

# LibreDragon

Implémentation d' une GUI LaTeX

23 Mai 2016

PEIRONE Olivier

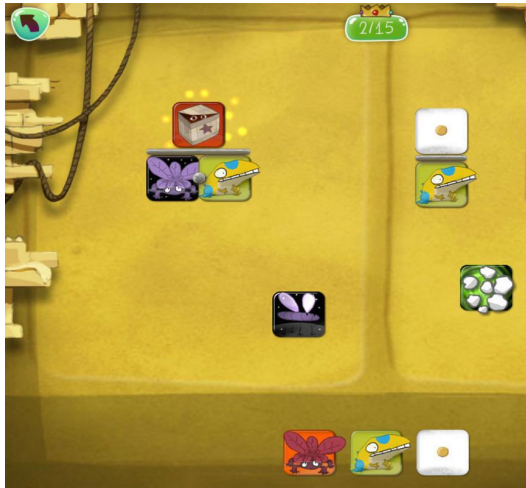
Université Aix-Marseille  
TER pour le M1 Informatique  
Encadrants : ARRIGHI Pablo, PERROT Kévin



# Le projet

## Présentation du sujet

**SECRETLY  
TEACHES  
ALGEBRA**



Fenêtre de jeu de DragonBox

Le projet

Présentation du sujet

L'existant

Architecture du

modèle

L'arbre

Les expressions

Les règles

Comment parser les  
opérateurs

Structure de graphics.cfg

Inclusion dans le  
programme

Implémenter une GUI  
: GraphicExpression-  
Factory

generateExpression

Fonctionnement global

Communication entre  
JAVA et MathJax : API  
REST

Démarrage d'un serveur

Intéraction avec localhost

Utilisation de MathJax

Identification d'une  
expression

Passage de MathJax à  
JAVA

Ce qu'il reste à faire

Démonstration

1

15

# Le projet

## L'existant

### Le projet

Présentation du sujet

L'existant

### Architecture du modèle

L'arbre

Les expressions

Les règles

### Comment parser les opérateurs

Structure de graphics.cfg

Inclusion dans le programme

### Implémenter une GUI : GraphicExpression-Factory

generateExpression

Fonctionnement global

### Communication entre JAVA et MathJax : API REST

Démarrage d'un serveur

Intéraction avec localhost

### Utilisation de MathJax

Identification d'une expression

Passage de MathJax à JAVA

### Ce qu'il reste à faire

Démonstration

2

15

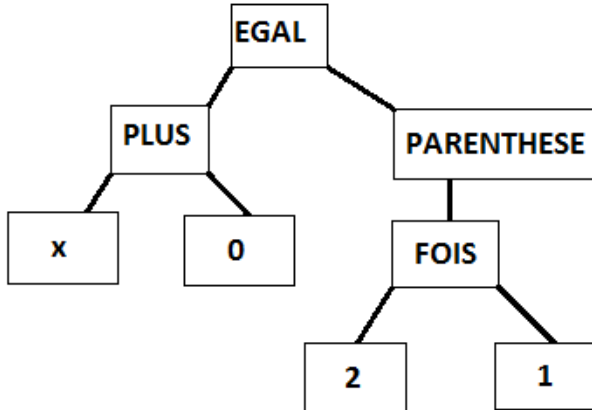


Fenêtre de jeu du projet de base

# Architecture du modèle

## L'arbre

3



Arbre correspondant à  $x + 0 = (2 * 1)$

# Architecture du modèle

## Les expressions

4

- ▶ Expression Binaire : Un seul opérateur, deux variables  
Exemple :  $2 + 3$ ,  $4 * 5$
- ▶ Expression Unaire : Une seule variable, un ou deux opérateurs  
Exemple :  $(a)$ ,  $-1$
- ▶ Expression Primaire : Une seule variable, aucun opérateur  
Exemple :  $a$ ,  $1$ ,  $\text{toto}$ ...

15

Le projet

Présentation du sujet

L'existant

Architecture du  
modèle

L'arbre

Les expressions

Les règles

Comment parser les  
opérateurs

Structure de graphics.cfg

Inclusion dans le  
programme

Implémenter une GUI  
: GraphicExpression-  
Factory

generateExpression

Fonctionnement global

Communication entre  
JAVA et MathJax : API  
REST

Démarrage d'un serveur

Intéraction avec localhost

Utilisation de MathJax

Identification d'une  
expression

Passage de MathJax à  
JAVA

Ce qu'il reste à faire

Démonstration

# Architecture du modèle

## Les règles

5

- Structure d'une règle :  
"Modèle d'entrée" => "Modèle de sortie"
- Exemple :  $A + A \Rightarrow 2 * A$

15

# Comment parser les opérateurs

Structure de graphics.cfg

## Le projet

Présentation du sujet

L'existant

## Architecture du modèle

L'arbre

Les expressions

Les règles

## Comment parser les opérateurs

Structure de graphics.cfg

Inclusion dans le programme

## Implémenter une GUI : GraphicExpression-Factory

generateExpression

Fonctionnement global

## Communication entre JAVA et MathJax : API REST

Démarrage d'un serveur

Intéraction avec localhost

## Utilisation de MathJax

Identification d'une expression

Passage de MathJax à JAVA

## Ce qu'il reste à faire

Démonstration

6

15

## Structure d'un opérateur défini dans graphics.cfg :

- ▶ TYPE ORIENTATION "OP. GAUCHE" "OP. DROIT"  
(A noter qu'on utilise pas l'orientation dans cette implémentation)
- ▶ Exemple d'une expression Unaire :  
PARENTHESIS h "(" ")"
- ▶ Exemple d'une expression Binaire :  
PLUS h "+" ""

# Comment parser les opérateurs

## Inclusion dans le programme

### Le projet

Présentation du sujet  
L'existant

### Architecture du modèle

L'arbre  
Les expressions  
Les règles

### Comment parser les opérateurs

Structure de graphics.cfg  
Inclusion dans le programme

### Implémenter une GUI : GraphicExpressionFactory

generateExpression  
Fonctionnement global

### Communication entre JAVA et MathJax : API REST

Démarrage d'un serveur  
Intéraction avec localhost

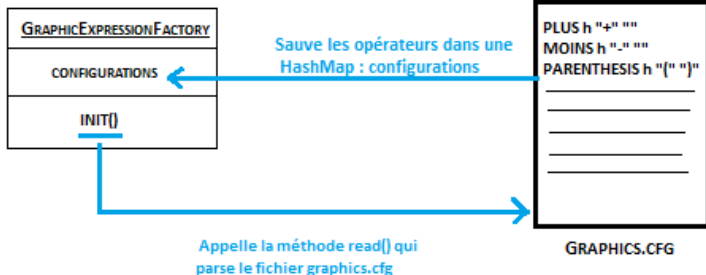
### Utilisation de MathJax

Identification d'une expression  
Passage de MathJax à JAVA

### Ce qu'il reste à faire

### Démonstration

7



15





# Implémenter une GUI : GraphicExpressionFactory

generateExpression

## Le projet

Présentation du sujet

L'existant

## Architecture du modèle

L'arbre

Les expressions

Les règles

## Comment parser les opérateurs

Structure de graphics.cfg

Inclusion dans le programme

## Implémenter une GUI : GraphicExpressionFactory

**generateExpression**

Fonctionnement global

## Communication entre JAVA et MathJax : API REST

Démarrage d'un serveur

Intéraction avec localhost

## Utilisation de MathJax

Identification d'une expression

Passage de MathJax à JAVA

## Ce qu'il reste à faire

## Démonstration

8

15

- ▶ generateBinaryExpression ->  
IDENTIFIANT filsGauche OPERATEUR filsDroit
- ▶ generateUnaryExpression ->  
IDENTIFIANT SYMBOLEGAUCHE fils SYMBOLE DROIT
- ▶ generatePrimaryExpression ->  
IDENTIFIANT littéral/chiffre

# Implémenter une GUI : GraphicExpressionFactory

## Fonctionnement global

### Le projet

Présentation du sujet  
L'existant

### Architecture du modèle

L'arbre  
Les expressions  
Les règles

### Comment parser les opérateurs

Structure de graphics.cfg  
Inclusion dans le programme

### Implémenter une GUI : GraphicExpressionFactory

generateExpression  
Fonctionnement global

### Communication entre JAVA et MathJax : API REST

Démarrage d'un serveur  
Intéraction avec localhost

### Utilisation de MathJax

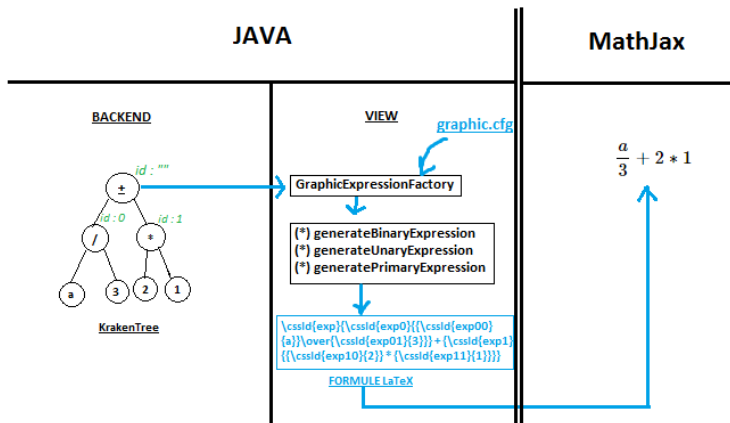
Identification d'une expression  
Passage de MathJax à JAVA

### Ce qu'il reste à faire

Démonstration

9

15



# Communication entre JAVA et MathJax :

## API REST

### Démarrage d'un serveur

#### Le projet

Présentation du sujet  
L'existant

#### Architecture du modèle

L'arbre  
Les expressions  
Les règles

#### Comment parser les opérateurs

Structure de graphics.cfg  
Inclusion dans le programme

#### Implémenter une GUI : GraphicExpression-Factory

generateExpression  
Fonctionnement global

#### Communication entre JAVA et MathJax : API REST

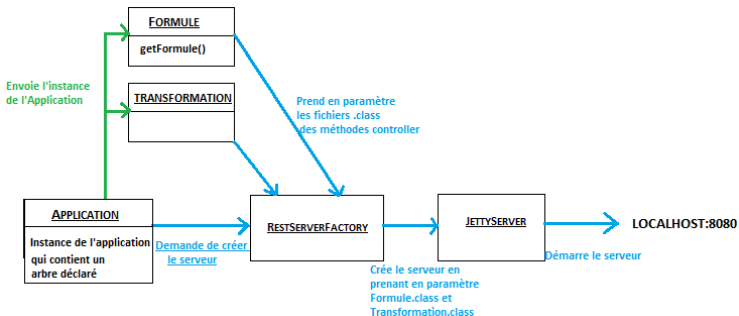
Démarrage d'un serveur  
Intéraction avec localhost

#### Utilisation de MathJax

Identification d'une expression  
Passage de MathJax à JAVA

#### Ce qu'il reste à faire

#### Démonstration



# Communication entre JAVA et MathJax :

## API REST

Intéraction avec localhost

### Le projet

Présentation du sujet  
L'existant

Architecture du modèle  
L'arbre  
Les expressions  
Les règles

### Comment parser les opérateurs

Structure de graphics.cfg  
Inclusion dans le programme

### Implémenter une GUI : GraphicExpression-Factory

generateExpression  
Fonctionnement global

### Communication entre JAVA et MathJax : API REST

Démarrage d'un serveur  
Intéraction avec localhost

### Utilisation de MathJax

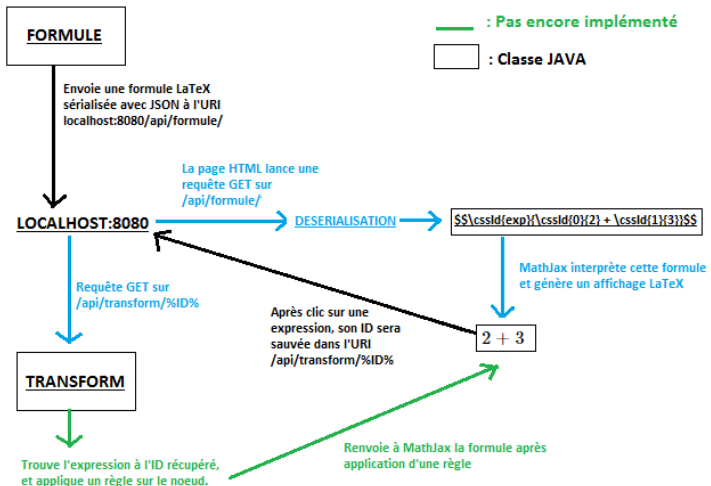
Identification d'une expression  
Passage de MathJax à JAVA

### Ce qu'il reste à faire

Démonstration

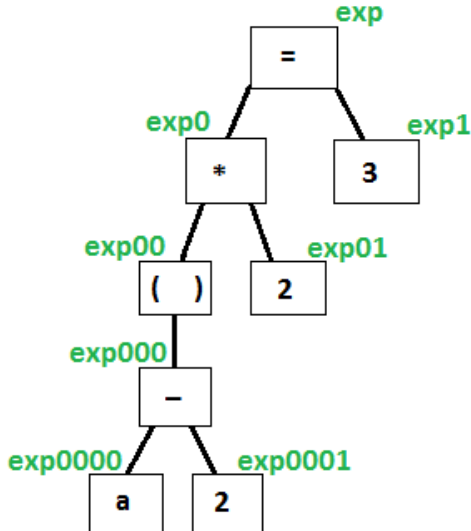
11

15



# Utilisation de MathJax

## Identification d'une expression



- Le projet
- Présentation du sujet
- L'existant
- Architecture du modèle
  - L'arbre
  - Les expressions
  - Les règles
- Comment parser les opérateurs
  - Structure de graphics.cfg
  - Inclusion dans le programme
- Implémenter une GUI : GraphicExpression-Factory
  - generateExpression
  - Fonctionnement global
- Communication entre JAVA et MathJax : API REST
  - Démarrage d'un serveur
  - Intéraction avec localhost
- Utilisation de MathJax
  - Identification d'une expression
  - Passage de MathJax à JAVA
- Ce qu'il reste à faire
- Démonstration

# Utilisation de MathJax

## Passage de MathJax à JAVA

### Le projet

- Présentation du sujet
- L'existant

### Architecture du modèle

- L'arbre
- Les expressions
- Les règles

### Comment parser les opérateurs

- Structure de graphics.cfg
- Inclusion dans le programme

### Implémenter une GUI : GraphicExpression-Factory

- generateExpression
- Fonctionnement global

### Communication entre JAVA et MathJax : API REST

- Démarrage d'un serveur
- Intéraction avec localhost

### Utilisation de MathJax

- Identification d'une expression
- Passage de MathJax à JAVA

### Ce qu'il reste à faire

### Démonstration

13

15

$$\left(\frac{a}{2}\right) * 2 = 3$$

APRES CLIC SUR L'EXPRESSION ENCADREE

Markers
Properties
Servers
Data Source Explorer
Snippets
Console

**KrakenTreeGradileTest [Gradle (STS) Build] Gradle Build on KrakenTreeGradileTest**

```

2016-05-22 13:14:21.854:INFO:oejs.ServerConnector:Thread-0: Started
2016-05-22 13:14:21.863:INFO:oejs.Server:Thread-0: Started @7826ms
MathJax a renvoie l'id :exp0001.
Avec celle-ci, on recupere l'expression : a.
Son equivalent LaTeX est : \\cssId{exp0001}{ {a} }.
                
```

# Ce qu'il reste à faire

## Le projet

Présentation du sujet

L'existant

## Architecture du modèle

L'arbre

Les expressions

Les règles

## Comment parser les opérateurs

Structure de graphics.cfg

Inclusion dans le programme

## Implémenter une GUI : GraphicExpression-Factory

generateExpression

Fonctionnement global

## Communication entre JAVA et MathJax : API REST

Démarrage d'un serveur

Intéraction avec localhost

## Utilisation de MathJax

Identification d'une expression

Passage de MathJax à JAVA

## Ce qu'il reste à faire

Démonstration

## Ce qu'il reste à faire :

- ▶ Gérer les évènements Drag-n-Drop
- ▶ Appliquer une règle à une expression renvoyée par MathJax

## Le projet

Présentation du sujet

L'existant

## Architecture du modèle

L'arbre

Les expressions

Les règles

## Comment parser les opérateurs

Structure de graphics.cfg

Inclusion dans le programme

## Implémenter une GUI : GraphicExpression-Factory

generateExpression

Fonctionnement global

## Communication entre JAVA et MathJax : API REST

Démarrage d'un serveur

Intéraction avec localhost

## Utilisation de MathJax

Identification d'une expression

Passage de MathJax à JAVA

## Ce qu'il reste à faire

## Ce que vous allez voir :

- ▶ Ce qu'il se passe quand on modifie le fichier de configuration des opérateurs
- ▶ Affichage LaTeX d'une expression
- ▶ Envoi de cette expression à MathJax
- ▶ Comment renvoyer cette expression à JAVA