# Présentation

L'objectif de ce projet est de proposer deux représentations et méthodes de résolution du jeu Bubble Blast, qui a connu un succès relativement important sur smartphones. **Règles. Définitions utilisées (neutrons, atomes, width & height). Adresse démo.**

# Représentation 1 : utilisation de JaCoP

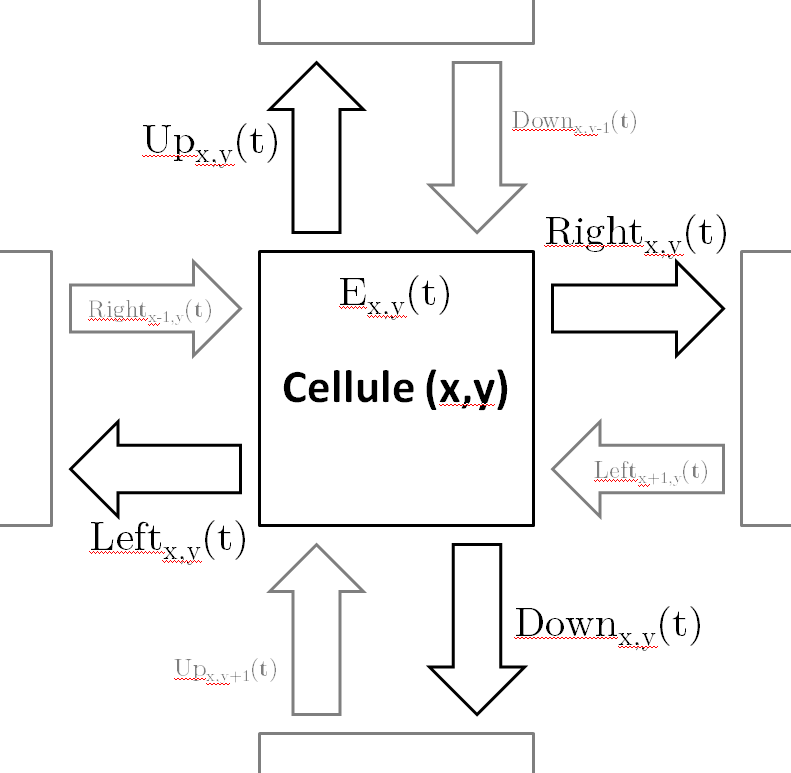
On propose ici une représentation adaptée à l'utilisation de la bibliothèque JaCoP. Elle repose sur une discrétisation du temps la plus fine possible, liée à la discrétisation de l'espace et à la vitesse de déplacement des neutrons : un intervalle de temps infinitésimal [t, t+1] est le temps nécessaire à un neutrons pour passer d'une case à la case voisine. On impose alors un instant maximal *tmax*. Chaque variable représentant l'état du jeu à un instant *t* sera dupliquée *tmax* fois, et le comportement du jeu sera implémenté par les contraintes reliant les variables de l'instant *t* à celles de l'instant *t+1*.

## Représentation

### Variables

On utilise alors les variables représentées ci-dessous (Figure 1) :

- *Ex,y(t)* représente l'énergie de l'atome contenu dans la case (*x*,*y*) à l'instant *t*. L'absence d'atome est caractérisée par la valeur : *Ex,y(t)* =*VOID*, *VOID* étant un entier caractéristique, typiquement *VOID* = -5 pour plus de praticité (cf **plus bas**).

Figure 1 : Variables utilisées

- *Dirx,y(t)*, avec *Dir* ∈ {*Left*, *Right*, *Down*, *Up*} traduit la présence d'un neutron situé à l'instant *t* dans la case (*x*,*y*) et se dirigeant dans la direction correspondante. *Dirx,y(t)* vaut 1 s'il y a un tel neutron, 0 sinon.

On ajoute les variables *Act(t)* représentant l'action du joueur : à l'instant *t*, si le joueur agit sur une case (*x,y*), alors *Act(t)* = (*x,y*) ; sinon *Act(t)* = *NOTHING* (avec *NOTHING* une valeur ne pouvant représenter une case du terrain).

Enfin, afin de pouvoir rechercher la solution optimale – c'est-à-dire celle minimisant le nombre d'actions du joueur –, on ajoute les variables *Cost(t)* : *Cost(t)* étant à l'instant *t* le nombre de fois où le joueur a agit sur le jeu.

En Java, le couple (*x,y*) sera représenté par l'entier x+y\*width. *Act(t)* est ainsi à valeur dans [0..*width*\**height*]∪{*NOTHING*} – et on peut prendre pour *NOTHING* la valeur *NOTHING* = -1. De plus, un ensemble *Var* de variables *Varx,y(t)* sera représenté par un tableau var[][]. La variable *Varx,y(t)* sera alors représentée par var[t][x+y\*width], de sorte que var[t] soit l'ensemble des valeurs de *Var* pour les différentes case à l'instant *t*.

### Contraintes

Divers phénomènes sont à représenter via les contraintes : le déplacement des neutrons, l'évolution des énergies des atomes et leurs potentielles explosions, et l'impossibilité de jouer tant qu'une réaction en chaîne est en cours. Il faut de plus imposer les conditions initiales du problème, ainsi que la condition de succès : à la fin, il ne doit plus rester d'atome. Enfin, il faut définir la fonction de coup, i.e. compter le nombre d'actions du joueur.

*• Évolution des énergies :*

On définit *Actx,y(t)* comme étant l'énergie ajoutée par le joueur en (*x,y*) à l'instant *t*.

On définit alors l'énergie brute à l'instant *t+1* par :

*Ebx,y(t+1)* := *Ex,y(t)* + *Leftx+1,y(t)* + *Rightx-1,y(t)* + *Upx,y+1(t)* + *Downx,y-1(t)* + *Actx,y(t+1)*

( *Ebx,y(t+1)* := Ancienne énergie + Neutrons arrivants + Action potentielle du joueur )

L'explosion d'un atome à l'instant *t* se traduit par le test :

*Xx,y(t)* := [ *Ebx,y(t)* > *Emax* ]

Tandis que l'absence d'atome est donnée par le test :

*Øx,y(t)* := [ *Ex,y(t)* = *VOID* ]

*Øx,y(t)* =

L'énergie *Ex,y(t)* est alors *VOID* si *Øx,y(t)*, et *Ebx,y(t)* sinon. (Prendre *VOID* = -5 permet d'assurer *Ebx,y(t)* ≤ 0 ≤ *Emax*.)

*• Déplacement des neutrons :*

Pour une direction *Dir* ∈ {*Left*, *Right*, *Down*, *Up*}, en notant (*x'*,*y'*) la case voisine d'où proviendrait éventuellement un neutron allant dans la direction considérée, on a :

*Dirx,y*(t+1) =

En effet, s'il y a explosion, un neutron est créé ; si la case contient un atome, tout neutron est absorbé ; sinon, un neutron traverse éventuellement la case.

*• Initialisation et succès :*

On traduit aisément les conditions initiales par des contraintes sur les *Ex,y*(0), tout en mettant à 0 les *Dirx,y*(0) pour *Dir* ∈ {*Left*, *Right*, *Up*, *Down*}.

On représente la succès par la variable *Success(t)* ∈ {*Vrai*, *Faux*} :

*Success(t)* := = *Øx,y(t)*

La condition de succès se traduit alors par la contrainte : *Success(tmax)* = *Vrai.*

Pour simplifier la résolution du problème, on peut de plus ajouter les contraintes :

∀ t, *Success(t)* ⇒ *Success(t+1)*

*• Action du joueur :*

Précisons tout d'abord le lien simple entre *Act(t)* et *Actx,y(t)* :

*Actx,y(t)* = [ *Act(t)* = (*x*,*y*) }

On ne joue que lorsqu'aucune réaction en chaîne n'est en cours et que le jeu n'est pas terminé. Cela se traduit par la condition :

De plus, on ne peut agir que sur une case contenant un atome :

∀(x,y), *Øx,y(t)* ⇒ [ *Act(t)* ≠(x,y) ]

*• Fonction de coût :*

Elle est simplement incrémentée lorsque le joueur agit :

*Cost(t)* =

## Résolution

**TODO**

# Représentation 2 : ####

Ici, on envisage une discrétisation plus large du temps : un instant correspond à une réaction en chaîne complète – et le joueur joue donc à chaque instant tant qu'il reste des atomes. Ainsi, les seules variables à prendre en compte sont les énergies *Ex,y(t)* et les actions *Actx,y(t)*. Il n'y a plus de neutrons à prendre en compte, et de nombreuses variables introduites pour plus de praticité sont dorénavant superflues (*Xx,y(t)*, *Act(t)*, ...). En revanche, l'établissement des contraintes est bien plus complexe : il est nécessaire de simuler l'intégralité de la réaction en chaîne pour établir le lien entre d'une part {*Ex,y(t)* | (*x*,*y*) ∈ terrain} et l'action effectuée à l'instant *t* et d'autre part {*Ex,y(t+1)* | (*x*,*y*) ∈ terrain}.

## Représentation

### Variables

Comme dit plus haut, on ne doit maintenant ne considérer que les énergies des atomes contenus dans les cases et les actions du joueur. On utilise donc les mêmes représentations que précédemment, mais en ne préservant que les variables d'énergie *Ex,y(t)*, d'action *Act(t)* et de coût *Cost(t)*.