

# Boulder Dash

v. 1.0



## 1 Préambule

*Boulder dash* est un jeu vidéo développé par First Star Software et sorti en 1984 sur ordinateurs Atari puis sur de nombreux supports. Dans le jeu, le personnage principal, appelé Rockford, doit creuser dans des grottes pour collecter des diamants et atteindre la sortie dans un temps limité. Il doit aussi éviter différentes créatures et obstacles, en particulier des chutes de rochers et doit faire très attention à ne pas provoquer d'avalanche qui risqueraient de le tuer ou l'enfermer dans la grotte.

Chaque « niveau » est représenté par une grille en 2 dimensions de cases contenant différents types d'éléments. On peut considérer que le jeu se déroule étape par étape, à chaque étape, Rockford peut se déplacer dans une des quatre directions haut, bas, gauche, droite. Pour simuler une exécution temps réel, nous ajoutons la possibilité de ne rien faire et rester immobile à Rockford. Après chaque déplacement, le niveau est mis à jour c'est-à-dire que tous les mouvements des autres éléments du plateau sont réalisés. Les différents éléments du jeu sont présentés dans la table 1 (page suivante). Les éléments rouges sont indispensables pour une permettre de faire une partie de boulder dash et donc dans le cadre du projet.

Un niveau de boulder dash pourra être représenté en mode terminal par un tableau de caractères, comme dans l'exemple ci-dessous :

```

WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
W...X...WccccccccccccccccW.....W
W.....WccccccccccccccccW.....W
W.....WccccccccccccccccW.....W
W.....WccccccccccccccccW.....W
W.....WccccccccccccccccW.....W
W.....WccccccccccccccccW.....W
W.....WccccccccccccccccW.....W
WwwwwwwwwWddddddddddddddW.....W
WaaaaaaaaWddddddddddddddW.....W
WaaaaaaaaWddddddddddddddW.....W
WaaaaaaaaFddddddddddddddWqqq.q.q.q.W
WaaaaaaaaFddddddddddddddWq.qqq.q.q.W
WaaaaaaaaFddddddddddddddWq.q.q.q.q.W
WaaaaaaaaFddddddddddddddWq.q.q.q.q.W
WaaaaaaaaFddddddddddddddWq.q.q.qqq.W
WaaaaaaaaFddddddddddddddWq.q.q.q.q.W
WdddddddddFddddddddddddddWq.q.q.q.q.W
W.....FddddddddddddddWq.q.q.q.q.W
W      FddddddddddddddWq.q.q.q.q.W
W      WddddddddddddddW..q.q.q.q.W
W      dddddddPdddddd..rq.qqq.q.W
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW

```

L'objectif du projet sera :

- Construire un programme java permettant de jouer une partie de boulder dash.
- Adapter ce programme pour développer des stratégies d'intelligence artificielle pour boulder dash.

Élément	Comportement	Char correspondant
Rockford	<ul style="list-style-type: none"> <li>— Peut se déplacer dans les 4 directions</li> <li>— Peut creuser dans la poussière et pousser les rocs</li> <li>— Peut sortir par une sortie ouverte pour finir un niveau</li> <li>— Peut être tué par une chute d'objet, pris dans une explosion, ou par le contact avec un ennemi</li> </ul>	R
Espace vide	<ul style="list-style-type: none"> <li>— Peut être creusé par Rockford, un ennemi, un objet ou un obstacle</li> </ul>	
Poussière	<ul style="list-style-type: none"> <li>— Peut être creusé par Rockford pour se déplacer</li> <li>— Peut être détruit par des explosions ou consommé par les amibes</li> </ul>	.
Roc	<ul style="list-style-type: none"> <li>— Peut tomber et rouler sur des murs, rocs ou diamants</li> <li>— Peut être poussé par Rockford</li> <li>— Peut être transformé en diamant s'il passe à travers un mur magique</li> <li>— Peut être détruit par une explosion</li> <li>— Peut être collecté par Rockford</li> </ul>	r
Diamant	<ul style="list-style-type: none"> <li>— Peut tomber ou rouler sur des murs, rocs ou diamants</li> <li>— Peut se transformer en roc s'il passe à travers un mur magique</li> <li>— Peut être détruit par une explosion</li> </ul>	d
Mur	<ul style="list-style-type: none"> <li>— Peut être détruit par une explosion</li> </ul>	w
Mur de titane	<ul style="list-style-type: none"> <li>— Ne peut être détruit</li> </ul>	W
Mur Magique	<ul style="list-style-type: none"> <li>— Ressemble à un mur (attention à l'affichage pour le joueur dans le mode interactif)</li> <li>— Peut être activé en faisant tomber un roc ou diamant dessus</li> <li>— Converti les rocs en diamants et inversement</li> <li>— Peut être détruit par les explosions</li> </ul>	M
Luciole	<ul style="list-style-type: none"> <li>— Se déplace vers la gauche et dans le sens des aiguilles du montre (si elle rencontre un mur)</li> <li>— Peut être tué par une chute de roc ou de diamant, ou au contact avec une amibe</li> </ul>	F, Q, o, O ou q
Libellule	<ul style="list-style-type: none"> <li>— Se déplace vers la droite et dans le sens inverse aux aiguilles d'une montre (si elle rencontre un mur)</li> <li>— Peut être tué par une chute de roc ou de diamant, ou au contact avec une amibe</li> <li>— Donne 9 diamants si elle est tuée (sauf par une explosion)</li> </ul>	B, b, c ou C
Amibe	<ul style="list-style-type: none"> <li>— Grandit d'une case aléatoirement tous les N tours (défini dans le fichier BDCFF)</li> <li>— Grandit uniquement sur les cases poussière ou espace vide</li> <li>— Peut tuer les lucioles ou libellules d'un simple contact</li> <li>— Se transforme en rocs si sa taille dépasse les 200 cases</li> <li>— Se transforme en diamants si elle ne peut plus grossir</li> <li>— Peut être détruit par les explosions</li> </ul>	a
Point de départ	<ul style="list-style-type: none"> <li>— Point de départ de Rockford dans le niveau</li> </ul>	P
Sortie	<ul style="list-style-type: none"> <li>— Ressemble à (et se comporte comme) un mur de titane</li> <li>— Est révélée/ouverte lorsque Rockford obtient un certain nombre de diamants</li> <li>— Permet de terminer le niveau si Rockford y rentre</li> </ul>	X

TABLE 1 – Éléments d'un niveau de boulder dash

De façon relativement abstraite, on peut voir l'espace des configurations du jeu de boulder dash comme un graphe où chaque sommet représente une configuration et où les successeurs d'un sommet sont les configurations obtenues grâce à un coup quelconque. Une partie consiste en un chemin à travers ce graphe entre un sommet de départ et un sommet d'arrivée.

## 2 Le sujet

Réaliser un programme java calculant des stratégies pour jouer à boulder dash. Votre programme permettra de :

- lire et interpréter un fichier au format BDCFF
- jouer en mode interactif sur un terminal
- concevoir et comparer des stratégies par une analyse du graphe des configurations

## 3 Fichier BDCFF

Boulder Dash est un jeu qui possède (encore aujourd'hui) une grande communauté de joueurs et de développeurs. Ceux-ci continuent de développer des adaptations du jeu ainsi que de nouveaux niveaux. Ils ont créé le format BDCFF pour représenter leurs niveaux. Vous trouverez sur celene un exemple de fichier dans ce format fournissant les informations nécessaires pour représenter 20 niveaux de boulder dash. Votre programme devra être en mesure de lire ce fichier de permettre de jouer dans les 20 niveaux qu'il définit. Ce fichier n'utilise qu'une petite partie des options du format BDCFF, il ne vous est pas demandé de prendre en compte tout le format dans ce projet, mais ce n'est pas interdit non plus ! Quelques informations concernant ce fichier : Le fichier global est décrit entre les balises [ game ] et [ / game ]. Chaque niveau est défini entre les balises [ cave ] et [ / cave ]. Les propriétés des niveaux sont fournies chacune sur une ligne.

Quelques exemples de propriétés de niveau :

Name=Cave 1	Nom du niveau
CaveDelay=6	Inutile dans votre version du jeu
CaveTime=150	Nombre de déplacements maximum avant de perdre (si plusieurs nombres, cela correspond à différents niveaux de difficulté)
DiamondsRequired=3	Nombre de diamants nécessaires pour faire apparaître la sortie
DiamondValue=30 15	Nombre de points par diamants, s'il y a deux nombres, le second est le nombre de points bonus donnés par chaque diamant
AmoebaTime=135	Nombre de tours avant que les amibes ne grandissent
MagicWallTime=6	Nombre d'utilisations max d'un mur magique
Colors=Black Black Red Blue Green	inutile dans votre version du jeu

Pour plus de précisions (en anglais) sur ce format de fichier : voir [http://www.boulder-dash.nl/bdcff\\_doc.html](http://www.boulder-dash.nl/bdcff_doc.html) Si dans un niveau (dans un fichier au format BDCFF), un élément apparaît qui n'est pas défini dans la table 1, vous pouvez le remplacer par un diamant.

## 4 Stratégies

Dans boulder dash, l'objectif est de récupérer un maximum de diamants avant de rejoindre la sortie. L'application d'une stratégie à un niveau donné devra retourner une suite d'actions (de déplacements) ainsi que le score obtenu si l'on réalise cette suite d'actions. Une bonne stratégie devra fournir le meilleur score possible dans le minimum d'actions. L'évaluation des stratégies pourra aussi s'appuyer sur le temps mis par le programme pour définir la meilleure suite d'action avec cette stratégie.

Nous définissons 5 stratégies :

1. Joueur Simplet : Le programme choisit au hasard l'ensemble des déplacements de Rockford.
2. Joueur Évolué : Le programme définit un certain nombre de stratégies aléatoires et les fait évoluer pour en améliorer le score (algorithme génétique - voir cours).
3. Joueur Directif : Le programme définit les plus courts chemins vers des cibles choisies aléatoirement (ou non) (voir cours).
4. Joueur Directif évolué : combinaison des deux stratégies ci-dessus.
5. Joueur Parfait : Le programme évalue l'ensemble des configurations du graphe pour obtenir le meilleur chemin (voir cours).

Le tableau suivant donne la valeur de l'option `strategie` utilisée dans la définition des commandes demandées :

Simplet	-simplet
Évolué	-evolue N N définit le nombre d'étapes d'évolutions/mutations autorisées par la stratégie (voir cours)
Directif	-directif
Directif évolué	-direvol
Parfait	-parfait

## 5 Que doit faire votre programme ?

Le programme java sera composé d'une unique archive jar **boulder\_dash.jar**. Ce programme devra s'exécuter dans un terminal avec les options suivantes :

- `java -jar boulder_dash.jar -name` affiche vos noms et prénoms
- `java -jar boulder_dash.jar -h` rappelle la liste des options du programme
- `java -jar boulder_dash.jar -lis fichier.bdcff` lis et affiche les paramètres d'un fichier BDCFF (de description de niveaux boulder dash).
- `java -jar boulder_dash.jar -joue fichier.bdcff [-niveau N]` offre la possibilité de jouer sur la console de manière interactive dans les niveaux décrits dans le fichier BDCFF. Le chemin utilisé par le joueur devra être sauvegardé dans un fichier au format .dash (voir exemple fourni). Un fichier BDCFF peut contenir plusieurs niveaux, si l'option `-niveau N` est passée au programme alors il ne jouera que le niveau N, sinon il devra permettre de jouer tous les niveaux décrits dans le fichier BDCFF, l'un après l'autre.
- `java -jar boulder_dash.jar -cal strategie fichier.bdcff -niveau N` calcule un chemin suivant une stratégie et renvoie ce chemin dans un fichier au format .dash (voir exemple fourni).
- `java -jar boulder_dash.jar -rejoue fichier.dash fichier.bdcff -niveau N` rejoue dans la console une partie de boulder dash en appliquant les déplacements fournis dans le fichier au format .dash fourni.
- `java -jar boulder_dash.jar -simul N strategie strategie fichier.bdcff -niveau N` évalue les deux stratégies en paramètre par simulation en lançant N parties et renvoie le score moyen, la longueur moyenne du chemin obtenu, et le temps moyen mis pour l'obtenir avec chacune des deux stratégies.

## 6 Quoi et quand rendre ?

A rendre le 15 mai

Quoi :

- format papier : rapport à remettre au secrétariat
- par mail :
  - rapport (format pdf)
  - le programme (fichier jar exécutable, contenant aussi les sources)
  - dossier compressé (au format tar.gz) contenant les données d'expérimentation.

## 7 Quelques références disponibles sur la toile et à la bibliothèque des sciences

1. [www.boulder-dash.nl](http://www.boulder-dash.nl), [http://strategywiki.org/wiki/Boulder\\_Dash](http://strategywiki.org/wiki/Boulder_Dash) Une mine d'informations sur boulder dash (avec de nombreux niveaux disponibles)
2. <http://www.boulder-dash.com/boulderdash.htm> Le site officiel de boulder dash
3. <http://www.codeincomplete.com/games/boulderdash/> un boulder dash en javascript dans votre navigateur, attention, il ne suit pas exactement les mêmes règles que celui que vous devez programmer
4. <http://www.emeraldmines.net/BDCFF/> et <http://www.elmerproductions.com/sp/peterb/BDCFF/> contiennent des informations sur le format BDCFF et des exemples de niveaux
5. <http://www.termSYS.demon.co.uk/vtansi.htm> contient des commandes de contrôle du terminal
6. <http://docs.oracle.com/javase/8/API/java>
7. <http://java.sun.com/docs/codeconv/index.html> donne les conventions d'écriture d'un programme java

8. La programmation en pratique, Brian W. Kernigan et Rob Pike. Editeur Vuibert informatique. Lire le premier chapitre sur la manière de commenter un programme et choisir des noms de variables
9. Au coeur de Java 2 (Volume 1), Notions Fondamentales, Cay S. Horstmann et Gary Cornell. Editeur CampusPress. Livre recommandé.
10. Algorithmique - 3ème édition - Cours avec 957 exercices et 158 problèmes, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Dunod.