**Algorithme fonction clique Droit**

Si selectiontype = 1 :

Si la première coordonnée du pion 1 est différent de 8

coord →[ pion 1[0], pion1[1]+1]

Si pion 2 différent de coord et qu'il n'y a pas de barrières

select\_case→ coord

confirmation →1

actualiserFenetre

Sinon

Si select\_barriere[2] différent de 7

select\_barriere[2] → select\_barriere[2]+1

confirmation →2

actualiserFenetre

L'algorithme est le même pour les fonction clique gauche, clique bas et clique haut, seulement certaines valeur changes.

**Algorithme fonction clique Entrer**

Si confirmation = 1

pion1 →select\_case

sinon si confirmation = 2

Si select\_barriere[0]:

Si barrierehorizontale(select\_barriere[1], select\_barriere[2]) = 0

barrierehorizontale(select\_barriere[1], select\_barriere[2]) = 1

Si tour = 1

barriere\_restante1 → barriere\_restante1-1

Sinon

barriere\_restante2 → barriere\_restante2 -1

Sinon

Si barriereverticale (select\_barriere[1], select\_barriere[2]) = 0

barriereverticale (select\_barriere[1], select\_barriere[2]) = 1

Si tour = 1

barriere\_restante1 → barriere\_restante1-1

Sinon

barriere\_restante2 → barriere\_restante2 -1

**Algorithme fonction clique espace**

Si select\_barriere[0] = 0

select\_barriere[0] = 1

Sinon

select\_barriere[0 ]= 0

**Algorithme fonction écoute**

Fenetre.bind('<Right>',cliqueDroite)

Fenetre.bind('<Left>',cliqueGauche)

Fenetre.bind('<Down>',cliqueBas)

Fenetre.bind('<Up>',cliqueHaut)

Fenetre.bind('<Return>',cliqueEntrer)

Fenetre.bind('<space>',cliqueEspace)

**Algorithme fonction victoire**

Si pionA2[0] = 8

retourner 2

Sinon Si pionA1[0] = 0

retourner 1

Sinon

retourner 0

**Algorithme fonction jeu**

Si victoire = 0

Si joueur = 0

tour →2

message → "Au tour du joueur 2"

Si barrieres\_restantes2 = 0

bouton → Faux

Sinon

bouton→ Vrai

confirmation→ 0

selectionType → 0

actualiserFenetre

Sinon

tour→ 1

message → "Au tour du joueur 1"

Si barrieres\_restantes1 =0

bouton → Faux

Sinon

bouton→ Vrai

confirmation→ 0

selectionType → 0

actualiserFenetre()

Sinon

message→"Victoire joueur"

tour→0

actualiserFenetre()

**Algorithme fonction quoridor**

case → creer\_matrice(9,9)

barrieres\_verticales = creer\_matrice(8, 9)

barrieres\_horizontales = creer\_matrice(9, 8)

creerFenetre()

ecouteClavier()

jeu(1)

**Algorithme fonction generer\_image**

pixels →creer\_matrice(69,69)

Pour a allant de 1 à 9

Pour b allant de 1 à 9

start → [8\*a,8\*b]

Pour c allant de 1 à 5

for d allant de 1 à 5

pixels[start[0]+c][start[1]+d] →"N"

Si select\_case = [a,b]

Pour e allant de 1 à 5

pixels[start[0]+e][start[1]] → "O"

pixels[start[0]+e][start[1]+4]→"O"

Pour f allant de 1 à 5

pixels[start[0]][start[1]+f] → "O"

pixels[start[0]+4][start[1]+f]→"O"

Si PionA2 = [a,b]

Pour g allant de 1 à 3

pixels[start[0]+2][start[1]+1+g]→"R"

Pour h allant de 1 à 3

pixels[start[0]+1+h][start[1]+2]→"R"

Sinon Si PionA1 = [a,b]

Pour i allant de 1 à 3

pixels[start[0]+2][start[1]+1+i]→"B"

Pour j allant de 1 à 3

pixels[start[0]+1+j][start[1]+2]→"B"

Pour k allant de 1 à 8

Pour l allant de 1 à 8

Si barrieres\_verticales[k][l]

start\_b → [k\*8,l\*8+5]

Pour m allant de 1 à 3

Pour n allant de 1 à 13

pixels[start\_b[0]+n][start\_b[1]+m]→"V"

Si select\_barriere[0] = 0

Si select\_barriere n'est pas égale à [0,-1,-1]

Si select\_barriere[1] = k et que select\_barriere[2] = 1

start\_b → [k\*8,l\*8+5]

Pour m allant de 1 à 3

Pour n allant de 1 à 13

pixels[start\_b[0]+n][start\_b[1]+m]→"T"

Pour o allant de 1 à 8

Pour p allant de 1 à 8

Si barrieres\_horizontales[o][p]

start\_b → [o\*8+5,p\*8]

Pour q allant de 1 à 3

Pour r allant de 1 à 13

pixels[start\_b[0]+q][start\_b[1]+r]→"V"

Si select\_barriere[0]

Si select\_barriere n'est pas égale à [1,-1,-1]

Si select\_barriere[1] = o et select\_barriere[2] = p

start\_b → [o\*8+5,p\*8]

Pour q allant de 1 à 3

Pour r allant de 1 à 13

pixels[start\_b[0]+q][start\_b[1]+r]→"T"

retourner pixels

**Algorithme fonction creer\_image**

Ouvrir le fichier plateau\_P3.ppm en mode écriture

écrire dans le fichier( P3, 69 69 , 255)

Pour a allant de 1 à 69

ligne→ ""

Pour b allant de 1 à 69

couleur→ pixels [a][b]

Si couleur = N

ligne→ligne + "0 0 0"

Sinon Si couleur = O

ligne→ligne + "255 125 0"

Sinon Si couleur = R

ligne→ligne + "255 0 0"

Sinon Si couleur = B

ligne→ligne + "0 0 255"

Sinon Si couleur = V

ligne→ligne + "80 255 0"

Sinon Si couleur = T

ligne→ligne + "40 221 227"

Sinon

ligne→ligne+"255 255 255"

ligne→ ligne +"n"

écrire dans le fichier(ligne)

fermer le fichier

**Algorithme fonction barrierepresente** (vérification de barrière présente)

barriere\_presente → Faux

coord\_barriere1 → [-1,-1]

coord\_barriere2 → [-1,-1]

Si case\_actuelle[0] différent de case\_future[0]

Si case\_actuelle[0] < case\_future[0]

coord\_barriere1 → [case\_actuelle[0],case\_actuelle[1]]

Si case\_actuelle[1] différent de 0

coord\_barriere2 →[case\_actuelle[0],case\_actuelle[1]-1]

Sinon

coord\_barriere1 → [case\_actuelle[0]-1,case\_actuelle[1]]

Si case\_actuelle[1] différent de 0

coord\_barriere2 → [case\_actuelle[0]-1,case\_actuelle[1]-1]

Si case\_actuelle[1] =8

coord\_barriere1 →[-1,-1]

Si coord\_barriere1 différent de [-1,-1]

Si barrieres\_horizontales[coord\_barriere1[0]][coord\_barriere1[1]]

barriere\_presente → Vrai

Si coord\_barriere2 différent de [-1,-1]

Si barrieres\_horizontales[coord\_barriere2[0]][coord\_barriere2[1]]

barriere\_presente → Vrai

Sinon

Si case\_actuelle[1] < case\_future[1]

coord\_barriere1 → [case\_actuelle[0],case\_actuelle[1]]

Si case\_actuelle[0] différent de 0

coord\_barriere2 → [case\_actuelle[0]-1,case\_actuelle[1]]

Sinon

coord\_barriere1 → [case\_actuelle[0],case\_actuelle[1]-1]

Si case\_actuelle[0] = 8

coord\_barriere1 → [-1,-1]

Si coord\_barriere1 différent de [-1,-1]

Si barrieres\_verticales[coord\_barriere1[0]][coord\_barriere1[1]]

barriere\_presente→ Vrai

Si coord\_barriere2 différent de [-1,-1]

Si barrieres\_verticales[coord\_barriere2[0]][coord\_barriere2[1]]

barriere\_presente → Vrai

Retourner barriere\_presente