

# DevoCellPlayer

Visualisez vos cultures in-silico

Documentation utilisateur

## 1 - Introduction

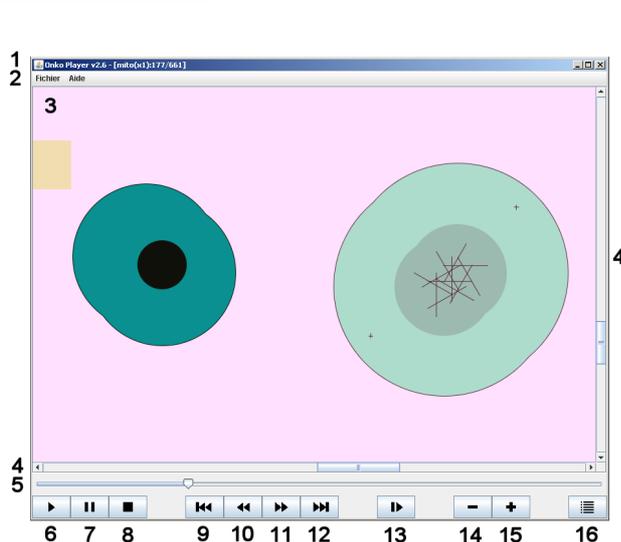
DevoCellPlayer est un logiciel de visualisation dédié à la simulation informatique de cultures cellulaires. Grâce à DCP, les présentations de vos résultats numériques seront plus explicites et faciliteront votre communication.

## 2 - Présentation

DevoCellPlayer reprend les fonctionnalités classiques de tout logiciel de visualisation de films : play, pause, stop, chapitres, etc...

Il y ajoute des fonctionnalités dédiées à la visualisation de simulations numériques : accélérer, ralentir, inverser le temps, tour par tour et zoom/dezoom.

### Vue d'ensemble :



- 1 : Barre de titre - [chapitre(vitesse):tour actuel/total tour]
- 2 : Barre des menus
- 3 : Surface de visualisation
- 4 : barres de défilement spatial (flèches clavier)
- 5 : barre de défilement temporel
  
- 6 : Play (touche Espace)
- 7 : Pause (touche Espace)
- 8 : Stop
  
- 9 : Chapitre précédent
- 10 : Accélérer en arrière
- 11 : Accélérer
- 12 : Chapitre suivant
  
- 13 : Tour par tour (touche Entrée)
  
- 14 : Dezoom (molette souris bas)
- 15 : Zoom (molette souris haut)
  
- 16 : Légende des couleurs choisies pour la simulation

Les barres de défilement spatiales et temporelle sont dotées de curseurs qui vous permettent de choisir l'espace-temps que vous souhaitez visualiser. Vous pouvez aussi faire défiler le cadre de visualisation simplement en cliquant avec le bouton gauche dans la fenêtre et en déplaçant le plan de visualisation par glissement. Avec le bouton droit, vous pouvez centrer la vue sur la zone désirée et vous pouvez zoomer/dézoomer avec la molette de la souris.

## 3 - Utilisation

DevoCellPlayer se branche sur votre simulation par l'intermédiaire d'une sortie texte formatée. DCP se charge de l'interprétation et de la mise en forme de l'affichage. Vous devez simplement programmer votre simulation pour créer cette sortie texte. Si vous travaillez en Java, vous bénéficierez d'outils qui vous faciliteront grandement la tâche. Si vous travaillez dans d'autres langages, vous pourrez développer les méthodes équivalentes en vous appuyant sur les algorithmes déjà développés en Java.

### 3.1 - Prérequis

DevoCellPlayer suppose que vous faites des simulations cellulaires au tour par tour. L'espace de simulation est de taille fixe. Le nombre de cellules est illimité, en théorie. En pratique, cela fera ralentir la visualisation et nécessitera plus de mémoire. La mémoire allouée par défaut permet de visualiser des simulations de grandes populations, cependant vous pourrez augmenter cette allocation si vos besoins l'exigent.

### 3.2 - Format des données

Pour être branchée à DevoCellPlayer, votre simulation doit produire une sortie texte composée de trois fichiers différents :

- description des états (\*.stt)
- description des actions (\*.act)
- description de la simulation (\*.dta)

Le fonctionnement de DevoCellPlayer est similaire à une animation scriptée en Flash. La simulation se déroule sur une ligne de temps répartie sur les lignes de temps individuelles des cellules.

Le fichier de description des états présente les états que vous désirez utiliser ainsi que les couleurs et formes pour les afficher. Les états sont des aspects statiques de l'affichage cellulaire. Indépendants des actions, ils peuvent vous servir à marquer certaines cellules d'une façon spécifique ou à indiquer une étape particulière de votre simulation. Les états peuvent être changés au cours d'une action sans modifier sa progression.

Le fichier de description des actions sert d'interprète entre les actions standard d'une simulation cellulaire (division, déplacement, gestion du substrat, etc...) et les actions que vous désirez utiliser. Une action se décrit par son nom, l'action standard à laquelle elle fait référence et différents paramètres de vitesse et de progression. Les actions influencent l'animation des cellules concernées mais pas leur état.

Le fichier de description de la simulation contient les informations nécessaires à une visualisation pertinente de vos résultats de calcul en appui sur les descriptions d'actions et d'états. Chaque cellule dispose de sa propre ligne de vie et interprète comme une actrice les actions et états dictés par son script. Les scripts des cellules-actrices peuvent être groupés sur le même tour de la ligne de simulation.

Pour des raisons de facilité d'écriture et de concision, la syntaxe adoptée pour rédiger les 3 fichiers d'un résultat de simulation est inspirée de la syntaxe JSON (cf <http://www.json.org/>)

Vous trouverez en annexe les diagrammes présentant les grammaires pour les trois documents que vous devez produire.

### 3.3 – Flux en temps réel

Vous pouvez utiliser DCP en parallèle avec le calcul de vos simulations. Pour cela, ils vous suffit d'initialiser les données avec les 3 documents standards et d'envoyer la sortie données de votre simulation dans le fichier `dcp.dta` situé dans le dossier d'installation de DCP. Cliquez ensuite dans « fichier → mode flux » pour activer la lecture en temps réel.

# Annexe A : syntaxe des documents à produire

Voici les spécifications grammaticales d'écriture des fichiers onkoVisu.

quelques indications :  
 cr signifie Carriage Return, retour à la ligne  
 R,G,B : Red, Green, Blue (Couleur)  
 le diamètre d'une cellule en stase normale est de 1

Pour faciliter l'utilisation je vous invite à noter les extensions comme suit :  
 Fichier états : maSimulation.stt  
 Fichier actions : maSimulation.act  
 Fichier données : maSimulation.dta

