

La plate-forme TurtleKit

Fabien MICHEL

LIRMM / SMILE

Master 2 - Moteurs de simulation

Plan

- 1 Inspirations
 - Le langage Logo
 - La révolution StarLogo
- 2 Contexte : la plate-forme MadKit
 - MadKit : vue générale
 - Les outils de simulation de la plate-forme MadKit
- 3 La plate-forme TurtleKit
 - TurtleKit : Vue générale
 - TurtleKit API
 - Exemples de simulations complexes
 - TurtleKit vs. autres

Plan

- 1 **Inspirations**
 - Le langage Logo
 - La révolution StarLogo
- 2 **Contexte : la plate-forme MadKit**
 - MadKit : vue générale
 - Les outils de simulation de la plate-forme MadKit
- 3 **La plate-forme TurtleKit**
 - TurtleKit : Vue générale
 - TurtleKit API
 - Exemples de simulations complexes
 - TurtleKit vs. autres

Le langage Logo : Origines

Qu'est-ce que le Logo ?

"Logo is the name for a philosophy of education and a continually evolving family of programming languages that aid in its realization." - Harold Abelson - Apple Logo, 1982

Au tout début

- milieu des années 60, Seymour Papert (mathématicien) ayant travaillé avec Jean Piaget (psychologue) à Genève co-fonde le MIT avec Marvin Minsky...
- Papert (et une équipe du MIT) crée la première version de Logo en 1967 ("Lisp sans parenthèse")

Le langage Logo : Origines

Qu'est-ce que le Logo ?

"Logo is the name for a philosophy of education and a continually evolving family of programming languages that aid in its realization." - Harold Abelson - Apple Logo, 1982

Au tout début

- milieu des années 60, Seymour Papert (mathématicien) ayant travaillé avec Jean Piaget (psychologue) à Genève co-fonde le MIT avec Marvin Minsky. . .
- Papert (et une équipe du MIT) crée la première version de Logo en 1967 ("Lisp sans parenthèse")

Le langage Logo : origines

Origines

- MIT fin des années 60
- Objectif : diriger un robot depuis un ordinateur avec des commandes simples (avance, recule, etc.)

Un robot. . .



. . . qui dessine



Le langage Logo : principes

caractéristiques d'une tortue logo

- propriétés : orientation, stylo (on/off)
- capacités : avancer, reculer, tourner

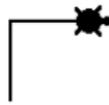
exemples de mots de base



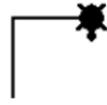
`forward 50`



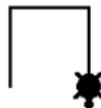
`right 90`



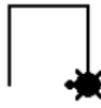
`forward 50`



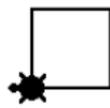
`right 90`



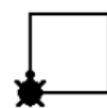
`forward 50`



`right 90`



`forward 50`

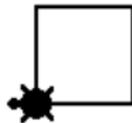


`right 90`

Le langage Logo : principes

création d'un nouveau mot

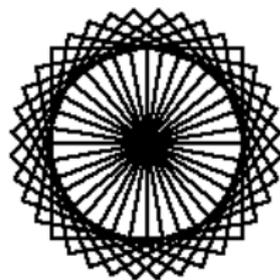
```
to square  
repeat 4 [forward 50 right 90]  
end
```



Le langage Logo : principes

et ainsi de suite...

```
to flower  
  repeat 36 [right  
    10 square]  
end
```

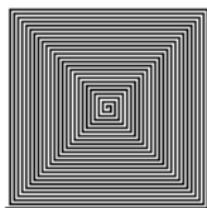


```
repeat 36 [right 10  
square]
```

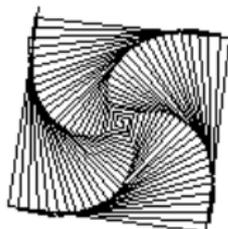
Le langage Logo : principes

vers des mots plus complexes

```
to spiral :size :angle  
if :size > 100 [stop]  
forward :size  
right :angle  
spiral :size + 2 :angle  
end
```



spiral 0 90



spiral 0 91

Plan

- 1 **Inspirations**
 - Le langage Logo
 - **La révolution StarLogo**
- 2 **Contexte : la plate-forme MadKit**
 - MadKit : vue générale
 - Les outils de simulation de la plate-forme MadKit
- 3 **La plate-forme TurtleKit**
 - TurtleKit : Vue générale
 - TurtleKit API
 - Exemples de simulations complexes
 - TurtleKit vs. autres

La révolution StarLogo

L'idée de Mitchel Resnick

- Début des années 90 : Multi-Logo “beyond the centralized mindset”
- un processus par tortue sur une architecture multi-processeur (nombreux problèmes)
- 1993/1994 : StarLogo (seulement mac au début, écrit en lisp)
- + Les *patches* : ajout d'une dynamique environnementale (diffusion/évaporation)

La révolution StarLogo

L'idée de Mitchel Resnick

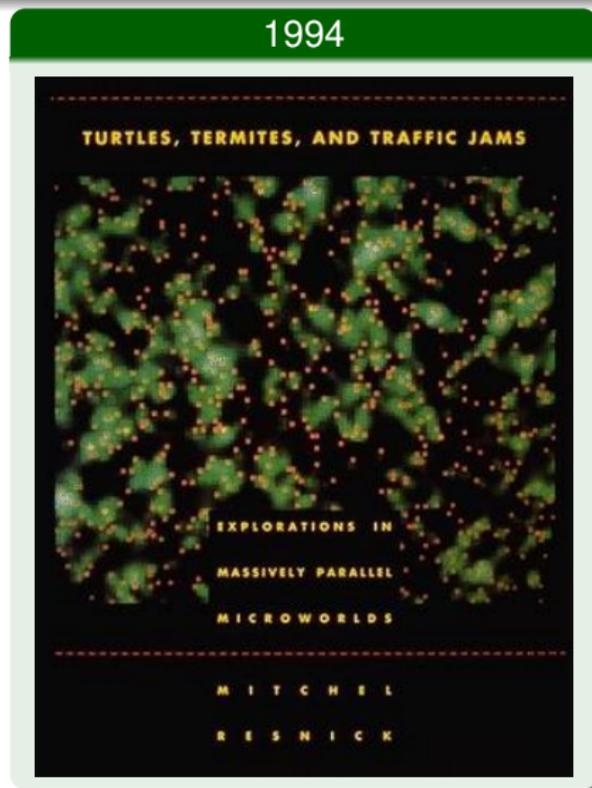
- Début des années 90 : Multi-Logo “beyond the centralized mindset”
- un processus par tortue sur une architecture multi-processeur (nombreux problèmes)
- 1993/1994 : StarLogo (seulement mac au début, écrit en lisp)
- + Les *patches* : ajout d'une dynamique environnementale (diffusion/évaporation)

La révolution StarLogo

L'idée de Mitchel Resnick

- Début des années 90 : Multi-Logo “beyond the centralized mindset”
- un processus par tortue sur une architecture multi-processeur (nombreux problèmes)
- 1993/1994 : StarLogo (seulement mac au début, écrit en lisp)
- + Les *patches* : ajout d'une dynamique environnementale (diffusion/évaporation)

La révolution StarLogo



La révolution StarLogo

aujourd'hui

- StarLogo version 2.21 (2006) (beaucoup de modèles)
- OpenStarLogo (??)
- StarLogo TNG (programmation graphique pour l'éducation)
- NetLogo (des tonnes de modèles)

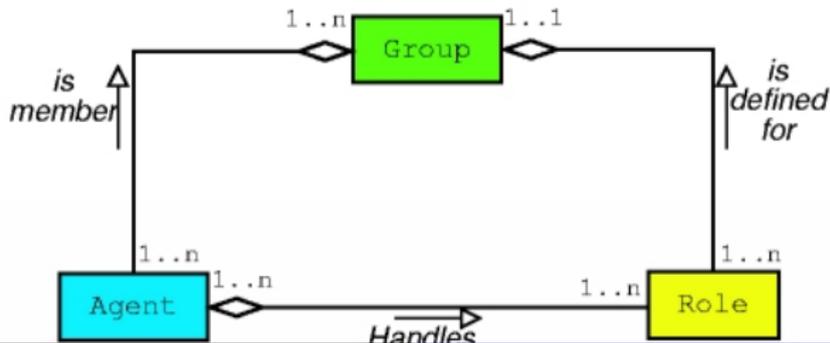
Plan

- 1 Inspirations
 - Le langage Logo
 - La révolution StarLogo
- 2 Contexte : la plate-forme MadKit
 - MadKit : vue générale
 - Les outils de simulation de la plate-forme MadKit
- 3 La plate-forme TurtleKit
 - TurtleKit : Vue générale
 - TurtleKit API
 - Exemples de simulations complexes
 - TurtleKit vs. autres

Caractéristiques

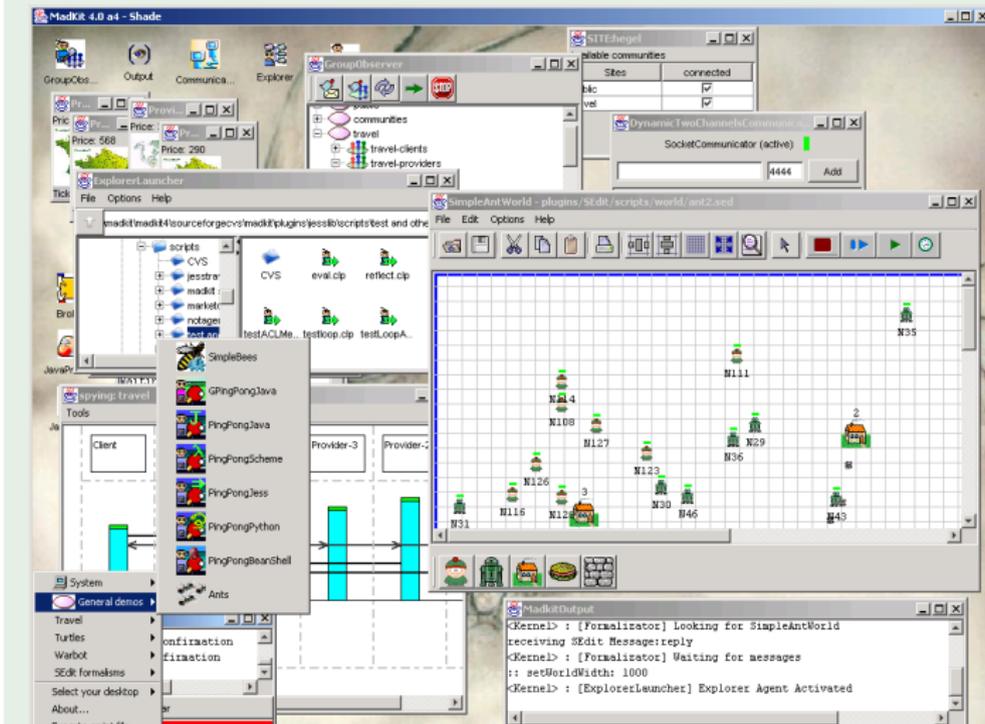
- Plate-forme pour le développement/déploiement de SMA
- Micro-noyau écrit en Java (distribuable P2P)
- Basée sur le modèle AGR :

Le modèle Agent/Groupe/Rôle



MadKit www.madkit.org

Le "bureau" MadKit



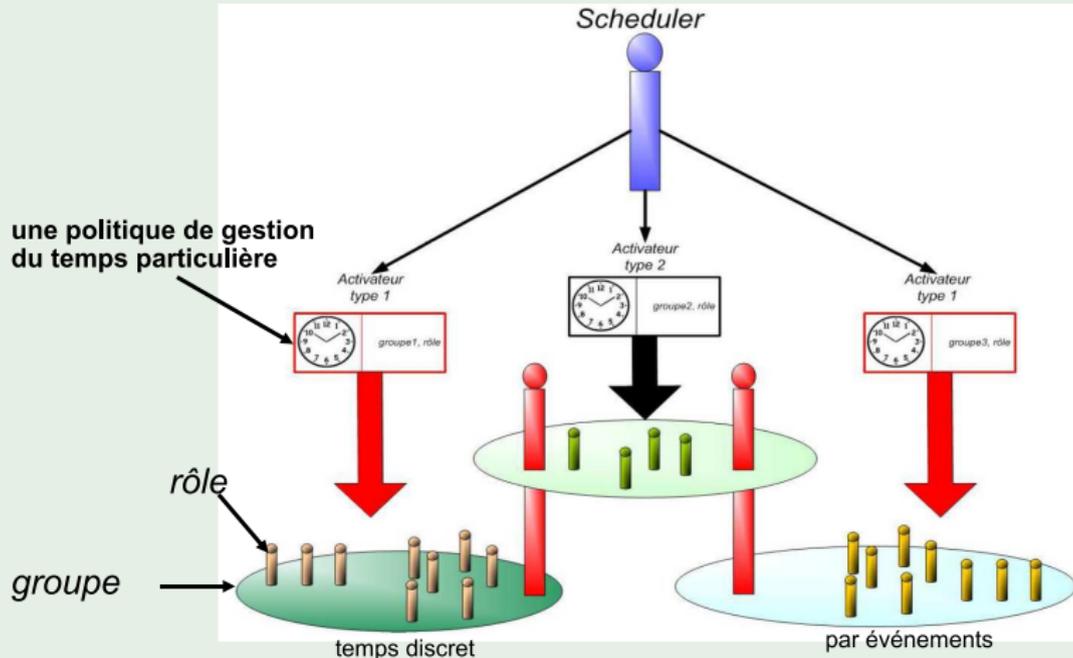
Plan

- 1 Inspirations
 - Le langage Logo
 - La révolution StarLogo
- 2 Contexte : la plate-forme MadKit
 - MadKit : vue générale
 - Les outils de simulation de la plate-forme MadKit
- 3 La plate-forme TurtleKit
 - TurtleKit : Vue générale
 - TurtleKit API
 - Exemples de simulations complexes
 - TurtleKit vs. autres

Les outils de simulation de la plate-forme MadKit

intégration dans MadKit d'un noyau de gestion du temps [Michel et al. 01]

Agent/Groupe/Rôle pour la simulation



Plan

- 1 Inspirations
 - Le langage Logo
 - La révolution StarLogo
- 2 Contexte : la plate-forme MadKit
 - MadKit : vue générale
 - Les outils de simulation de la plate-forme MadKit
- 3 La plate-forme TurtleKit
 - TurtleKit : Vue générale
 - TurtleKit API
 - Exemples de simulations complexes
 - TurtleKit vs. autres

TurtleKit

En quelques mots

- plate-forme écrite en java (au plus près des mots logo)
- réalisée avec MadKit
- Avantages :
 - Agent/Groupe/Rôle
 - messages
 - fonctionnalités MadKit
 - facile à étendre
 - et surtout, contrôle total sur la simulation

TurtleKit

En quelques mots

- plate-forme écrite en java (au plus près des mots logo)
- réalisée avec MadKit
- Avantages :
 - Agent/Groupe/Rôle
 - messages
 - fonctionnalités MadKit
 - facile à étendre
 - et surtout, contrôle total sur la simulation

TurtleKit

En quelques mots

- plate-forme écrite en java (au plus près des mots logo)
- réalisée avec MadKit
- Avantages :
 - Agent/Groupe/Rôle
 - messages
 - fonctionnalités MadKit
 - facile à étendre
 - et surtout, contrôle total sur la simulation

TurtleKit

En quelques mots

- plate-forme écrite en java (au plus près des mots logo)
- réalisée avec MadKit
- Avantages :
 - Agent/Groupe/Rôle
 - messages
 - fonctionnalités MadKit
 - facile à étendre
 - et surtout, contrôle total sur la simulation

TurtleKit

En quelques mots

- plate-forme écrite en java (au plus près des mots logo)
- réalisée avec MadKit
- Avantages :
 - Agent/Groupe/Rôle
 - messages
 - fonctionnalités MadKit
 - facile à étendre
 - et surtout, contrôle total sur la simulation

TurtleKit

En quelques mots

- plate-forme écrite en java (au plus près des mots logo)
- réalisée avec MadKit
- Avantages :
 - Agent/Groupe/Rôle
 - messages
 - fonctionnalités MadKit
 - facile à étendre
 - et surtout, contrôle total sur la simulation

Multiagent simulation platforms

Tur

- Dedicated to a particular area of reasearch
 - Cormas (Ecology), MissionLab (Robotic)...
- Generic: 2 different approaches:

NEWBIE

Difficulty of designing
complex models

Too simplistic features

Difficulty of model reuse

EXPERT

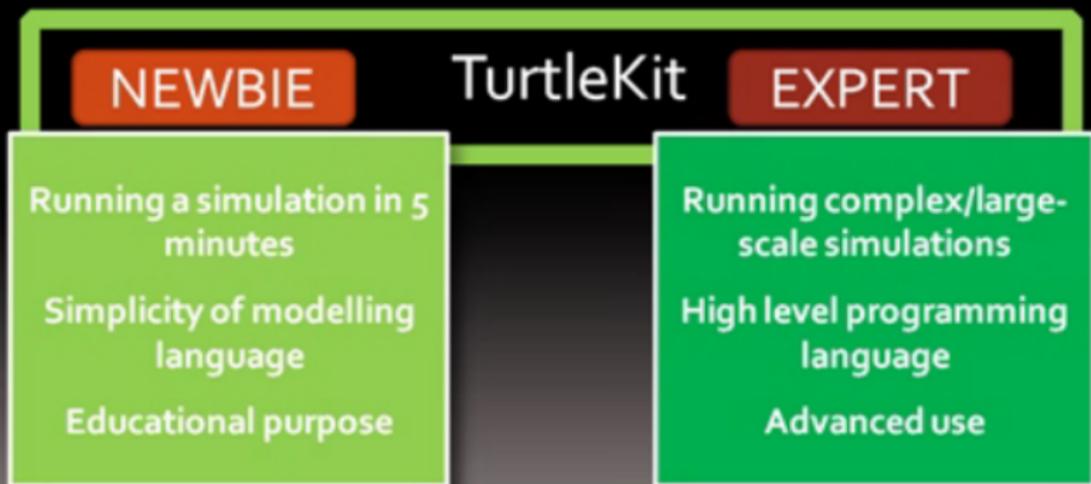
Programming skills
required

Difficulty of using
predefined models

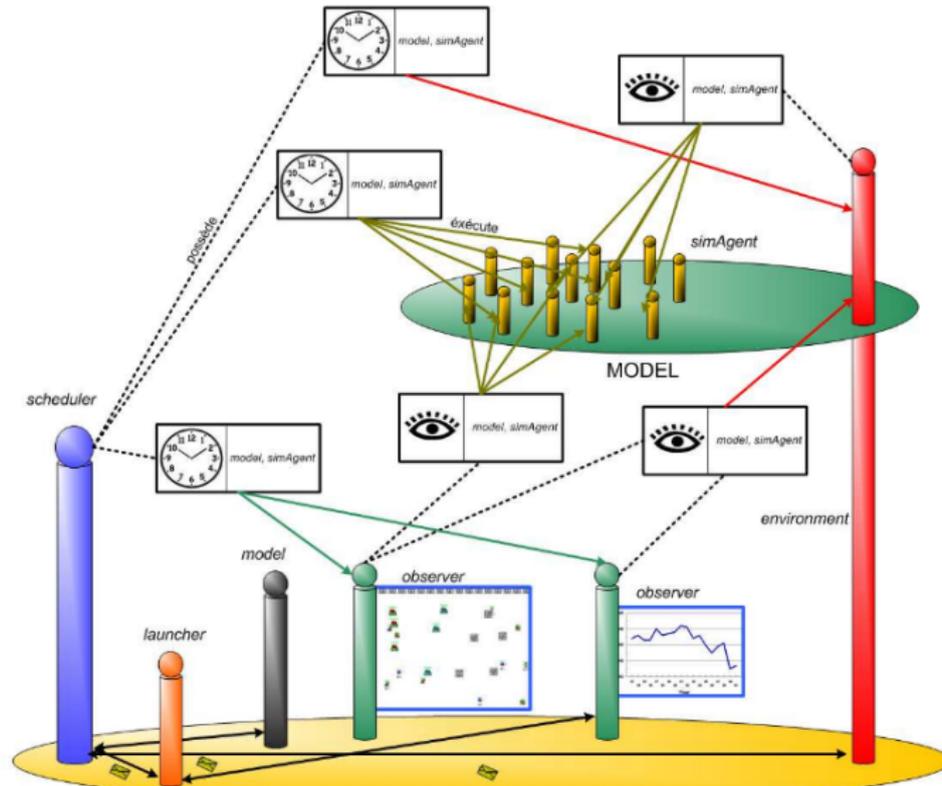
Multiagent simulation platforms

Tur

- Dedicated to a particular area of reasearch
 - Cormas (Ecology), MissionLab (Robotic)...
- Generic: 2 different approaches:



TurtleKit : moteur de simulation AGR



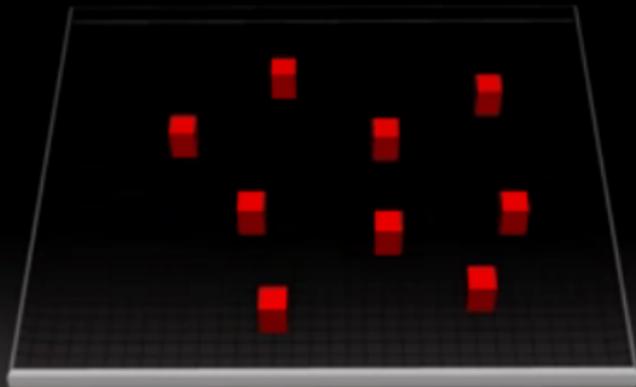
Overview



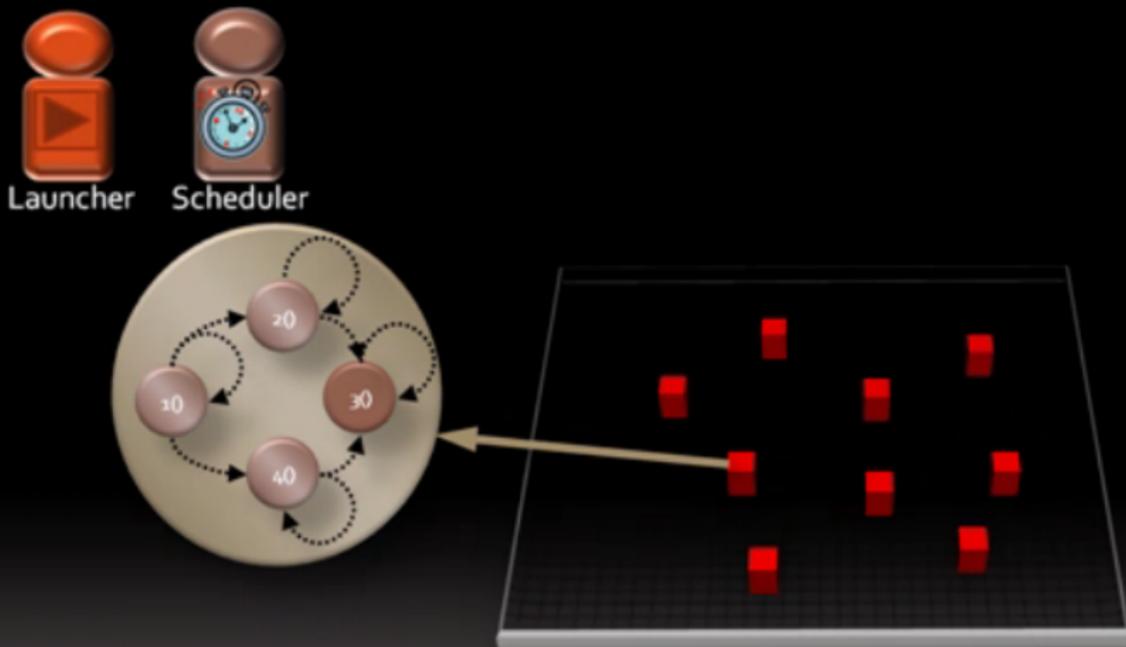
Launcher



Scheduler



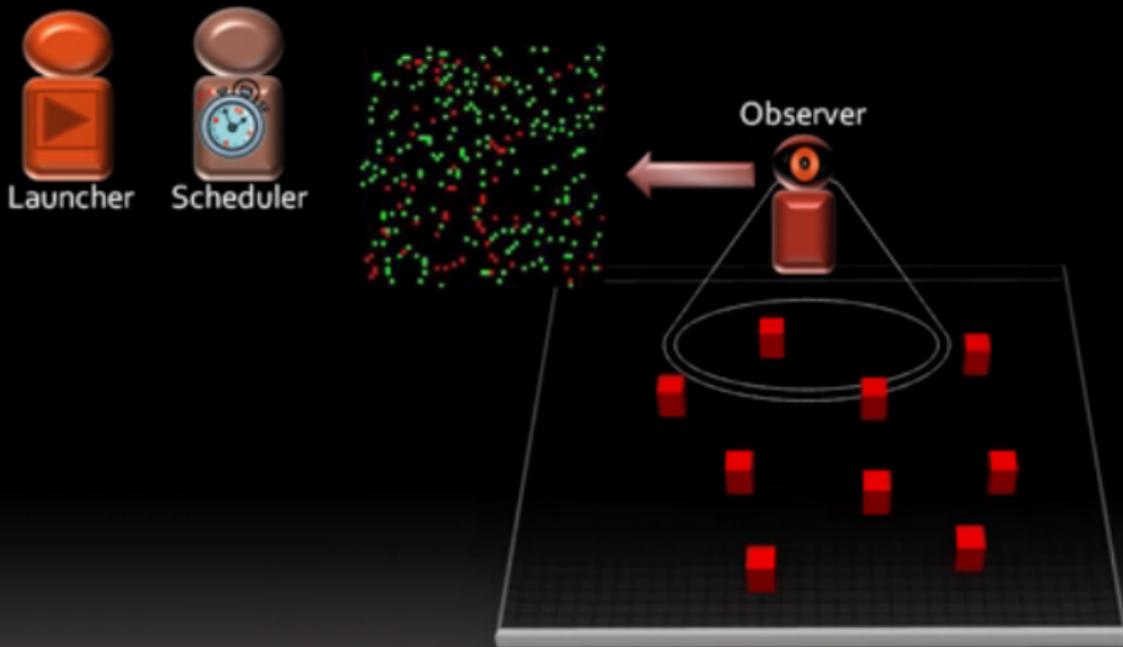
Overview



Tur

Tur

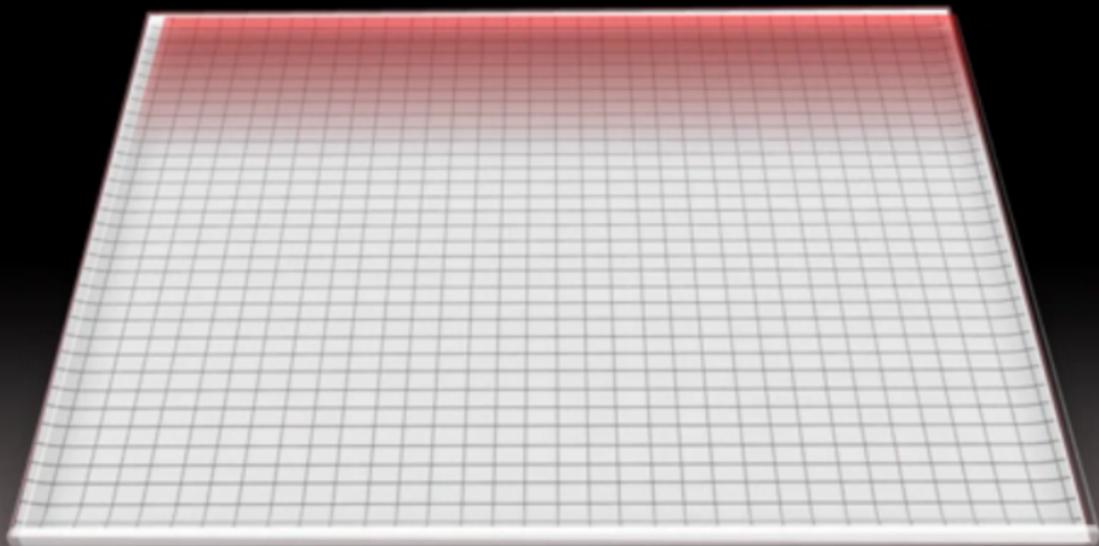
Overview



Environment

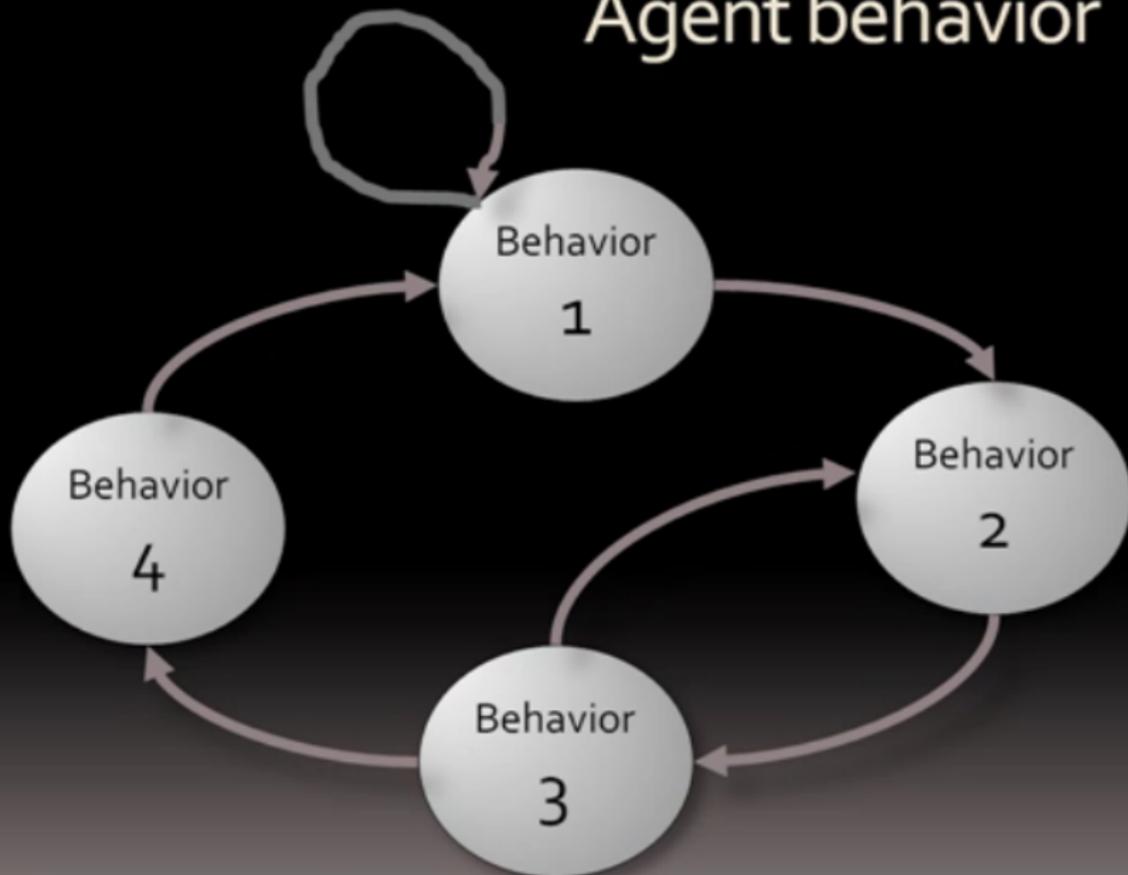
Tur 2D GRID

- MOVE
 - Towards(Agents), to(x,y)...
- GET/SET
 - Objects, Color, Pheromones...



Agent behavior

Tur

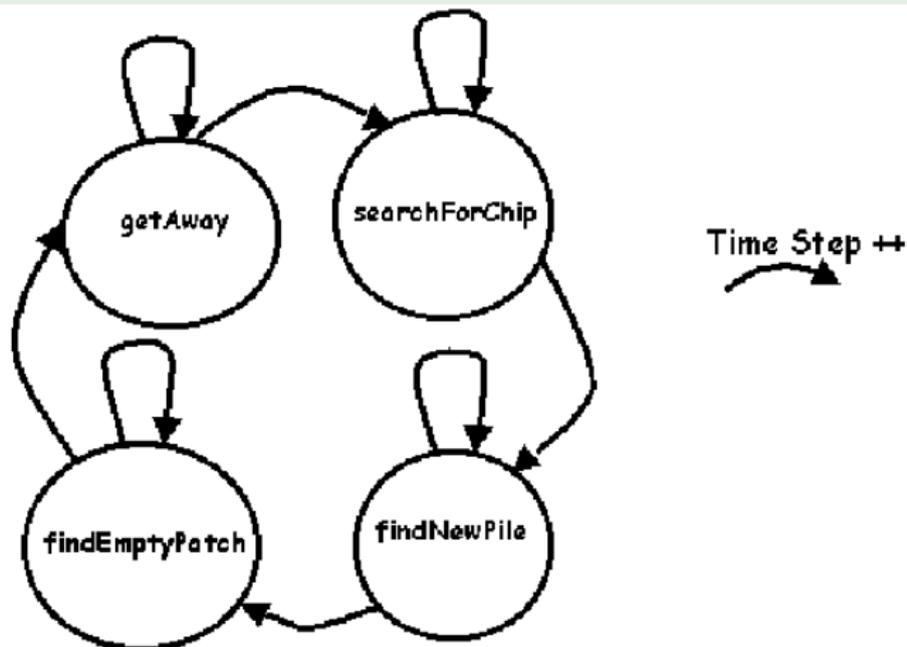


Plan

- 1 Inspirations
 - Le langage Logo
 - La révolution StarLogo
- 2 Contexte : la plate-forme MadKit
 - MadKit : vue générale
 - Les outils de simulation de la plate-forme MadKit
- 3 La plate-forme TurtleKit
 - TurtleKit : Vue générale
 - **TurtleKit API**
 - Exemples de simulations complexes
 - TurtleKit vs. autres

TurtleKit

Exemple de comportement : la termite électronique



TurtleKit : exemple de code

StarLogo vs. TurtleKit

```
to search-for-chip
  if pc = yellow ; if find a wood chip...
    [stamp black ; remove wood chip from
     setshape termite-wood-shape ; turn o
     jump 20
     stop] ; exit procedure
  wiggle
  search-for-chip
end
```

```
public String searchForChip()
{
  wiggle();

  if (getPatchColor() == Color.yellow)
  {
    setPatchColor(Color.black);
    randomHeading();
    fd(20);
    return("findNewPile");
  }
  else
    return ("searchForChip");
}
```

TurtleKit : API turtle

API

void	<u>bk</u> (int nb) turtle move backward
int	<u>countTurtlesAt</u> (int a, int b) return the number of turtles in the patch situated at (a,b) units away
int	<u>countTurtlesHere</u> ()
int	<u>createTurtle</u> (<u>Turtle</u> t) create a turtle at the creator position (xcor,ycor) returns the ID of the new turtle
double	<u>distance</u> (double a, double b) returns the distance from the patch (a,b).
void	<u>dropMark</u> (java.lang.String markName, java.lang.Object theMark) Drop a mark on the patch
void	<u>dropMarkAt</u> (java.lang.String markName, java.lang.Object theMark, int a, int b)
int	<u>dx</u> () return the x-increment if the turtle were to take one step forward

TurtleKit : API turtle

API

boolean	<code>getHidden</code> ()
java.lang.Object	<code>getMark</code> (java.lang.String variableName) get a mark deposited on the patch
java.lang.Object	<code>getMarkAt</code> (java.lang.String variableName, int a, int b)
java.awt.Color	<code>getPatchColor</code> ()
java.awt.Color	<code>getPatchColorAt</code> (int a, int b) get the color of the patch situated at (a,b) units away
double	<code>getPatchVariable</code> (java.lang.String variableName) return the value of the corresponding patch variable
double	<code>getPatchVariableAt</code> (java.lang.String variableName, int a, int b) return the value of the patch situated at (a,b) units away
<code>Turtle</code>	<code>getTurtleWithID</code> (int a) return the Turtle with the specified ID, null if not alive
int	<code>getWorldHeight</code> ()
int	<code>getWorldWidth</code> ()
void	<code>giveUpRole</code> (java.lang.String role) the turtle will no longer play the specified role

TurtleKit : API turtle

API

void	<u>giveUpRole</u> (java.lang.String role) the turtle will no longer play the specified role
void	<u>home</u> () teleport the turtle to the center patch
void	<u>incrementPatchVariable</u> (java.lang.String variableName, double value) set the value of the corresponding patch variable
void	<u>incrementPatchVariableAt</u> (java.lang.String variableName, double value, int a, int b)
boolean	<u>isMarkPresent</u> (java.lang.String markName) test if the corresponding mark is present on the patch (true or false)
boolean	<u>isMarkPresentAt</u> (java.lang.String markName, int a, int b) test if the corresponding mark is present on the patch situated at (a,b) units away
boolean	<u>isPlayingRole</u> (java.lang.String role)
void	<u>moveTo</u> (int a, int b) teleport the turtle to patch (a,b).
int	<u>mySelf</u> () return the turtle ID

TurtleKit : API turtle

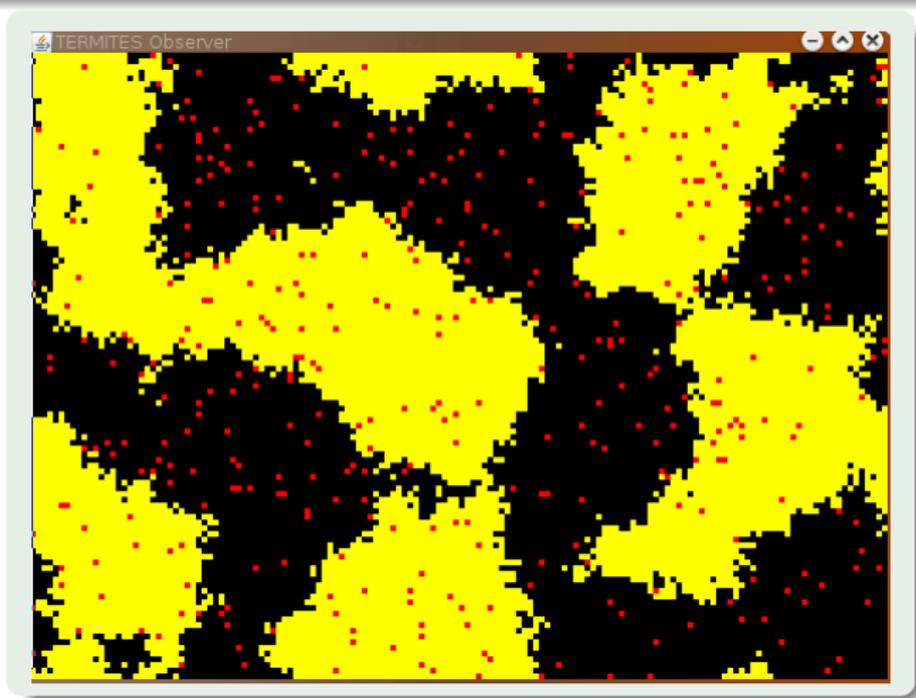
API

void	<u>playRole</u> (java.lang.String role) one way to identify a kind of turtle: give them a Role in the simulation.
void	<u>randomHeading</u> ()
void	<u>setColor</u> (java.awt.Color c)
void	<u>setHeading</u> (double direction) set the turtle heading to the value of direction
void	<u>setHidden</u> (boolean b) if true, the turtle hides itself (not draw)
void	<u>setPatchColor</u> (java.awt.Color c)
void	<u>setPatchColorAt</u> (java.awt.Color c, int a, int b) set the color of the patch situated at (a,b) units away
void	<u>setup</u> ()
void	<u>setX</u> (double a)
void	<u>setXY</u> (double a, double b)

TurtleKit : API turtle

API	
double	<u>towards</u> (double a, double b) returns direction to the patch (a,b).
void	<u>turnLeft</u> (double a)
void	<u>turnRight</u> (double a)
<u>Turtle</u> []	<u>turtlesAt</u> (int a, int b) return turtles who are on the patch situated at (a,b) units away
<u>Turtle</u> []	<u>turtlesHere</u> () return other turtles on the current patch
int	<u>xcor</u> ()
int	<u>ycor</u> ()

TurtleKit : Termites démo



TurtleKit : écrire ses propres observers

Étendre la classe observer (1. pour les patches)

```
public class FlavorViewer extends Viewer
{
    public void paintPatch(Graphics g, Patch p,int x,int y,int CellSize)
    {
        int a = ((int) p.getVariableValue("flavor"))%256;
        g.setColor(new Color(a,a,a));
        g.fillRect(x,y,CellSize,CellSize);
    }
}
```



TurtleKit : écrire ses propres observers

Étendre la classe observer (2. pour les turtles)

```
public void paintTurtle(Graphics g,Turtle t,int x,int y,int cellSize)
{
    g.setColor(t.getColor());
    g.fillOval(x,y,cellSize*3,cellSize*3);
}
public void paintPatch(Graphics g,Patch p,int x,int y,int cellSize)
{
    g.setColor(p.getColor());
    g.fillOval(x,y,cellSize*3,cellSize*3);
}
```

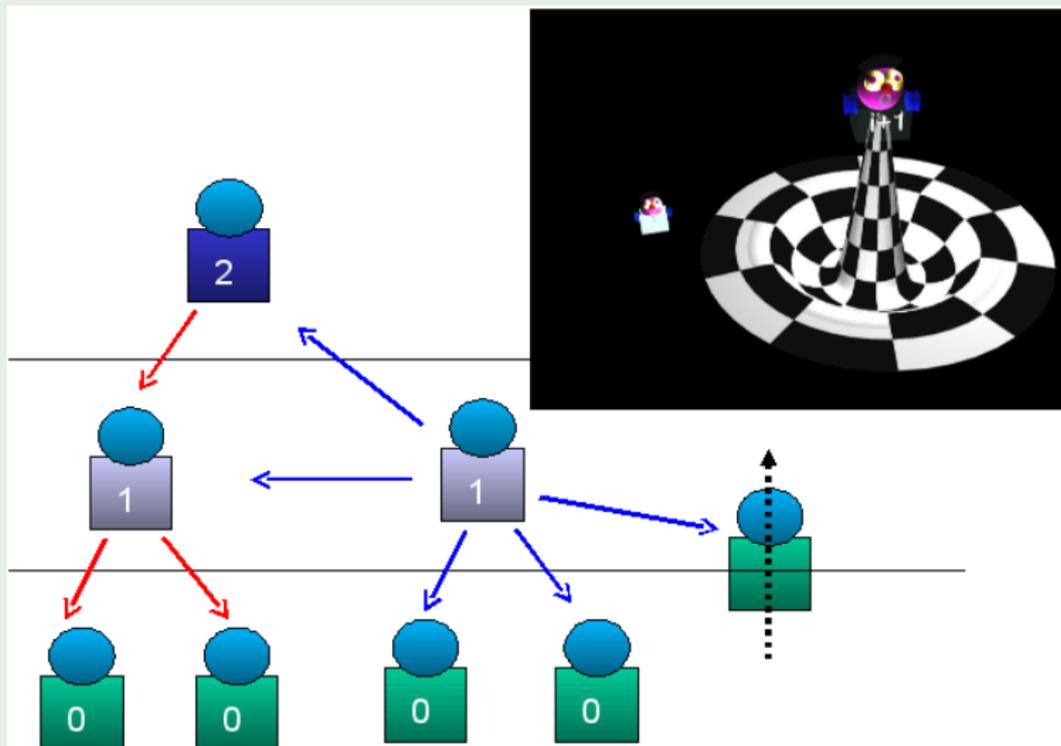


Plan

- 1 Inspirations
 - Le langage Logo
 - La révolution StarLogo
- 2 Contexte : la plate-forme MadKit
 - MadKit : vue générale
 - Les outils de simulation de la plate-forme MadKit
- 3 La plate-forme TurtleKit
 - TurtleKit : Vue générale
 - TurtleKit API
 - **Exemples de simulations complexes**
 - TurtleKit vs. autres

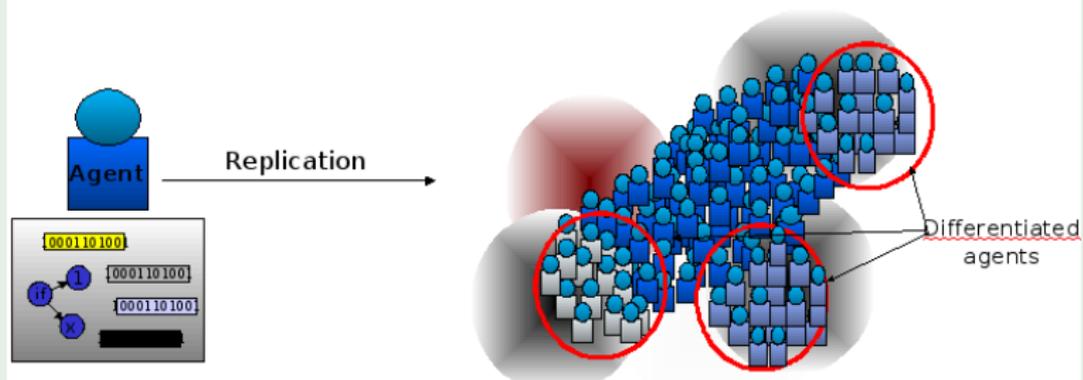
TurtleKit : exemple d'utilisation avancée

Un modèle pour l'émergence multi-niveaux [Beurier 05]



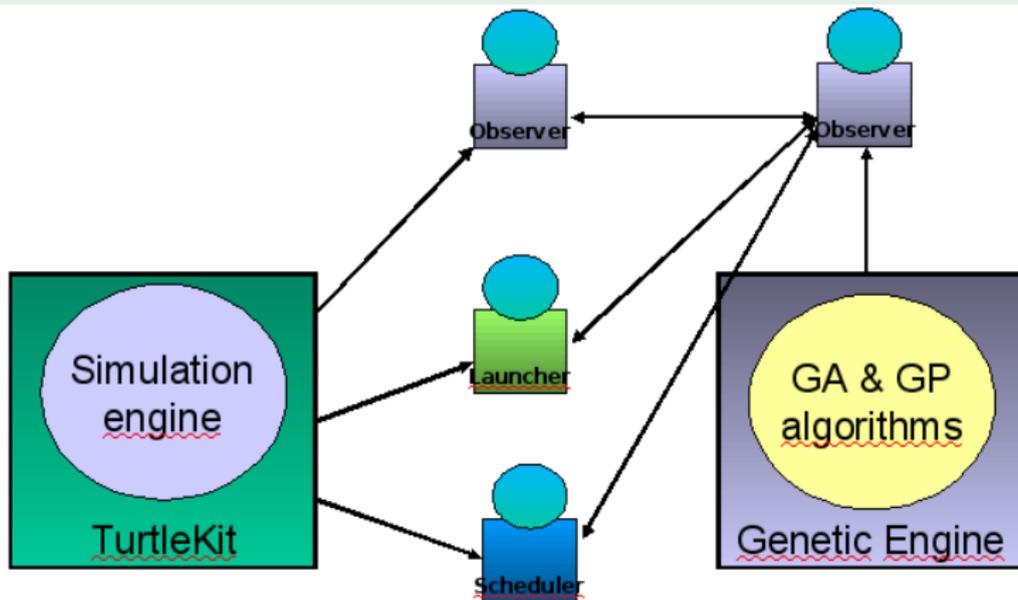
TurtleKit : exemple d'utilisation avancée

Un modèle inspirée de l'embryogénie [Beurier 06]



TurtleKit : exemple d'utilisation avancée

Un modèle inspirée de l'embryogénie



Plan

- 1 Inspirations
 - Le langage Logo
 - La révolution StarLogo
- 2 Contexte : la plate-forme MadKit
 - MadKit : vue générale
 - Les outils de simulation de la plate-forme MadKit
- 3 La plate-forme TurtleKit
 - TurtleKit : Vue générale
 - TurtleKit API
 - Exemples de simulations complexes
 - TurtleKit vs. autres

TurtleKit vs. autres

Avantages rappels

- MadKit
- Java (pas un seul fichier, héritage, etc.)
- plate-forme faite pour être étendue
- mais surtout. . .

TurtleKit vs. autres

Avantages rappels

- MadKit
- Java (pas un seul fichier, héritage, etc.)
- plate-forme faite pour être étendue
- mais surtout...

TurtleKit vs. autres

Avantages rappels

- MadKit
- Java (pas un seul fichier, héritage, etc.)
- plate-forme faite pour être étendue
- mais surtout...

TurtleKit vs. autres

Avantages rappels

- MadKit
- Java (pas un seul fichier, héritage, etc.)
- plate-forme faite pour être étendue
- mais surtout. . .

TurtleKit possède un vrai modèle temporel !!

