

# Projet de Programmation

*Scrabble - Java*

Oral - Week 3

# Historique du jeu



## Alfred Mosher Butts

Architecte au chômage, il décide de créer un jeu alliant chance et réflexion.

## James Bruno

Après de nombreux lancements infructueux, un homme d'affaire rachète les droits et renomme le jeu "Scrabble".

## Jack Strauss

Président d'une chaîne de magasin, après avoir découvert le jeu, il décide de le commander en masse et lance sa popularisation.

## 150 millions d'exemplaires

Le scrabble est un phénomène mondial, un des jeux de société le plus vendu dans le monde.

⇒ Dès les années 1980, les premiers logiciels de jeu contre l'ordinateur ont vu le jour, suivis par l'explosion des versions en ligne comme Internet Scrabble Club (ISC) ou l'application mobile Scrabble Go.

⇒ De nos jours, de nombreux projets open source sont disponibles en ligne ainsi que de nombreuses applications pour tous les appareils

# Règles et explication

Les éléments à ne pas rater :

- Bonus
- Connexité
- Orthogonalité
- Priorité de bonus

Décompte des points :

- bonus de lettre
- somme des valeurs
- bonus de mots
- scrabble ?

Dictionnaire et répartition des lettres pour le français



Exemple pour "Article"

- le mot est valide
- pas de bonus de lettre
- somme = 9
- mot double = 18

# Complexité du sujet

**3 challenges pour permettre l'intégration de l'IA dans un jeu humain :**

1. La recherche exhaustive des coups légaux
2. Indexer le lexique pour trouver le meilleur apport de points en un temps infime
3. Simuler de vrais tours de jeux qui prendront en compte l'état du plateau après le coup

# Existant

*Le scrabble en 3 points culturels :*



- Littérature scientifique : algorithme de référence de Steven Gordon (GADDAG) ou le travail d'Appel et Guy (DAWG)

⇒ minimisent l'espace mémoire tout en maximisant la vitesse de parcours du dictionnaire

- Logiciels de référence : Quackle est le standard pour l'analyse stratégique

⇒ utilise des simulations de Monte-Carlo pour comparer les coups (espérance et gain des tours futurs)

- Plateformes : L'ISC (Internet Scrabble Club) sert de référence pour la pratique compétitive ou classique, garantissant une application stricte des règles internationales.

⇒ Scrabble GO = +10 Millions de téléchargements

# Algorithme et structures de données

Stratégie :

- Ne pas ouvrir le jeu ?
- Maximiser son score ?

Comment y répondre ?

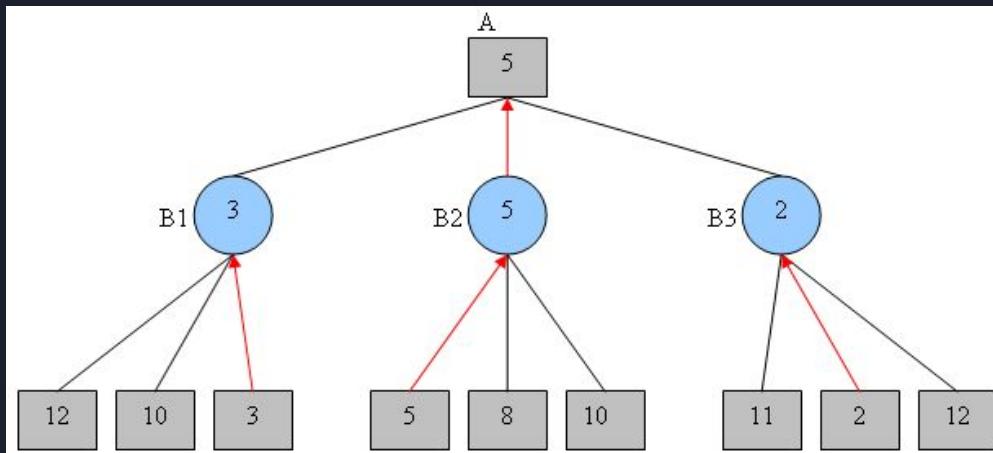
⇒ Deux algorithmes :

- MinMax
- Expectiminimax





# Algorithme MinMax

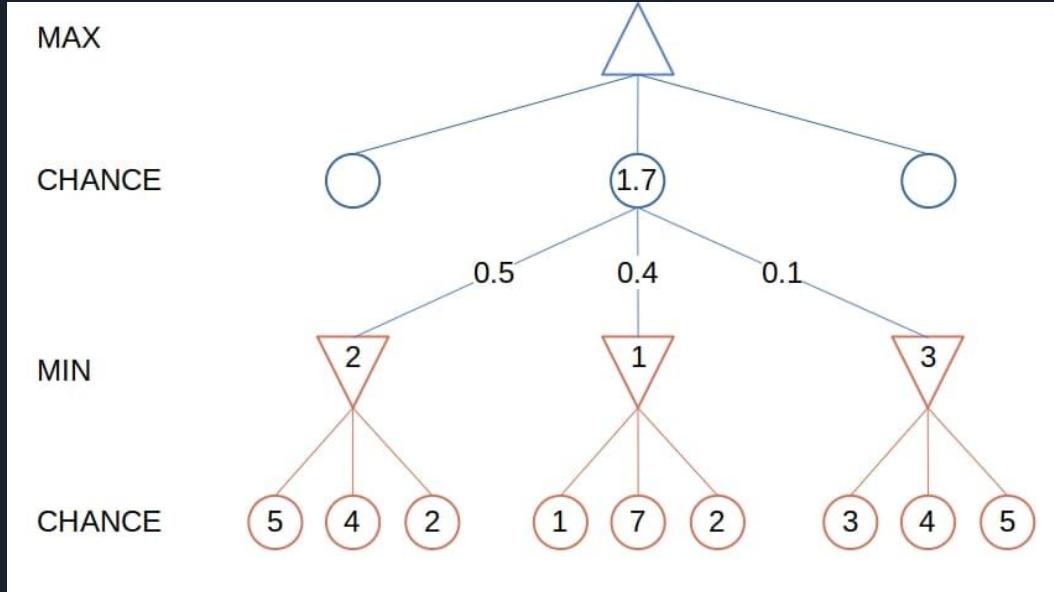


Noeud carré **MAX** : coup qui **maximise** mon score final

Noeud rond **MIN** : coup qui **minimise** mon gain

Problème : Jeu à info imparfaite, **IA surdéfensive** !

# Algorithme Exceptiminimax



**MAX** : Noeud triangle haut : coup qui **maximise** mon score final

**MIN** : Noeud triangle bas : coup qui **minimise** mon gain

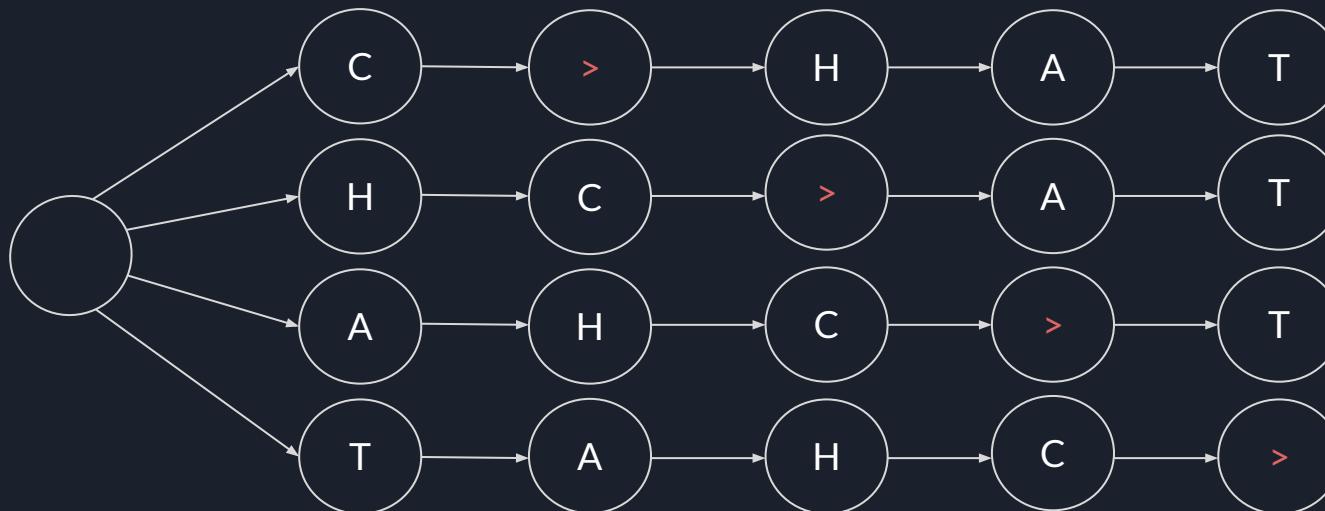
Nouveau type de noeud rond,  
**CHANCE** : modélise le **hasard** du tirage

IA plus réaliste qui peut prendre des risques

# Algorithme et structures de données

La structure de donnée GADDAG :

- Proposée en 1994 par Steven Gordon
- Spécialement créée pour les jeux comme le Scrabble
- Utilise les lettres d'ancrage déjà présentes sur le plateau pour créer un découpage logique



Exemple des représentations du mot chat dans l'arbre généré avec GADDAG

# Algorithme et structures de données

## Avantages :

- Très rapide
- Déterministe et exhaustive
- Impossible de sortir des mots qui ne sont pas dans le dictionnaire

	DAWG overall	DAWG per move	GADDAG overall	GADDAG per move	Ratio (D/G)
CPU time					
Expanded	9:32:44	1·344s	3:38:59	0·518s	2·60
Compressed	8:11:47	1·154s	3:26:51	0·489s	2·36
Page faults					
Expanded	6063		32,305		0·19
Compressed	1011		3120		0·32
Arcs Traversed	668,214,539	26,134	265,070,715	10,451	2·50
Per sec (compressed)	22,646		21,372		
Anchors used	3,222,746	126·04	1,946,163	76·73	1·64
Number of moves	25,569		25,363		
Average score	389·58		388·75		

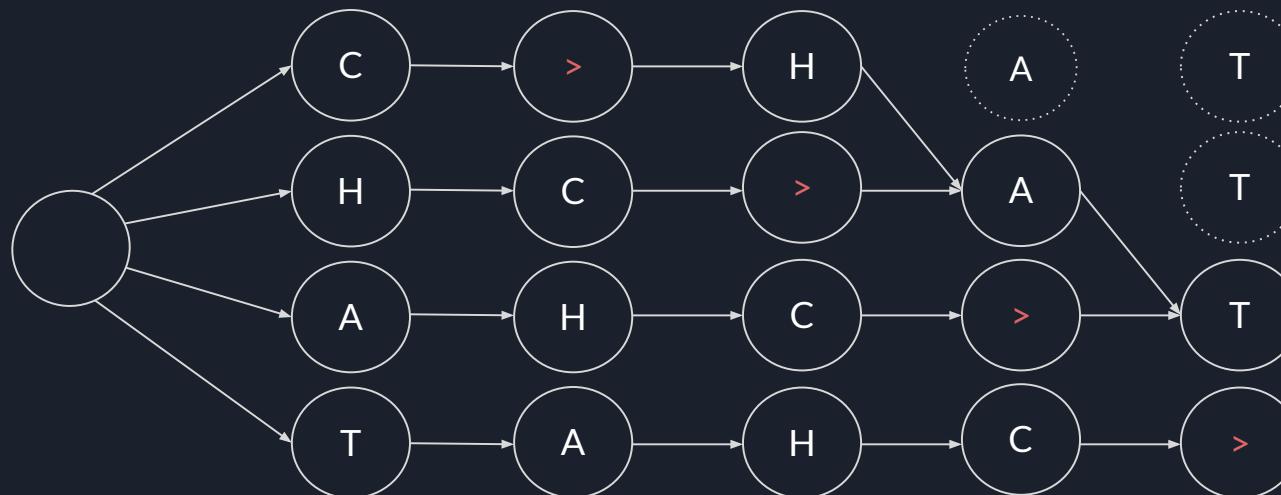
Comparaison de la performance relative entre DAWG et GADDAG faite par Gordon en 1994

# Algorithme et structures de données

Inconvénients :

- Très lourd
- Plus complexe à comprendre et à implémenter
- Crédit initiale longue

Moyen trouvé par Steven A. Gordon pour minimiser au maximum l'impact sur la mémoire :



# Algorithme et structures de données

Fonctionnement de l'algorithme :

Input :

- La structure GADDAG créée au préalable
- Le plateau de jeu
- Le chevalet du joueur

Output :

- Liste de tous les coups valides :
- Mot (en forme GADDAG)
- Position
- Direction
- Score
- Lettres du chevalet consommées

# Architecture du projet

## 1.1 : Découpage du projet au format MVC

### 1.Controller and IO

```
└── scrabble
    ├── App.java
    └── controller
        └── GameController.java
    └── io
        ├── ConfigLoader.java
        └── GameLoader.java
```

### 2.View

```
└── view
    ├── GameView.java
    ├── UserInterface.java
    ├── cli
    │   └── CommandHandler.java
    │       └── TerminalView.java
    └── gui
        └── JavaFxView.java
```

### 3.Model

```
└── model
    ├── ai
    │   ├── AIPlayer.java
    │   ├── MLAgent.java
    │   └── MinimaxSolver.java
    ├── core
    │   ├── Bag.java
    │   ├── Board.java
    │   ├── Game.java
    │   ├── HumanPlayer.java
    │   ├── Move.java
    │   ├── MoveHandler.java
    │   ├── Rack.java
    │   ├── Scoring.java
    │   ├── Square.java
    │   ├── Tile.java
    │   └── UndoRedo.java
    ├── dictionary
    │   ├── Dawg.java
    │   ├── Dictionary.java
    │   ├── Gaddag.java
    │   └── Lexicon.java
    ├── enums
    │   ├── Direction.java
    │   ├── MoveType.java
    │   └── SquareType.java
    ├── factories
    │   └── StandardBoardFactory.java
    ├── interfaces
    │   └── Player.java
    ├── network
    │   ├── GameClient.java
    │   ├── GameServer.java
    │   └── NetworkManager.java
    └── utils
        └── Point.java
```

# Architecture du projet

