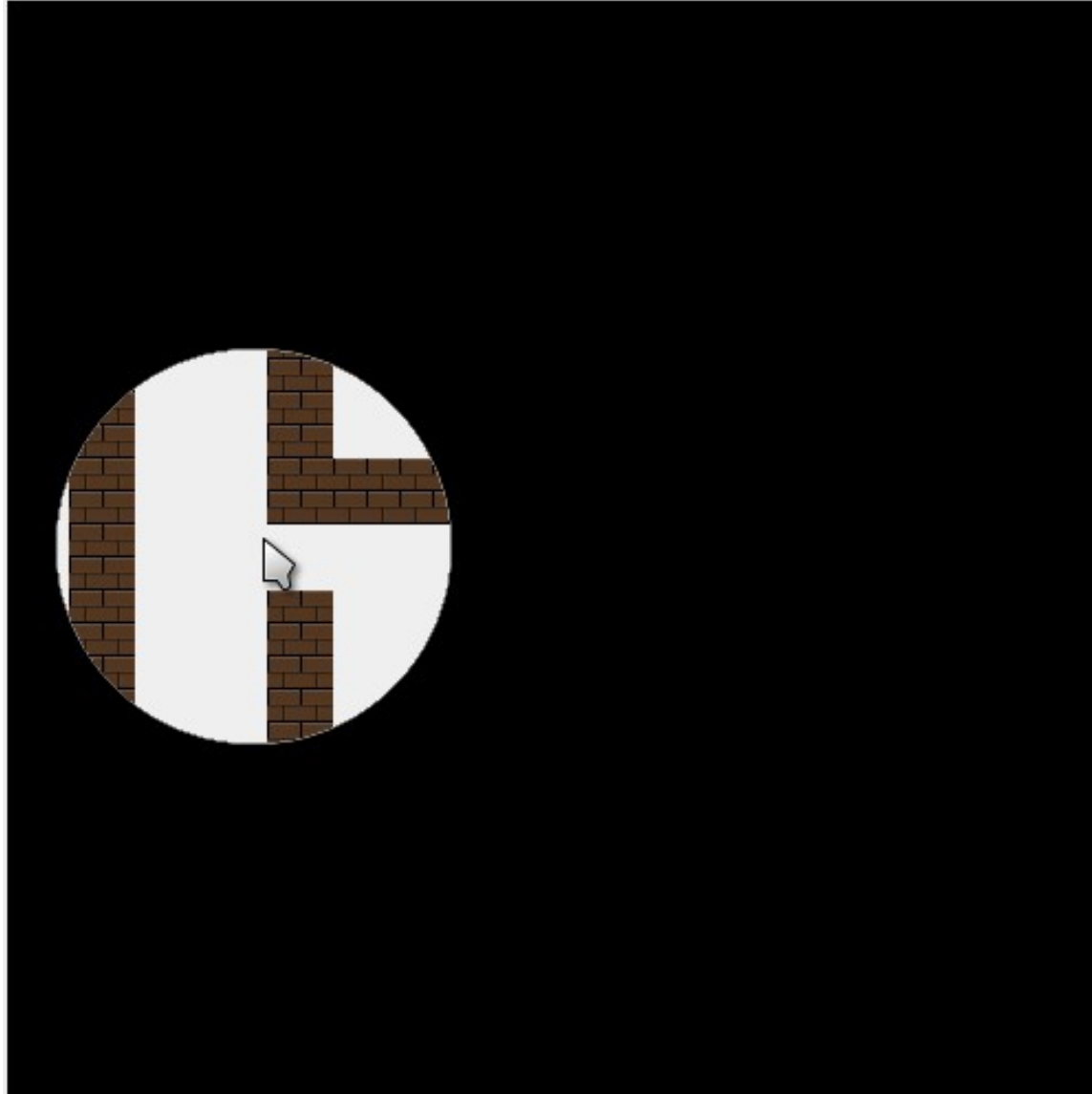


# Problèmes du programme actuel

- Ajouter un autre labyrinthe prendrait du temps (boutons...)
- Tableau des scores limité à 3 joueurs
- Résolution non-adaptée à tous les ordinateurs (min 900x900)
- Légers ralentissements lors de la prise d'un bonus

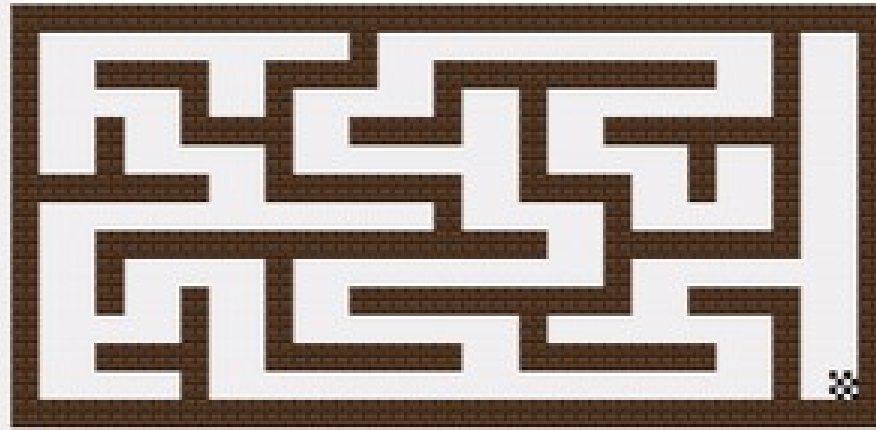
# Prolongements possibles

## Contrôle à la souris



# Prolongements possibles

## Labyrinthe aléatoire



# Prolongements possibles

- Augmenter le nombre de labyrinthes
- Composition et intégration d'une musique
- Section « création de labyrinthe »
- Ouvrir des labyrinthes avec internet



**Bilan**