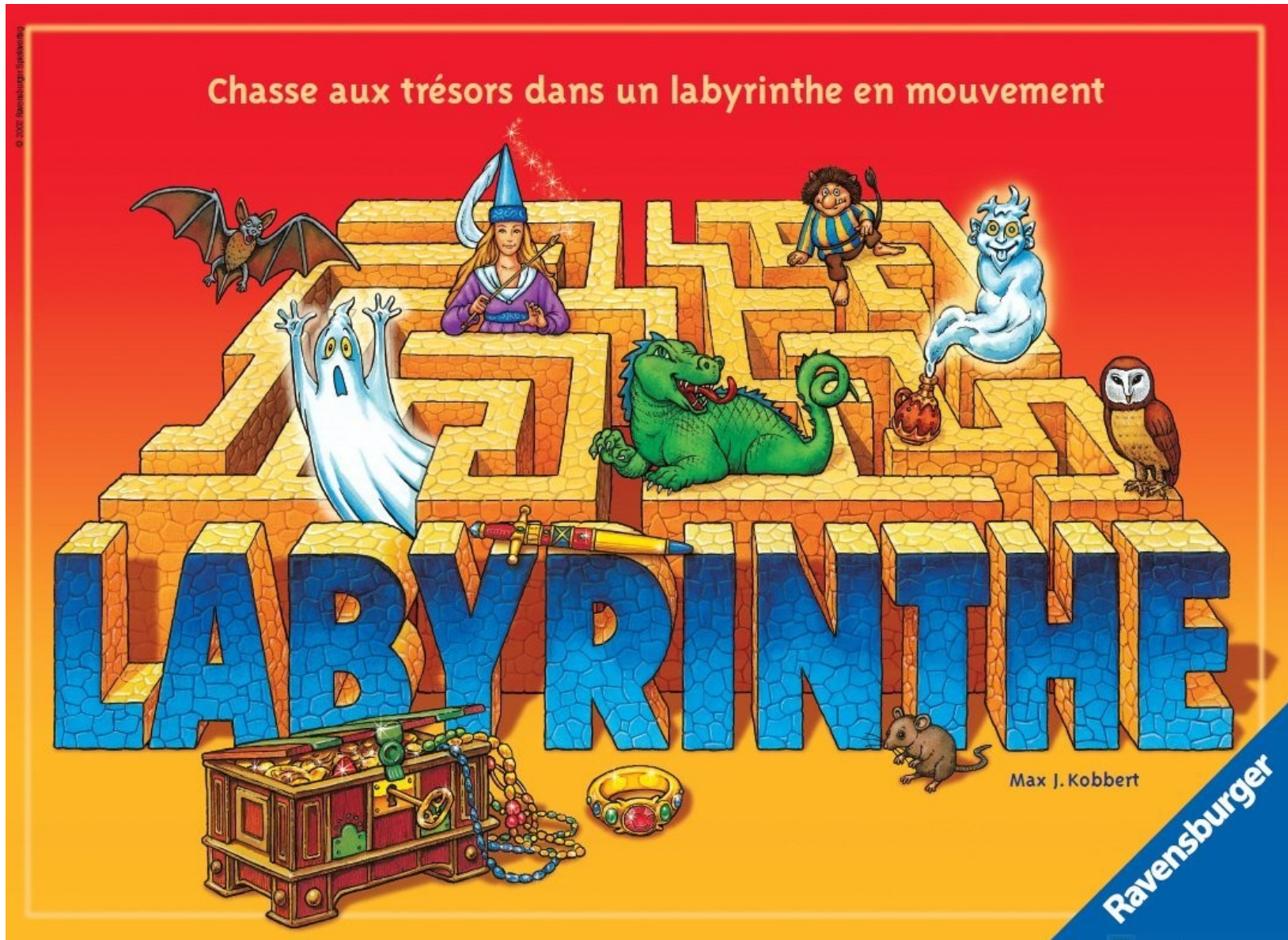


Projet Labyrinthe



Sommaire

I) Répartition du travail

II) Les structures de données

- Cartes

- Joueurs

- Matrice

- Labyrinthe

III) Méthodes de test

IV) Principaux algorithmes

V) Bugs et extensions

Répartition du travail

BARBIERI Baptiste

MOISY Marvin

18/01/15 Matrice.py + carte.py + joueur.py + labyrintheTexte

19/01/15 labyrinthe.py + débuggage

20/01/15 Version objet + pdf + diapo oral

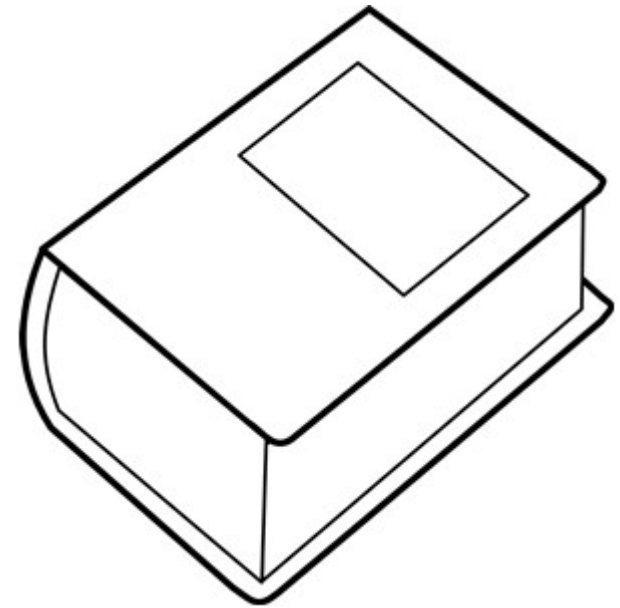
Les structures de données

Carte : Dictionnaire

Joueur : Dictionnaire

Matrice : Liste de listes

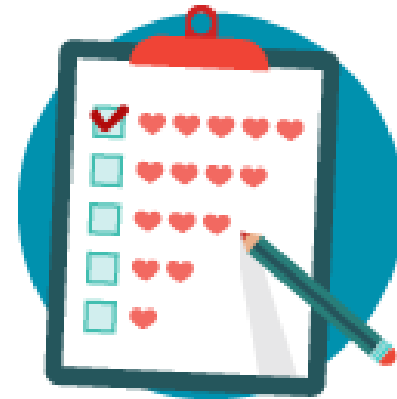
Labyrinthe : Dictionnaire



Méthodes de test

Lancer le jeu

Test sur les matrices avec la fonction
afficheMatrice



Principaux algorithmes

Algorithme de recherche de chemin :

```
def marquageDirect(calque,mat,val,marque):  
    ans = False  
    for lig in range(mat.getNbLignes()):  
        for col in range(mat.getNbColonnes()):  
            if (calque.getVal(lig, col) == 0 and  
                ((passageEst(mat.getVal(lig, col), mat.getVal(lig, col+1)) and val == calque.getVal(lig, col+1))  
                 or (passageOuest(mat.getVal(lig, col), mat.getVal(lig, col-1)) and val == calque.getVal(lig, col-1))  
                 or (passageNord(mat.getVal(lig, col), mat.getVal(lig-1, col)) and val == calque.getVal(lig-1, col))  
                 or (passageSud(mat.getVal(lig, col), mat.getVal(lig+1, col)) and val == calque.getVal(lig+1, col))  
                ))):  
                calque.setVal(lig, col, marque)  
            ans = True  
    return ans
```

```
def passageEst(carte1,carte2):  
    if carte1 == None or carte2 == None: return False  
    return (not carte1.murEst()) and (not carte2.murOuest())
```

Principaux algorithmes

Algorithme de décalage des lignes et colonnes :

```
def decalageLigneADroite(self, numLig, nouvelleValeur=0):  
    self.matrice[numLig].insert(0, nouvelleValeur)  
    return self.matrice[numLig].pop()  
  
# decale la colonne numCol d'une case vers le haut en insérant  
# dans la case ainsi libérée  
# la fonction retourne la valeur de la case "ejectée" par le c  
def decalageColonneEnHaut(self, numCol, nouvelleValeur=0):  
    ans = self.getVal(0, numCol)  
    for i in range(1, self.getNbColonnes()):  
        self.setVal(i-1, numCol, self.getVal(i, numCol))  
    self.setVal(self.getNbLignes()-1, numCol, nouvelleValeur)  
    return ans
```

Bugs et extensions

