



# DES IDEES PUISSANTES DANS LA CLASSE

UTILISATION DE SQUEAK POUR  
L'AMELIORATION DE L'APPRENTISSAGE  
DES MATHEMATIQUES ET DES SCIENCES

Trad. 1.2

B.J. Allen-Conn & Kim Rose



*Epilogue par Alan Kay*

Turing Award 2005 — Prix Nobel d'informatique 2005

Many of the designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances where the authors are aware of a claim, the product names appear in italics. Readers, however, should contact the appropriate companies for more complete information regarding trademarks, registration and product details.



**BE CAREFUL!** The activities and projects in this work were designed with safety and success in mind. But even the simplest activity or the most common materials could be harmful when mishandled or misused. Use common sense whenever you're exploring or experimenting.

Copyright © 2003 by B.J. Allen-Conn, Kim Rose and Viewpoints Research Institute, Inc.  
All rights reserved.

Cover art by Peter Maguire.

This book was prepared using the L<sup>A</sup>T<sub>E</sub>X document preparation system and reproduced by Typecraft Wood & Jones from final *PDF* files supplied by the authors.  
[www.typecraft.com](http://www.typecraft.com)

Printed in the United States of America.

Published by Viewpoints Research Institute, Inc., Glendale, California.  
[www.viewpointsresearch.org](http://www.viewpointsresearch.org)



The publication of this book was funded in part by the National Science Foundation under Grant No. 0228806. Opinions expressed are those of the authors and do not necessarily reflect those of the Foundation.

ISBN 0-9743131-0-6

11 10 9 8 7 6 5 4 3 2 1

<b>Préface</b>	<b>v</b>
<b>Conduite de voitures</b>	<b>1</b>
Projet 1 : Principes de base d'etoy et dessin d'une voiture . . . . .	1
Excursion : Sauvez votre travail . . . . .	5
Projet 2 : jouer avec la voiture . . . . .	7
Projet 3 : Conduite d'une voiture avec un volant . . . . .	13
Projet 4 : Créer une voiture intelligente suivant une route . . . . .	17
Projet 5 : Deux voitures avec des vitesses variables . . . . .	23
Projet 6 : Plus de piment avec des voitures de compétition . . . . .	31
Projet 7 : La course fantastique . . . . .	35
Excursion : Création d'animations . . . . .	39
Excursion : Mesures . . . . .	43
Projet 8 : A propos de la vitesse . . . . .	47
Projet 9 : Des voitures aux balles . . . . .	53
<b>Apprendre la gravité</b>	<b>57</b>
Excursion : gravité . . . . .	57
Projet 10 : Un autre regard sur la chute des balles . . . . .	61
Projet 11 : Mesurer la distance . . . . .	65
Excursion : Créer des films . . . . .	68
Projet 12 : Une simulation de chute de balles . . . . .	73
<b>Epilogue</b>	<b>77</b>
<b>Glossaire</b>	<b>85</b>
<b>Lectures futures</b>	<b>90</b>
<b>Remerciements</b>	<b>91</b>



«Qu'est-ce qu'une idée puissante?», «A quel domaine de la vie ou de l'apprentissage faites-vous allusion?» Nous nous sommes confrontés aux commentaires d'une version encore avancée de ce livre.

Avant de commencer une discussion sur les «idées puissantes» nous voulons exprimer notre profond et humble respect à Seymour Papert et Alan Kay. Papert — un enseignant renommé, mathématicien, informaticien, et créateur du langage de programmation Logo — fut le premier à faire référence aux «idées puissantes» dans son livre *Mindstorms : Children, Computers and Powerful Ideas*. Mais que voulait dire Seymour par «idées puissantes»? Il les a définies par des «outils intellectuels». Seymour croyait qu'explorer de telles idées par l'utilisation des ordinateurs, autant que des travaux manuels, pourrait offrir aux enfants un face-à-face avec leurs intuitions. Il pensait que les ordinateurs pourraient aider les enfants en leur permettant d'extérioriser leurs attentes imaginatives, et, en utilisant les machines comme simulateurs et modèles, ils pourraient repenser et remodeler leur connaissances intuitives.

Alan Kay — notre ami, tuteur, guide de la puissance derrière Squeak — a été profondément influencé par Seymour et ses idées. Ce sont celles-ci qui sont à l'origine de l'idée de «Dynabook» d'Alan voilà plus de 30 ans, et qui l'ont poussé à créer des outils dynamiques pour enfants destinés à amplifier leur apprentissage. Alan nous a guidé au cours de ces dernières années et ensemble nous avons développé cette suite de projets reposant sur les fondations de Seymour et d'Alan.

Dans notre travail avec les enfants nous avons rencontré plusieurs situations où leurs intuitions reposent sur des idées qui sont en conflit avec les réalités du monde — des choses simples comme le lâcher d'une balle depuis le toit de l'école, par exemple. En suivant leurs intuitions, les enfants pensent souvent qu'un poids de 6 kilogrammes touchera le sol avant une balle de mousse. Après avoir expérimenté un certain nombre d'activités sans ordinateur telles que des mesures de lâcher de balles, une partie des enfants se rendent compte que leurs intuitions étaient fausses, tandis que d'autres pensent au contraire qu'ils ont raison. Après plusieurs expérimentations et simulations que les enfants réalisent et créent eux-même avec l'aide d'ordinateurs, la réalité des faits rentre en conflit avec leurs intuitions — une puissante idée sur comment le monde fonctionne, monde qu'ils peuvent maintenant comprendre en profondeur.

Ce livre a été écrit pour partager plusieurs projets d'idées pouvant être mis en pratique à l'intérieur ou à l'extérieur d'une salle de classe. Il propose et étudie un certain nombre de ces idées puissantes à travers des projets que les enfants créent en Squeak et via des activités annexes que l'on appelle «excursions». Les excursions introduisent des idées puissantes pouvant être explorées avec Squeak ou bien elles illustrent des concepts clefs à travers des activités manuelles et sans ordinateur. Ce livre a pour but d'être un point de départ pour comprendre comment les projets Squeak peuvent être utilisés pour amplifier l'apprentissage.

Les projets de ce livre explorent les idées puissantes en sciences tel que le zéro, les nombres positifs et négatifs, les coordonnées  $x$  et  $y$ , les facteurs multiplicatifs, l'accélération et la gravité. Une variable peut également être considérée comme une idée puissante dans le domaine de l'informatique. Ces puissantes méthodes d'enseignement incluent le travail collaboratif, guident et modèlent la compréhension des étudiants à travers une large variété de médias.

La base pour explorer ces idées est une série de projets commençant par construire sa propre voiture. En effet, il est connu que les enfants veulent «conduire une voiture». Les projets se construisent les uns à partir des autres en introduisant de nouvelles notions. Il est supposé que les notions de mathématiques seront données préalablement ou avec les projets de ce livre. De plus, nous pensons qu'utiliser plusieurs types de médias (livres, Internet, vidéos, etc.) aident les étudiants à gagner en compréhension et à multiplier leurs points de vues.

## PREFACE

Squeak est bien plus qu'un traitement de textes — c'est *un traitement d'idées*. C'est un langage, un outil, et un média offrant un environnement. Si vous n'avez pas déjà Squeak, vous pouvez le télécharger depuis le site web de Squeakland (<http://www.squeakland.org>). Squeak est libre, gratuit, et marche sur les principales plateformes. Il est le résultat d'une communauté libre guidée par le *Viewpoints Research Institute, Inc.*, une organisation à but non-lucratif (<http://www.viewpointsresearch.org/about.html>).

Idéalement, avant d'utiliser ce livre, vous devriez installer Squeak sur votre ordinateur et suivre l'introduction en ligne qui peut être trouvée sur <http://www.squeakland.org/author/etoys.html>.

Squeak est un puissant système possédant différents points d'entrés pour différents types d'utilisateurs. Les projets de ce livre sont tous basés sur le niveau «Etoy» de Squeak. Ces projets peuvent être téléchargés depuis le site web [Squeakland.org](http://www.squeakland.org). «Etoys» («Electronique», «Education», «Exitant», «Exploration») sont des modèles, simulations et jeux construits en assemblant des briques dans des scripts destinés à envoyer des commandes à des objets dessinés. Par la suite, comme les utilisateurs gagnent en maîtrise à la construction de ces scripts, ils peuvent utiliser d'autres facettes de Squeak plus en rapport avec leurs nouvelles compétences. Les utilisateurs experts de Squeak (programmeurs professionnels et développeurs de médias) ne devraient pas utiliser Etoy pour leurs créations, mais un autre niveau offrant de meilleures possibilités en utilisant une autre interface.

Nous vous invitons à rejoindre la communauté en ligne de Squeakland en souscrivant à la liste de diffusion (<http://www.squeakland.org/join>). Nous vous encourageons à partager vos idées et exemples de projets. Les projets contenus dans ce livre sont basés sur quelques puissantes idées. Notre rêve consiste à créer un millier de tels projets et de les partager via l'Internet, CD-ROM, et livres. Nous espérons que vous y contribuerez en ajoutant de nouvelles connaissances et en nous aidant à développer Squeak.

Nous espérons que vous trouverez ces projets ludiques et qu'ils libéreront de puissantes idées en vous et vos enfants. Enfin, nous remercions profondément Alan et Seymour pour leurs idées, leurs passions, et pour les inspirations qu'ils nous ont données à explorer comment les ordinateurs peuvent aider les enfants à comprendre les idées puissantes.

BJ Allen-Conn,  
Kim Rose

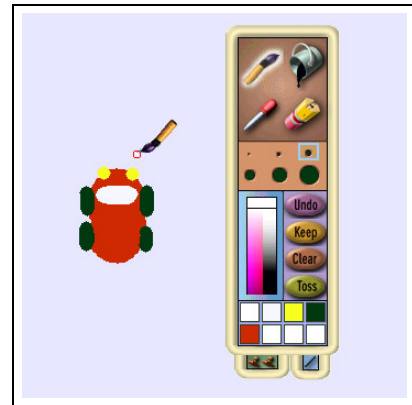
Los Angeles, California  
Juillet 2003

Ce livre a été traduit bénévolement par Stéphane Ducasse, Alexandre Bergel, François Schnell, Noury Bouraqaadi et Serge Stinckwich. Des parties de cette traduction ont été relues par Florence Ducasse, Clément Ducasse, Hilaire Fernandes, Isabelle Rondel, Rémi Sussan...

Vous avez entre les mains la version 1.2; de la traduction n'hésitez pas à nous contacter pour des améliorations.

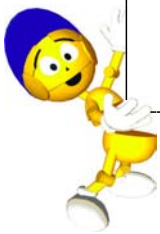
## PROJET 1

# PRINCIPES DE BASE D'ETOY ET DESSIN D'UNE VOITURE



La première partie de ce projet est présentée de manière narrative et a pour but de montrer les bases du système Ettoy. Les eToys sont créés dans un *monde*. L'unité de sauvegarde ou de publication est un *projet*. Les dessins qui sont *gardés* deviennent des objets manipulables.

La seconde partie du projet 1 demande aux élèves de dessiner leur premier objet, une voiture. Ce faisant les élèves deviendront familiers avec les outils de dessins de eToys. L'activité «Dessiner une voiture» est une activité importante pour la compréhension des projets suivants.







## Présentation des projets

Au début de chaque projet cette colonne introduira :

### Prérequis Squeak

- Les outils spécifiques ou parties du système EToy nécessaires à la réalisation du projet.

### Concepts associés Math

- Les concepts mathématiques ou scientifiques illustrés par le projet. Ces concepts correspondent souvent aux programmes suivis en classe.

### Objectifs d'apprentissage

- Les objectifs et résultats escomptés pour l'élève.

Lancez Squeak, l'environnement contenant les eToys en cliquant sur le raccourci (ou alias) Squeak qui a été créé sur votre bureau après que vous l'avez pris depuis la toile.



Vous devez voir un écran vide contenant deux tiroirs : Navigator et Supplies.



Cet écran est un *monde* dans le jargon eToy, c'est là que vous allez créer, explorer et apprendre! C'est dans ce monde que vous allez construire les projets présentés dans ce livre.

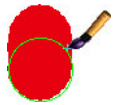
Les projets créés dans ce monde peuvent être composés d'une grande variété de types de médias tels que dessins, textes, vidéo ou photos (jpegs ou bmps). Les projets de ce livre utilisent principalement des dessins (sketches) créés à l'aide des outils de dessin de Squeak. On peut avoir accès aux outils de dessin en cliquant sur le bouton en forme de brosse ou de pinceau qui se trouve dans le tiroir orange Navigator en bas de l'écran.



Les outils de dessin créent des dessins (*sketches en anglais*) qui lorsqu'on décide de les garder, deviennent des objets. Ces objets peuvent être programmés pour se comporter selon vos désirs.

Ce sont les projets qui sont sauvés ou publiés en Squeak. Les projets peuvent être publiés sur votre disque dur local, sur un serveur local de votre classe ou directement sur la toile. Les commandes pour publier un projet se trouvent dans le tiroir de navigation Navigator. Si vous pressez sur le bouton **Publish It!** du tiroir Navigator vous obtenez une liste d'options. Quand vous sélectionnez une des options pour publier un projet, vous allez devoir donner un nom à votre projet. Une fois publié, le projet sera sauvegardé dans un fichier terminant par «.pr»(indiquant que le fichier est un projet Squeak) dans votre disque local dans le dossier MySqueak.

Ce projet est le premier d'une série de projets utilisant une petite voiture. Lancez Squeak. Ouvrez l'outil de dessin en cliquant sur la brosse dans le tiroir de navigation Navigator.



Sélectionnez une jolie couleur et utilisez la plus grosse brosse pour peindre un ovale. Cela va être le corps de votre voiture.



Sélectionnez une brosse de plus petite taille et dessinez des roues.



Sélectionnez la couleur blanche et dessinez un petit pare-brise. Finalement dessinez les feux de la voiture. Pour cela utilisez une couleur différente de celle de la voiture.

Quand vous êtes satisfait de votre dessin, cliquez sur le bouton **Keep**. Cela va transformer votre dessin en un objet et cacher les outils de dessin.



Placez le curseur de la souris au centre de la voiture et attendez un moment. Des icônes doivent apparaître de manière transparente. Si vous cliquez sur n'importe quelle icône toutes les icônes apparaissent complètement. Sur Windows Alt-click, click droit sur Gnu/Linux ou Command-click sur Mac, font apparaître les icônes d'un objet directement.



La flèche verte située au centre de l'objet indique la direction qu'il va suivre. Cette direction peut être changée avec la souris. Un nom peut être donné à une voiture. Il suffit pour cela de cliquer sur «sketch».



Cette colonne dans les projets introduira :

## Défi

- Défis suggérés qui permettent une extension du projet et utilisent les concepts et objectifs du projet.

## Remarques

- Notes associées à l'intention des parents ou enseignants.

## SAUVER VOTRE TRAVAIL : UNE EXPLORATION

Quand vous téléchargez Squeak dans votre ordinateur, un répertoire est créé sur votre bureau appelé MySqueak. C'est dans ce répertoire que les projets enregistrés seront stockés. Pour placer un projet dans ce répertoire, cliquez sur le tiroir Navigator puis, appuyez sur le bouton **Publish It !**. Une fenêtre de dialogue apparaîtra sur la droite.



### Navigator

Share < PREV NEXT > PUBLISH IT! Newer? Tell! FIND Brow

### Please describe this project

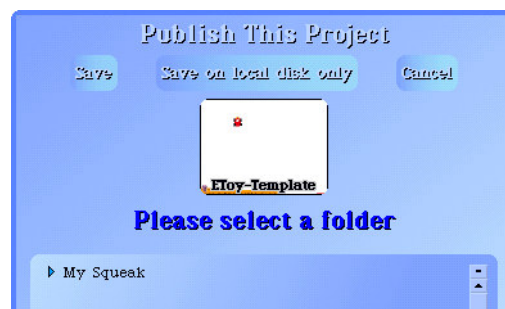
Name: EToy-Template

OK

More

Cancel

Une fois le projet nommé, cliquez sur le bouton **OK** dans la fenêtre. Quand la fenêtre bleue de publication apparaît, cliquez sur MySqueak pour le sélectionner puis cliquez sur **Save**. Votre projet sera sauvegardé dans le répertoire MySqueak situé sur le bureau de votre ordinateur.



Maintenant que le projet est sauvegardé, vous pouvez quitter Squeak en cliquant sur le bouton **QUIT** du tiroir Navigator.



### Chargement de projets

Pour charger un projet dans Squeak, ouvrez le tiroir Navigator et cliquez sur le bouton **Find**.

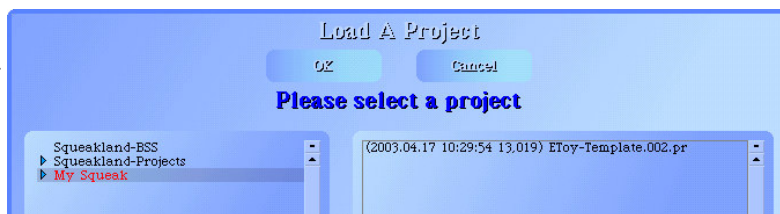
### Navigator

Share < PREV NEXT > PUBLISH IT! Newer? Tell! FIND Brow

Quand la fenêtre bleue apparaît, sélectionnez MySqueak. Tous vos projets sauvegardés apparaîtront sur le côté droit de la fenêtre.

Sélectionnez le projet que vous voulez ouvrir en le sélectionnant puis

en cliquant sur le bouton **OK**. Cette opération chargera le projet sélectionné dans Squeak.





## PROJET 2

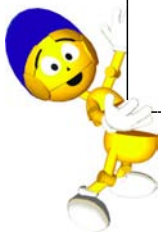
# JOUER AVEC LA VOITURE



Dans ce projet, les élèves vont explorer la manipulation et le changement des caractéristiques ou des comportements de leur objet (la voiture) en changeant ses valeurs et réglages dans le monde et dans son panneau de contrôle.

L'objet dessiné offre aux élèves une représentation sous forme d'icône. Le panneau de contrôle révèle les caractéristiques de l'objet (sa position dans le monde, son orientation, sa taille et sa couleur) à travers une représentation symbolique (en montrant des valeurs numériques). Le «jeu de Montessori» se réalise au travers de la récompense des enfants en terme d'une coordination main-oeil, alors qu'ils acquièrent progressivement de la vitesse dans la compréhension et l'utilisation de la puissance des symboles.

Pour des enfants, se rendre compte que la direction est un nombre, est une idée très puissante. De même, considérer la position de la voiture dans le monde comme des nombres, est aussi puissant. Durant l'exploration et le jeu, ils verront comment chacune de ces valeurs peut être positive, négative ou nulle.



## Prérequis Squeak

- Dessiner.
- Nommer les objets.
- Définir la direction avant d'un objet.
- Obtenir le panneau de contrôle d'un objet.
- Sauvegarder un projet.

## Concepts associés Math

- Le concept de coordonnées «x» et «y».
- Le concept de nombres positifs et négatifs.
- Directions comme angles et nombre de degrés dans un cercle.

## Objectifs d'apprentissage

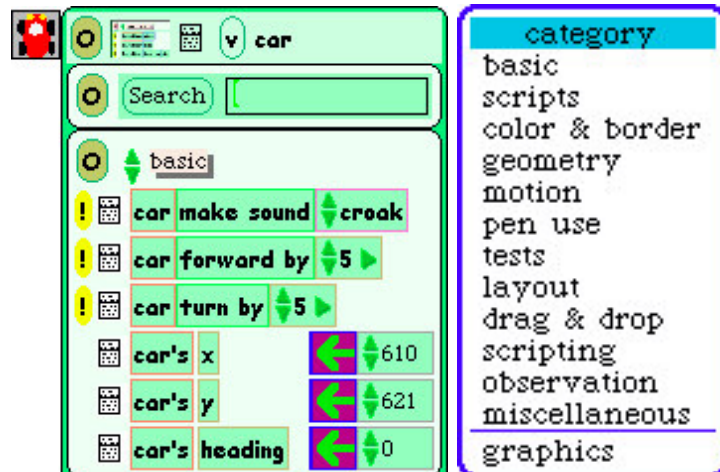
- Comprendre les systèmes de coordonnées.
- Fournir les bases pour la compréhension de formes géométriques simples et d'angles.

Dans ce projet, vous allez explorer votre dessin de voiture comme un objet. Si vous avez sauvegardé votre voiture comme un projet, vous devez charger ce dernier maintenant. Si vous n'avez pas sauvegardé votre voiture, vous avez besoin de dessiner une nouvelle voiture pour ce projet.

Placez la souris sur la voiture ou cliquez sur elle pendant que vous enfoncez le bouton Command (Mac) ou Alt (Windows) pour révéler ses icônes. Si vous laissez le curseur de la souris sur n'importe quelle icône pendant un moment, vous verrez apparaître un ballon d'aide pour vous en rappeler la fonction. La figure ci-dessous montre la voiture avec toutes ses icônes et la description du rôle de chaque icône.




Cliquez sur l'icône turquoise «eyeball» pour ouvrir le panneau de contrôle de l'objet. Le panneau de contrôle ci-dessous montre la catégorie «basic». Des catégories supplémentaires peuvent être révélées en cliquant sur le nom de la catégorie ou sur les flèches, vers le haut ou vers le bas, à la gauche de n'importe quel nom de catégorie.



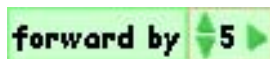
Cliquez sur l'icône bleue «rotation» de la voiture et déplacez la vers la gauche et la droite. Remarquez le changement de la valeur dans la brique `car's heading` (direction) du panneau de contrôle.

Prenez la voiture et déplacez-la dans le monde. Remarquez les changements dans les briques valeurs `car's x` (x de la voiture) et `car's y` (y de la voiture).

Regardez les briques dans le panneau de la catégorie **basic**. Vous verrez deux types de briques : certaines sont précédées  par un point d'exclamation jaune et d'autres pas. Les briques qui suivent un point d'exclamation sont des *briques d'action*.

Cliquer sur un point d'exclamation exécute l'action une seule fois. Maintenir un point d'exclamation «enfoncé» fera «tourner» l'action de manière répétée. Essayez de conduire la voiture «autour du monde» en utilisant les points d'exclamation.

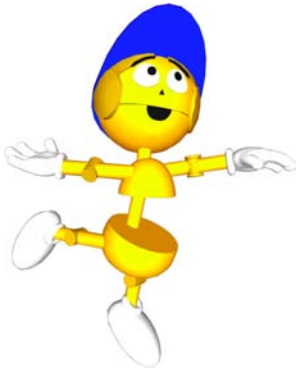
Les valeurs numériques des briques peuvent être changées soit en cliquant sur les flèches vers le haut ou vers le bas, soit par la sélection de la valeur courante, puis la saisie d'un nouveau nombre suivi de «return» (ou «enter»). Jouez avec ces nombres et utilisez le point d'exclamation pour continuer à explorer.



Qu'arrive-t-il à votre voiture si vous rendez négative la valeur de la brique `forward by`, puis si vous cliquez ou maintenez enfoncé le point d'exclamation? Explorez encore et expérimentez!

Les briques qui ne sont pas précédées par un point d'exclamation sont des briques «valeur». Chacune d'elles est suivie d'une flèche verte qui affecte ou initialise cette valeur (en remplaçant par n'importe quel nombre). Changez les valeurs de ces briques et remarquez ce qui arrive à la voiture.





Des **scripts** sont créés pour les objets en groupant des briques. Afin d'obtenir un script, glissez la brique **Voiture emptyScript** (ou «Sketch» si vous n'avez pas nommé votre dessin «Voiture») à partir de la catégorie **scripts** vers le monde. Si la catégorie **scripts** n'est pas visible dans le panneau de contrôle de votre objet, faites-la apparaître là maintenant (comme expliqué précédemment).



Pour créer un script, ajoutez les briques **Car forward by 5** au script en les déposant dans le script vide. Quand un script est prêt à «accepter» des briques, un rectangle vert clair apparaîtra pour indiquer qu'il est prêt.

**Note :** Le curseur *doit* être dans le script afin qu'il puisse accepter des briques. Si le graphique correspondant à la brique est dans le script sans que le curseur le soit, les briques ne seront pas acceptées dans le script.



N'importe quelle chaîne de briques action (celles précédées d'un point d'exclamation) créera son propre script quand elle est déposée dans le monde. Les briques valeur doivent être ajoutées à un script ou à une chaîne de briques déjà dans un script. Elles ne créeront pas leur propre script une fois déposées dans le monde.

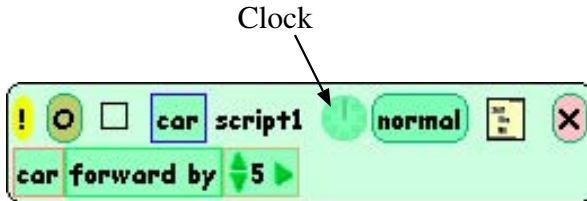
Des briques mal placées peuvent être supprimées de deux façons : en les faisant glisser dans la corbeille (disponible dans le tiroir Supplies) ou en faisant apparaître leur icône puis en cliquant sur l'icône «détruire (delete)» (X). Une fois que les briques ont été liées, elles ne peuvent être dissociées. Si vous avez créé une chaîne de briques par accident ou si vous trouvez des briques dont vous n'avez plus besoin, jetez-les dans la corbeille.

Corbeille

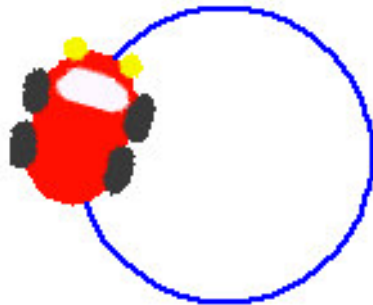




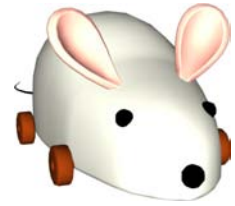
Les scripts peuvent être «activés» en cliquant sur la petite horloge du script ou en sélectionnant le bouton «normal» et en changeant son réglage à «ticking» (marche). Pour exécuter une seule fois un script, cliquez sur le point d'exclamation.



Que pouvez-vous ajouter à ce script pour faire à la fois avancer et tourner la voiture? Des tracés de crayon peuvent être définis pour n'importe quel objet, de sorte à visualiser son mouvement dans le monde. Faites apparaître la catégorie **pen use** dans le panneau de contrôle de la voiture. Mettez la valeur de `penDown` à «true» et mettez le script en mode execution automatique («ticking»). Notez les tracés laissés par la voiture. La taille du crayon peut être modifiée en augmentant la valeur numérique correspondante. La couleur du crayon peut également être modifiée.

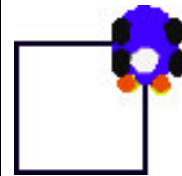


Lorsqu'un projet contient plusieurs scripts, il est sage d'utiliser le groupe de boutons «stop step go» qui se trouve dans le tiroir Supplies et de le placer sur le Monde. Un clic sur **go** placera tous les scripts de votre projet en mode execution automatique («ticking»). Un clic sur **stop** arrêtera tous les scripts.



## Défi

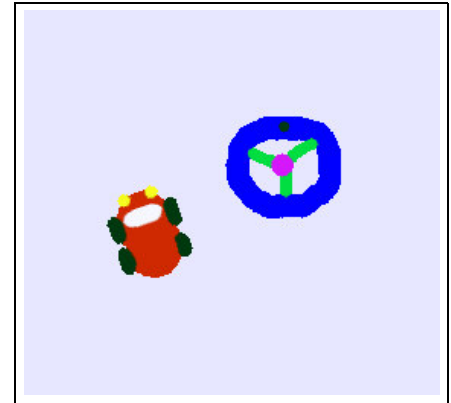
- Explorez pour découvrir combien de figures géométriques peuvent être créées en réalisant des scripts simples pour la voiture.





## PROJET 3

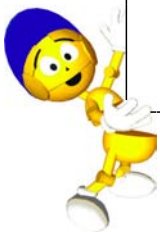
# CONDUITE D'UNE VOITURE AVEC UN VOLANT



Dans ce projet les élèves vont apprendre à créer deux objets et à les faire interagir. Jusqu'à présent, les élèves ont manipulé un seul objet avec un script. Dans ce projet, les élèves vont devoir prendre des briques d'un objet (le volant) et les ajouter à un script relatif à un autre objet (la voiture). Ceci demande un effort de pensée pour un élève et peut prendre un peu de temps pour certains.

Ce projet nécessite la création d'un eToy avec plusieurs objets ainsi que la création de scripts pour ces objets. Certains scripts vont contenir des propriétés de deux objets. Ce projet nécessite aussi qu'un élève passe du panneau de contrôle d'un objet à l'autre et donc pense aux différents panneaux de contrôle présents dans le projet.

Une fois ce projet fini, les élèves sont prêts à créer d'autres projets utilisant plusieurs objets avec des scripts semblables. C'est souvent à ce moment que les élèves changent leurs objets en animaux, avions ou autres petites images car ils découvrent que les capacités de Squeak sont aussi grandes que leur imagination.



## Prérequis Squeak

- Dessiner.
- Nommer des objets.
- Définir la direction d'un objet.
- Créer des scripts.

## Concepts associés Math

- Le concept de nombres positifs et négatifs.
- Le concept de direction.

## Objectifs d'apprentissage

- Assimiler le fait que la direction de la voiture est liée à la direction du volant.
- Comprendre comment la valeur numérique de la direction du volant peut aider à conduire la voiture.
- Comprendre comment les briques textuelles sont des représentations de valeurs.
- Formuler des hypothèses basées sur des découvertes et faire des prédictions.


Dans ce projet vous allez apprendre comment conduire une voiture avec un volant.

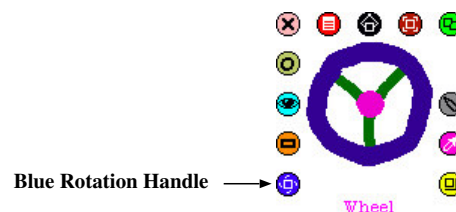


Dessinez une voiture et un volant. Ces objets doivent être dessinés séparément. Un exemple de chaque est montré au-dessus. Donnez un nom à la voiture et au volant.

Le but de cette leçon est d'utiliser le concept de nombres positifs et négatifs pour aider à conduire la voiture. Pour faire cela, vous allez devoir connecter ensemble la voiture et le volant. La voiture sera l'objet que l'on scripte.

Ouvrez le panneau de contrôle de la voiture pour voir ses briques et prenez la brique forward by (avance de) et posez-la sur le monde. Prenez la brique turn by (tourne de) et placez la dessous la brique forward by à l'intérieur du même script.

Faites apparaître les icônes du volant et cliquez sur l'icône «eyeball»  pour faire apparaître ses briques. Maintenant cliquez et maintenez enfoncée l'icône bleue «rotation» du volant qui permet de tourner le volant à droite et à gauche. Pendant que vous tournez le volant, regardez dans le panneau de contrôle et faites attention à ce qui se passe avec la valeur numérique de la brique direction.



Il est important de comprendre qu'une fois que les deux objets sont connectés ensemble et que la direction du volant indique zéro, la voiture va aller tout droit. Quand la direction du volant indique un nombre négatif, la voiture va tourner sur la gauche, quand il indique un nombre positif, la voiture va tourner sur la droite.

L'étape suivante est donc de connecter le fait que la voiture tourne avec la direction du volant. Pour cela on prend la brique `heading` (direction) du volant et on la place sur la *valeur* de la brique `turn by` (tourne de) de la voiture.



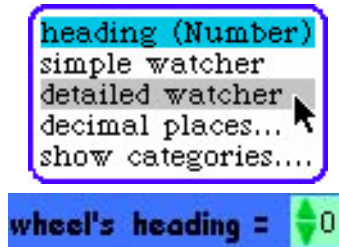
Une fois le script créé, vous pouvez conduire la voiture. Dans la catégorie **basic** du volant, mettez la valeur de `heading` à 0. Lancez le script et rapidement faites apparaître les icônes du volant. Maintenez appuyée l'icône bleue «rotation» et conduisez la voiture.

Comme les deux objets sont connectés, si vous faites tourner le volant en utilisant l'icône bleue «rotation», la voiture va tourner dans la même direction.

Si vous avez des problèmes pour comprendre la direction que prend la voiture, imaginez-vous à la place du conducteur à l'intérieur de la voiture. Il est possible que vous ayez besoin de ralentir la voiture, pour cela changez la valeur de la brique `forward by`. Un *indicateur* détaillé est un autre outil de Squeak qui peut vous aider. Pour en obtenir un, cliquez sur le petit menu que l'on trouve à droite de la brique `wheel's heading`.

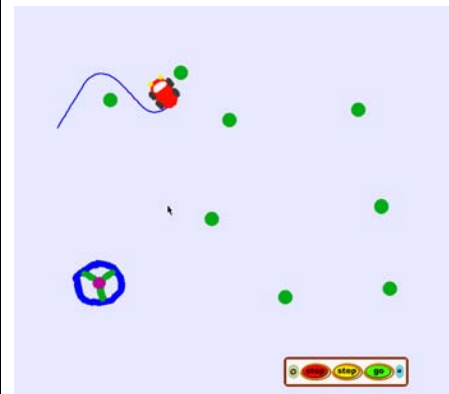


Sélectionnez «detailed watcher» (indicateur détaillé). Placez l'indicateur sur le monde. Maintenant, quand vous conduisez la voiture, vous pouvez utiliser les changements de valeur de l'indicateur comme guide. Rappelez-vous qu'une valeur nulle pour la direction signifie que la voiture va aller tout droit. Amusez-vous bien en conduisant.

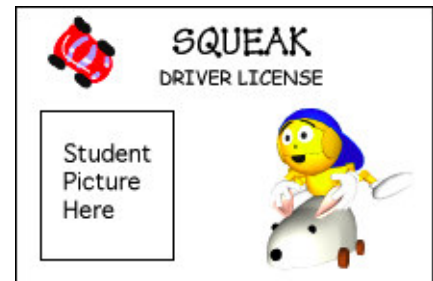


## Défi

- Construisez une course d'obstacles. Essayez de conduire la voiture au travers des obstacles sans les toucher. Utilisez les stylos pour que la voiture laisse une trace de son passage.



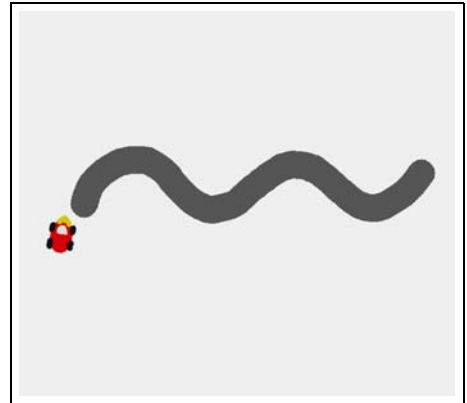
- Décernez-vous un permis de conduire «Squeak».





## PROJET 4

# CREER UNE VOITURE INTELLIGENTE SUIVANT UNE ROUTE

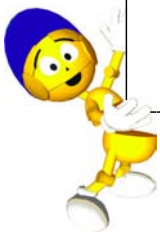


Dans ce projet les élèves vont créer une voiture robot programmée pour suivre, à l'aide d'un détecteur de peinture, une route d'une seule couleur. Ce projet introduit l'important concept de notion de retour d'information et l'utilisation de conditions.

Comme dans le projet 3, les élèves doivent penser à la direction de la voiture et l'utilisation de nombres négatifs et positifs pour contrôler la trajectoire de la voiture. La voiture va exécuter un script utilisant une condition (si/alors) qui teste la couleur perçue par le détecteur de peinture.

Des sons peuvent être ajoutés à ces projets pour offrir une autre dimension au retour d'information. Les sons peuvent indiquer que le véhicule est en dehors de la route ou qu'elle a rencontré un obstacle.

L'exploration de la notion de retour d'information est un moyen pour faire un pont entre le monde des machines et celui de la biologie. Une grande variété de EToys peuvent être créés, basés sur ce modèle de retour d'information pour l'exploration du comportement de fourmis, poissons ou autre animal.



## Prérequis Squeak

- Utilisation de condition test brique (Yes/No).
- Catégorie dans un panneau de contrôle : **tests**.

## Concepts associés

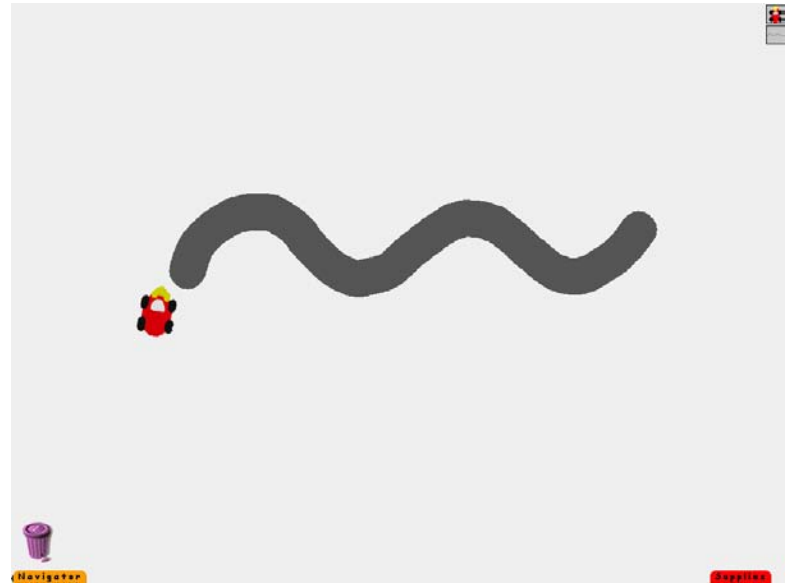
- Condition simple.
- Utilisation de nombres positifs et négatifs pour contrôler une direction.

## Objectifs d'apprentissage

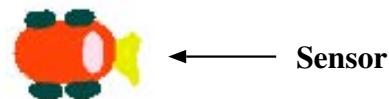
- Comprendre comment les courbes de la route peuvent être liées à la trajectoire de la voiture.
- Utilisation de retour d'information à l'aide d'un capteur pour permettre à la voiture de suivre la route.



L'objectif de ce projet est de «programmer» la voiture afin qu'elle analyse la route et qu'elle suive la route d'elle-même sans utiliser de volant.



L'idée de cette leçon est d'utiliser un détecteur de peinture ayant une seule couleur pour créer un retour d'information. Cet exemple utilise un détecteur jaune placé à l'avant de la voiture.



Un retour d'information est utilisé pour déterminer la direction dans laquelle la voiture doit tourner pour corriger sa trajectoire et suivre la route.

Ce projet nécessite le transfert des connaissances acquises lors des projets sur les nombres positifs et négatifs. Quand la voiture sort de la route par la gauche, elle doit corriger sa trajectoire en tournant d'un nombre positif et quand elle sort de la route par la droite, elle doit corriger sa trajectoire en tournant d'un nombre négatif.



Le comportement de la voiture est créé en introduisant une brique *test* (ou condition) dans le script de la voiture. Pour trouver une brique de test, cliquez sur la petite icône beige qui se trouve en haut d'un script.



Lorsque vous cliquez sur cette icône, vous obtenez une brique Test/Yes/No qui peut être ajoutée dans un script. L'étape suivante est de donner des instructions au détecteur.



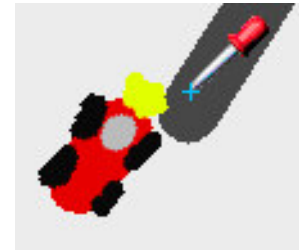
Trouvez la catégorie **tests** dans le panneau de contrôle de la voiture. Sélectionnez les briques color sees (color voit) de cette catégorie.



Prenez ces briques et déposez les dans le script à côté du mot Test.

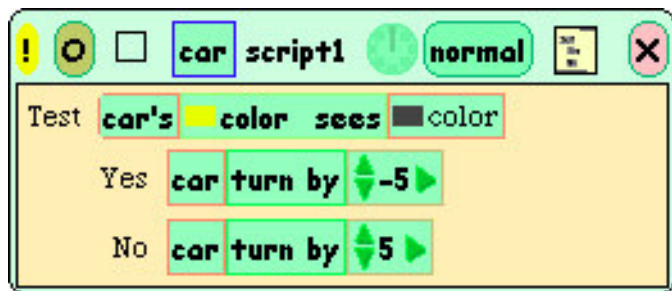


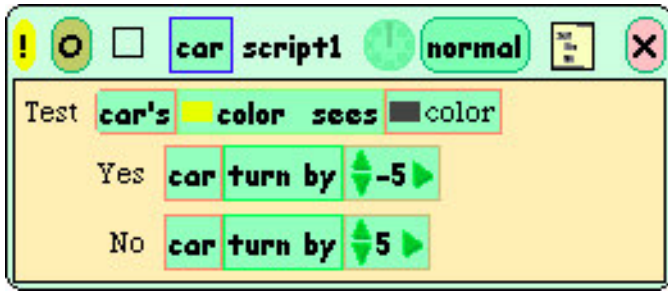
Nous voulons que le *détecteur de couleur* de la voiture «voit» la couleur de la route. Cliquez donc sur le premier rectangle coloré et placez la pipette sur le détecteur de couleur. Le premier rectangle coloré doit maintenant être de la même couleur que le détecteur. Cliquez sur le second rectangle coloré qui suit le mot «sees» et refaites la même opération mais en sélectionnant la couleur de la route cette fois.



Pour compléter le test nous devons déterminer la direction dans laquelle la voiture doit tourner pour suivre la route. Ceci dépend en fait si la voiture est placée à droite ou à gauche de la route.

Dans cet exemple la voiture a été placée sur la gauche de la route.





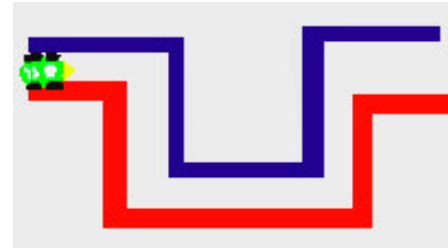
Lancez le script en cliquant sur son horloge ou en changeant le bouton de **paused** (attente) à «ticking»(execution automatique). Est-ce que la voiture suit la route? Si ce n'est pas le cas, ajustez les valeurs des briques turn by (tourne de) jusqu'à ce que cela fonctionne.

Il y a bien sûr d'autres façons de résoudre ce problème. Par exemple, plusieurs parties de la voiture peuvent jouer le rôle de détecteur de couleurs. Pouvez-vous penser à d'autres façons de créer une voiture robot?

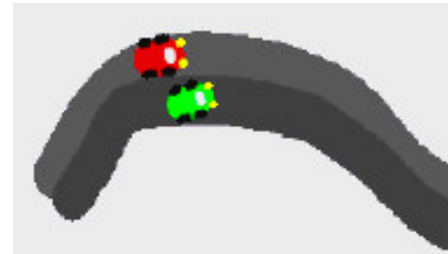


Ben!  
Là, il faut  
*vraiment* réfléchir!!!

### Défis



- En utilisant ce que vous avez appris, programmez une voiture ou un autre objet pour qu'il se déplace dans un labyrinthe.



- Dessinez une deuxième voiture. Placez les deux voitures sur une route à deux voies avec deux couleurs différentes. Pouvez-vous les faire avancer côte à côte tout en restant sur leur propre voie?

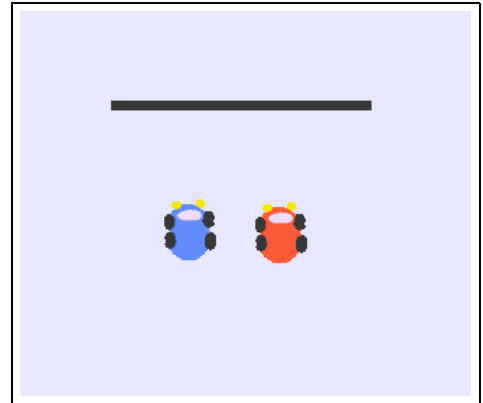
Vous pouvez agrandir la taille de la route ou réduire celle de vos voitures en utilisant l'icône jaune «resize» (change taille).

A quoi d'autre devez-vous faire attention pour réaliser ce défi?



## PROJET 5

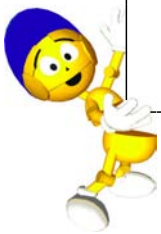
# DEUX VOITURES AVEC DES VITESSES VARIABLES



Ce projet est le premier dans lequel des variables sont créées et utilisées. La notion de nombres aléatoires est introduite et des valeurs aléatoires sont affectées à la vitesse de chaque voiture.

Lorsque les élèves créent plusieurs voitures et affectent des valeurs aléatoires aux vitesses, ils commencent à penser à la vitesse moyenne et la valeur médiane car ils simulent des courses entre leurs voitures et ils peuvent créer des graphes de leurs résultats.

Ce projet établit la base pour une réflexion plus profonde au sujet de cette chose nommée vitesse. De plus, il introduit les concepts de mouvement constant ou accéléré, et de taux de variation de la vitesse.



## Prérequis Squeak

- Utiliser des briques test.
- Utiliser des briques color sees.
- Catégories dans un panneau de contrôle : **scripting**.
- Nommer des scripts.
- Utiliser un indicateur détaillé.

## Concepts associés Math

- Comprendre les variables et comment les créer et les utiliser.
- Comprendre le concept d'aléatoire appliqué aux nombres.
- Comprendre les concepts de moyenne et médiane.
- Utiliser une ligne de points.

## Objectifs d'apprentissage

- Créer une variable dont la valeur est un nombre aléatoire.
- Utiliser une ligne de points pour enregistrer la vitesse de la voiture.
- Utiliser l'information contenue dans une ligne de points pour calculer la moyenne de la vitesse de la voiture.
- Formuler des hypothèses basées sur des découvertes et faire des prédictions.

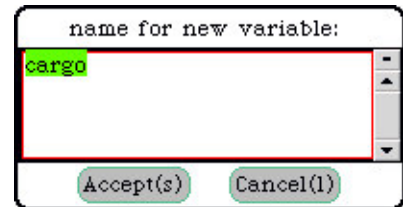
Qu'est-ce qu'une variable ? Une variable est quelque chose qui peut avoir n'importe quelle valeur. Les variables peuvent modifier le résultat d'un événement. La vitesse d'un objet est une variable, tout comme sa position en x ou y dans le Monde. Pour créer une variable nous devons créer un objet. Commencez par peindre une voiture et donnez lui un nom.

Pour créer une variable, cliquez sur le bouton «v» que l'on



trouve dans le panneau de contrôle à côté du nom d'un objet.

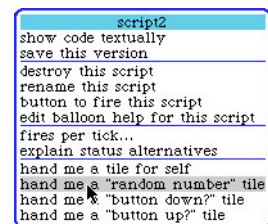
Donnez le nom «speed»(vitesse) à la variable et cliquez sur le bouton **Accept**. Cela crée la variable et l'ajoute au panneau de contrôle de votre objet.



Déposez une flèche verte d'affectation dans le Monde. Cela va créer un script autour de la brique speed (vitesse) de la voiture.



Une fois le script créé, cliquez sur le nom de votre objet. Le menu présenté à droite doit apparaître. Sélectionnez «hand me a random number tile» (donnez une brique nombre aléatoire). Une brique de nombre aléatoire se trouve aussi dans le tiroir Supplies.



Remplacez la valeur de speed dans le script par la brique random (aléatoire). Changez le nombre aléatoire pour qu'il soit plus petit que 20.



Pour faire avancer la voiture en utilisant sa vitesse, déposez les briques car's speed (vitesse de la voiture) et remplacez la valeur de la brique forward by (avancer de) de la voiture par la brique speed.



A chaque tic d'horloge, la voiture va avancer d'un nombre de pixels compris entre 1 et un nombre aléatoire représentant le maximum. Pour voir cela, créer un indicateur détaillé («detailed watcher»). L'indicateur va montrer le nombre de pixels duquel la voiture avance à chaque tic d'horloge.

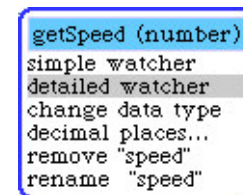
A côté de la brique de la variable speed dans la catégorie variables, il y a une icône qui ressemble à un petit menu. Cliquez dessus et sélectionnez «detailed watcher» du menu. Placez cet indicateur sur le Monde.

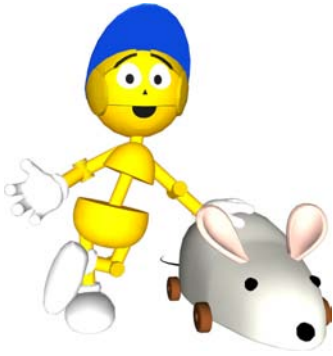
Cliquez sur le point d'exclamation ! du script pour le lancer une fois. Quelle valeur l'indicateur rapporte-t-il comme vitesse de la voiture ? Faites cela plusieurs fois.



## Remarque

- Les projets EToy qui suivent dans ce livre vont utiliser des variables.



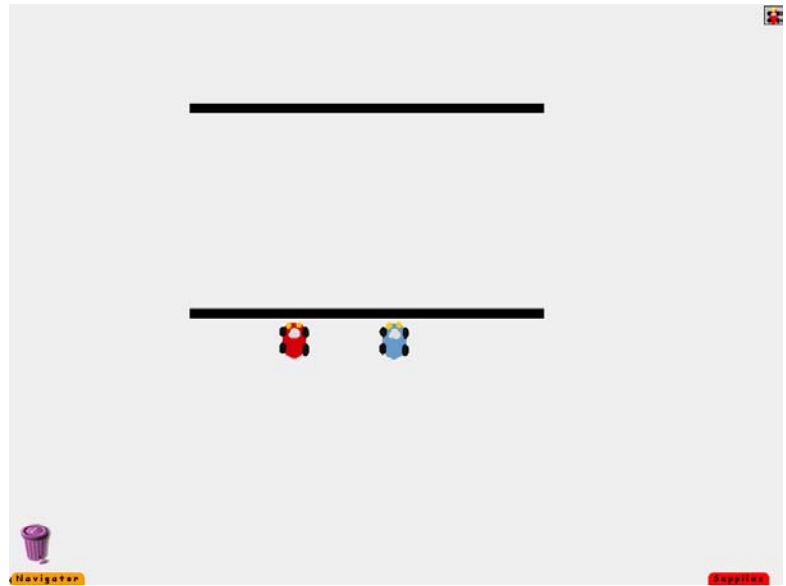


Hé! Est-ce que je peux faire participer ma souris à une course?

Continuez en dessinant une autre voiture. Les deux voitures doivent être dessinées en pointant vers le haut de l'écran. Cela signifie que leur direction sera mise à zéro. Donnez un nom à chaque voiture comme par exemple sa couleur.

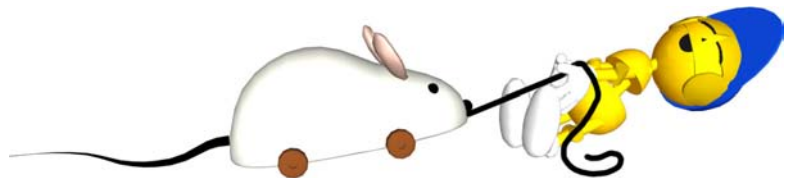


Dessinez une ligne droite. C'est la ligne de départ de la course. Placez-la en bas de l'écran. Dessinez une autre ligne et placez-la au dessus de la première. C'est la ligne d'arrivée.



Faites apparaître un panneau de contrôle pour chaque voiture et ajoutez une *nouvelle variable* à chacune d'elles. Nommez la variable vitesse («speed»). Mettez un nombre aléatoire dans chaque variable vitesse. Dans ce projet, mettez la même valeur maximale pour le nombre aléatoire pour chaque voiture.

Allez avance — Je veux gagner cette course!!!

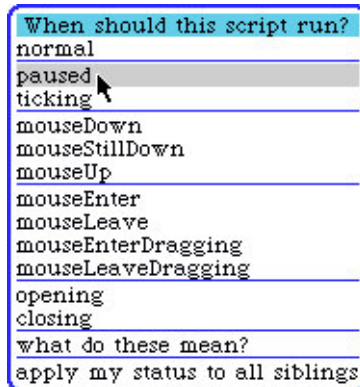
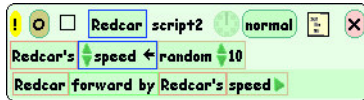




Pour chaque voiture, créez un script programmant la voiture pour avancer de la distance indiquée par sa variable vitesse comme nous l'avons montré auparavant.

Pour chaque voiture, créez un indicateur détaillé.

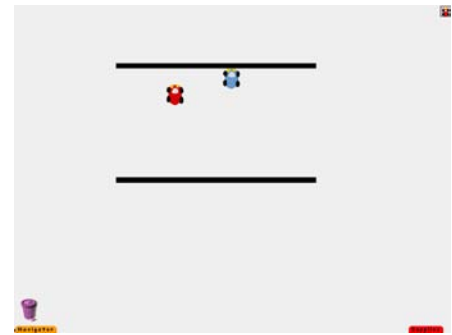
Quand les scripts sont prêts, cliquez sur le bouton **normal** en haut du scriptor et sélectionnez le choix «*paused*» du menu.



Prenez du tiroir Supplies l'ensemble de boutons **stop step go**.



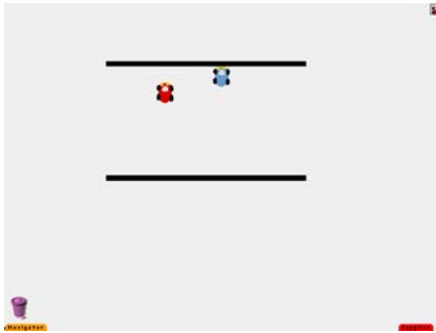
Cliquez sur le bouton **step**. Observez les nombres dans les indicateurs détaillés, changez et représentez les nombres aléatoires générés par la variable «*speed*». Cliquez sur le bouton **go**. Répétez l'expérience. Est-ce toujours la même voiture qui gagne ? Pourquoi ?



Alors on fait la course ?  
Je serai le premier !



## CONDUITE DE VOITURES : PROJET 5



En utilisant le concept de la moyenne vous pouvez dessiner la vitesse avec laquelle chaque voiture bouge à chaque tic d'horloge. De cette manière il est possible de trouver la vitesse moyenne de chaque voiture.

Des lignes de points peuvent être utilisées pour trouver la vitesse moyenne. Les points dessinés sur la grille sont déterminés par le nombre aléatoire qui est généré chaque fois que le bouton step est appuyé.

Blue Car										Red Car									
X										X									
1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

La procédure doit être faite au moins dix fois et les résultats montrés sur une ligne de points. Ensuite la vitesse moyenne peut être déterminée pour chacune des voitures.

Maintenant vous pouvez prédire, à partir de vos résultats, quelle voiture gagnerait si les deux voitures étaient en compétition. Vous pouvez aussi expliquer pourquoi vous êtes arrivé à cette conclusion et le prouver à l'aide des données que vous avez rassemblées.

Pour ajouter une autre dimension à ce projet, vous pourriez demander aux participants que leurs voitures s'arrêtent d'elles-mêmes quand elles franchissent la ligne d'arrivée. Des modifications sont nécessaires pour résoudre cela. Enlevez la ligne de départ et laissez la ligne d'arrivée.

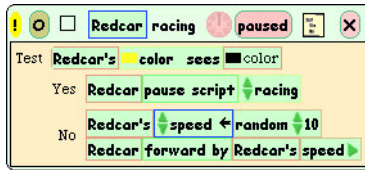
Il y a deux voitures dans ce projet et chacune a un script associé. Il est important de donner des noms à chacun des scripts. Cela vous aidera quand les projets deviendront plus excitants.

Cliquez sur le mot «script» en haut de chaque script pour lui donner un nom. Dans l'exemple suivant le script est nommé «racing»(course).



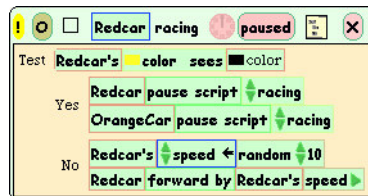
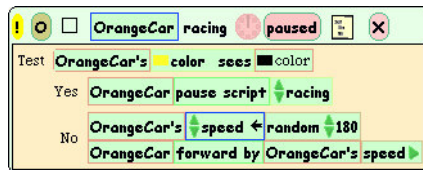
Ajoutez les briques Test dans le script de chaque voiture.

Ajoutez les briques color sees (voit couleur) à chaque script. Pour trouver la brique qui est nécessaire pour arrêter une voiture lorsqu'elle a franchi la ligne d'arrivée, sélectionnez la catégorie **scripting**. Une fois que le script a un nom, toutes les briques de cette catégorie vont être composées à partir du nom que vous venez de donner. Sélectionnez les briques pause script et placez-les à côté de la brique Yes dans le test. Bougez les briques speed et forward by dans la partie No position des tests (comme les exemples ci-dessous). Pour commencer la course, cliquez sur le bouton go button qui lancera tous les scripts.

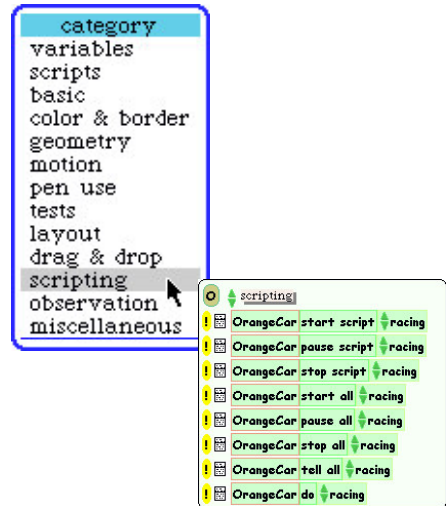
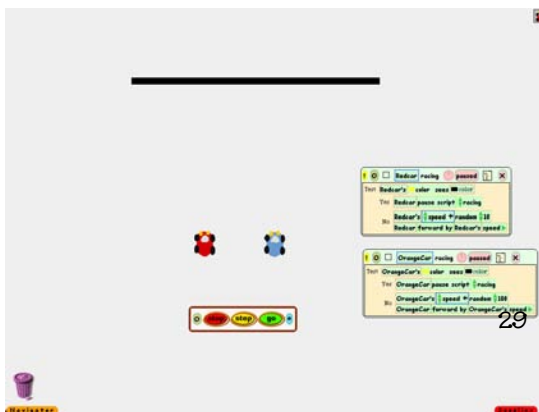


scripts.

Chaque voiture franchissant la ligne d'arrivée s'arrête toute seule. Pour produire l'arrêt de toutes les voitures dès que l'une d'entre elle finit la course, juste quelques nouvelles briques sont nécessaires. Ces dernières sont disponibles dans la catégorie **scripting**. Choisissez les briques pause script de chaque autre voiture et placez-la sous la brique "pause script" dans la brique Yes contenue dans le test. Répétez cette manipulation pour chacune des voitures.



Il y a d'autres façons de résoudre ce problème. La méthode décrite avant est juste l'une d'entre elles.



## Défi

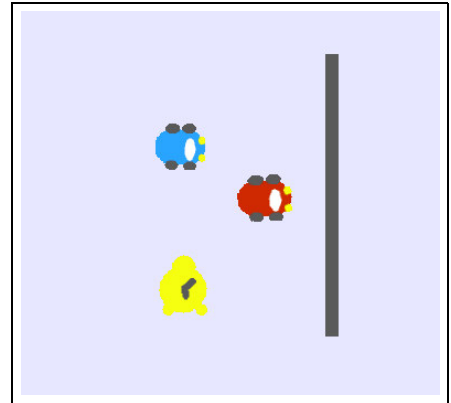
- Aligned plusieurs voitures pour la course. Comment faire pour arrêter les voitures quand elle atteignent la ligne d'arrivée ?
- A combien d'autres façons pouvez-vous penser pour arrêter les deux voitures quand l'une atteint la ligne d'arrivée ?
- Créez un chronomètre qui montre combien de temps il a fallu au vainqueur pour gagner la course.
- Créez un objet qui démarre toutes les voitures en même temps.
- Comment créeriez-vous un bouton remettant à leur place de départ toutes les voitures une fois la course finie ?





## PROJET 6

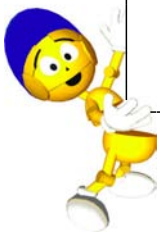
# PLUS DE PIMENT AVEC DES VOITURES DE COMPETITION



Ce projet donne aux élèves l'opportunité de penser encore aux concepts qui leur ont été présentés lors des projets précédents.

Les élèves vont avoir l'opportunité de créer une nouvelle variable, nommée time (temps), et vont continuer à utiliser et à combiner plusieurs outils eToy tels que les boutons, les scripts de remise à zéro (reset) et les instructions conditionnelles.

Ce projet étend la phase d'apprentissage en proposant une nouvelle série de défis basés sur la course entre deux voitures. Il renforce la compréhension des projets précédents.



J'ai une idée qui résoud le défi de l'horloge et peut-être qu'on va me laisser participer à la course.



### Prérequis Squeak

- Créer des boutons.
- Changer l'étiquette d'un bouton.
- Changer la façon dont un script fonctionne.

### Concepts associés Math

- Le pouvoir de l'addition : «increase by»(ajoute).



The Timer

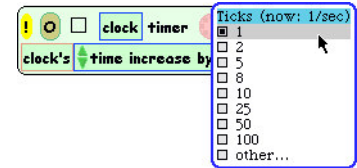


clock's time = 0

Pour résoudre le défi du chronomètre. Dessinez en un et donnez-lui un nom. Ajoutez lui une variable time (temps). Créez un script augmentant time (increase by).



Cliquez sur le chronomètre en maintenant appuyé et changez le «tick time » de telle sorte que le chronomètre pulse toutes les secondes.



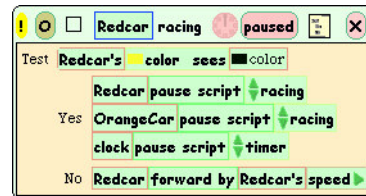
Obtenez un indicateur détaillé pour ce script.

Pour utiliser le chronomètre pour qu'il montre le temps qu'il faut à la voiture gagnante pour franchir la ligne d'arrivée, il faut seulement ajouter un ensemble de briques aux scripts des deux voitures. Dans la brique Yes du test, ajoutez



les briques montrées à gauche.

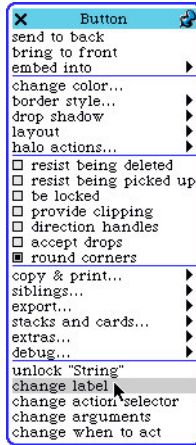
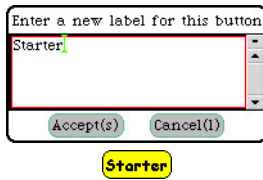
Voici un exemple d'un tel script :



Une façon de résoudre le défi «starter »présenté lors du chapitre précédent est d'utiliser un bouton **Press me** qui se trouve dans le tiroir Supplies. Faites apparaître les icônes du bouton et cliquez sur l'icône «menu».



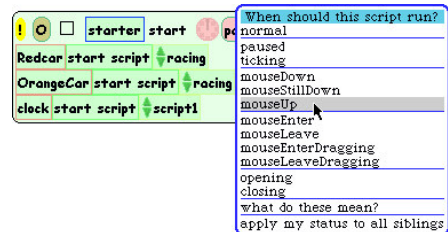
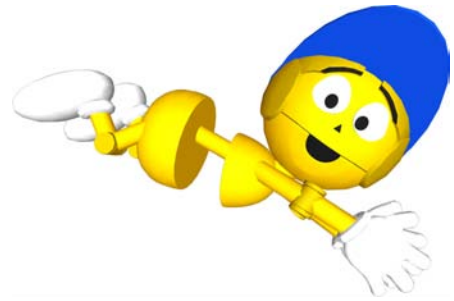
Nommez le bouton «starter»(début).



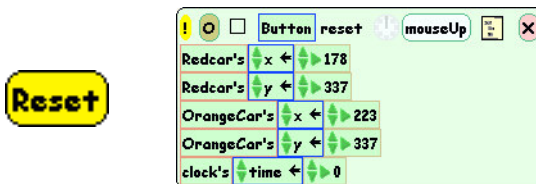
Créez un script pour ce bouton qui va exécuter tous les autres scripts comme par exemple :



Cliquez en maintenant appuyé le bouton **paused** situé en haut du script. Sélectionnez «mouseUp» (souris levée) de façon que les scripts soit exécuté chaque fois que vous relâchez le bouton de la souris en étant sur le bouton que vous venez de créer.



Finalement, faites un bouton qui va remettre à zéro tous les scripts de façon à ce que la course puisse être exécutée plusieurs fois. Nous devons aussi connaître les *coordonnées en x et y de chaque voiture* au début de la course afin de pouvoir les repositionner pour de nouvelles courses. Le bouton **reset** (remise à zéro) replacera les voitures à leur position de départ et remettra à zéro le chronomètre.



## Remarques

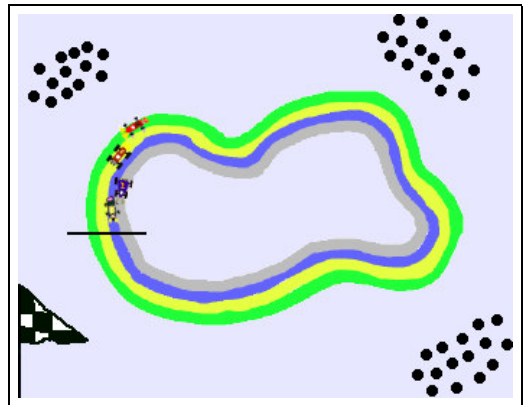
- Il y a plusieurs solutions pour ces défis. Celles présentées sont juste une façon de résoudre ces défis.
- C'est une bonne pratique que de créer des scripts de remise à zéro.





## PROJET 7

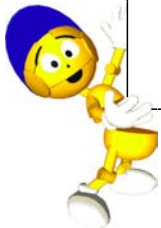
# LA COURSE FANTASTIQUE



La course fantastique est le point culminant de tous les projets précédents. Elle permet aux élèves d'utiliser les concepts acquis jusqu'à maintenant et pour la première fois de prendre un objet ayant été programmé individuellement et de l'importer dans un autre projet.

Les élèves peuvent élaborer des stratégies afin que leur voiture gagne une course sur une piste à plusieurs couleurs et voies. Ceci permet aussi de réfléchir à la longueur totale de chaque voie et si le choix d'une voie peut offrir un avantage. Il est possible de demander aux élèves de déterminer la longueur de chaque voie afin de définir une position de départ juste pour toutes les voitures.

Ce projet met en avant une interaction de groupe et entre élèves puisque chaque voiture rentre en compétition avec les autres lors de chaque course.





Je suis prêt pour la course fantastique!

## Prérequis Squeak

- Dessiner des objets.
- Utiliser des briques test.
- Utiliser des briques color sees.
- Utiliser des briques random.
- Catégorie dans un panneau de contrôle : **scripting**.
- Nommer des scripts.

## Concepts associés Math

- Contrôle du comportement d'objets basé sur la notion de retour d'information (feedback).
- Points de départ décalés pour l'égalisation de la distance sur des parcours de différents rayons.

## Objectifs d'apprentissage

- Synthèse des concepts appris pour créer une voiture en compétition sur une piste à plusieurs voies.



## La course fantastique



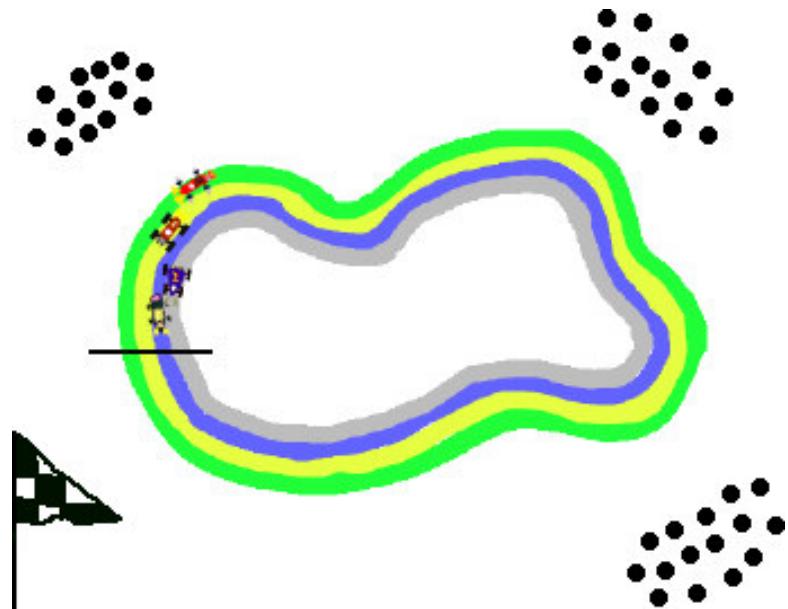
Dans ce projet plusieurs voitures se rencontrent en compétition sur une piste à plusieurs voies.

Pour cette activité, chaque participant se doit de dessiner la voiture la plus folle qui soit, la seule contrainte étant que la voiture doit avoir des lumières qui fonctionneront comme les capteurs de la voiture.

Les participants doivent écrire un script pour leur voiture qui utilise la brique color sees, une variable nommée speed (vitesse) et un générateur de nombres aléatoires. Ils doivent essayer de conduire la voiture sur leur propre voie et trouver la meilleure vitesse pour manoeuvrer la voiture aussi vite que possible tout en restant dans leur propre voie.



Les voitures vont être ensuite importées dans un projet unique qui contient une piste de course ayant quatre voies, chacune de couleur différente.





Une fois qu'une voiture est prête, son projet peut être ouvert dans un ordinateur relié en réseau et une copie de la voiture peut être ajoutée au tiroir Supplies du projet partagé *la course fantastique*. Chacun des participants à la course peut dessiner un nombre indiquant la voie sur laquelle sa voiture va concourir.

Alors des discussions peuvent avoir lieu. Est-ce que toutes les voitures doivent partir de la même place? Est-ce que l'on est avantagé sur certaines voies? Une décision doit être prise et les voitures placées à une position appropriée.

Les règles de la course doivent être expliquées à tous les participants : durant la course aucune voiture ne doit interférer avec les autres et une voiture doit rester sur sa propre voie. Toute interférence avec une autre voiture disqualifie la voiture immédiatement. La première voiture à franchir la ligne d'arrivée est le vainqueur.

Une fois que les voies sont assignées à chaque voiture, chaque participant doit avoir le temps de modifier ses scripts afin d'adapter la voiture à la nouvelle couleur de la voie. Les participants doivent aussi avoir le temps de tester la vitesse aléatoire de leur voiture sur la nouvelle voie et vérifier s'ils veulent changer la valeur limite de la vitesse.



## Remarques

- Préparez-vous pour une leçon pleine d'excitation! Les participants vont s'exclamer et supporter leur voiture.
- Pour ajouter une touche finale à la course, les gagnants de chaque groupe peuvent entrer en compétition afin de déterminer le vainqueur de la classe entière. Prévoyez des prix.

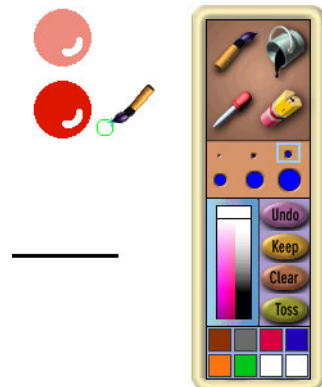
Et le gagnant est...





La création d'animations est une des utilisations du concept «increase by»(augmente de). La suite de ce chapitre montre comment des objets peuvent être transformés et prendre vie en utilisant cette puissante idée. C'est aussi le moment de se détacher de la conduite de voitures et de commencer à s'amuser en créant une grande variété d'objets animés.

Commencez un nouveau projet. Utilisez la palette d'artiste et choisissez le plus gros pinceau pour peindre une balle. Dessinez aussi un reflet sur la balle et peignez une ellipse à une courte distance en dessous de la balle pour représenter son ombre. Validez votre dessin en pressant le bouton **Keep**.

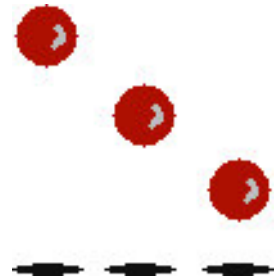


Dupliquez le premier dessin. En utilisant le premier dessin comme guide et de nouveau la palette d'artiste déplacez l'ombre de la balle afin qu'elle soit plus proche de la balle.

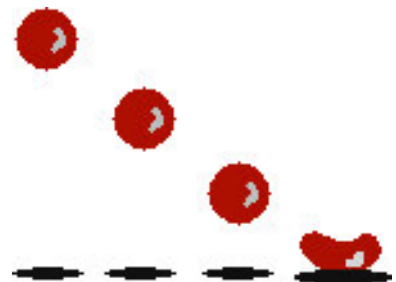


Répétez cette manipulation jusqu'à ce que vous ayez le nombre de balles que vous souhaitez avoir dans votre animation. Cette séquence s'appelle *intermédiaire*.

Dessinez la dernière balle de la séquence plus plate que les autres. Vous pouvez exagérer afin d'obtenir une meilleure animation.



Voici un exemple avec 4 étapes.



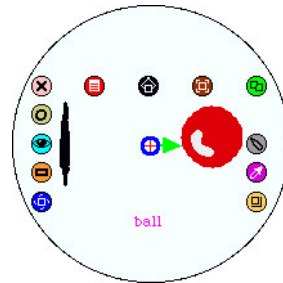
Toutes les balles d'une séquence doivent avoir le même point de référence. Il s'agit du point dans chacune d'elles qui reste le même et à la même place durant l'animation. Dans cet exemple les balles tombent par contre les ombres restent immobiles : les ombres sont donc le point de référence.

# CREATION D'ANIMATIONS : UNE EXPLORATION

Faites apparaître les icônes. Cliquez sur l'icône bleue «rotation». Tournez la sur la droite ou la gauche.

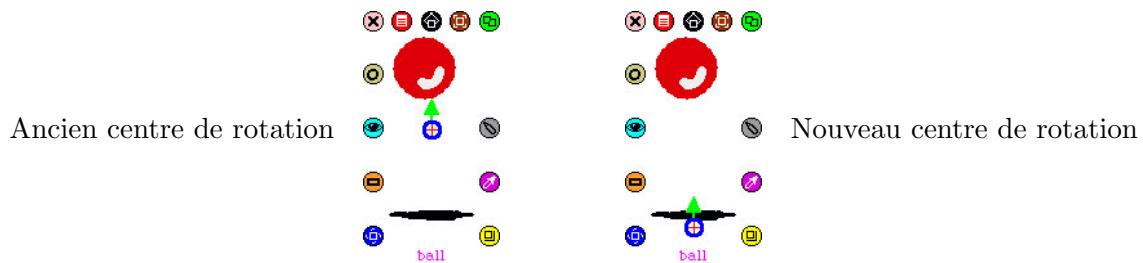


Ce symbole va apparaître au centre de votre dessin. Il indique le centre de rotation de l'objet et le point de référence pour une animation. Si vous imaginez un cercle autour de votre dessin, la balle et son ombre tourne autour du centre du cercle.



Pour que l'animation fonctionne sans accros, le centre de rotation doit être le même pour chaque dessin. Déplacer le centre de rotation en dessous de l'ombre de la balle fait changer son centre de rotation.

Pour changer le centre de rotation, tenez appuyer la touche shift pendant que vous bougez le symbole du centre de rotation à sa nouvelle position. Dans cet exemple, le centre de rotation est bougé du centre à la position juste en dessous de l'ombre de la balle et ceci pour chaque dessin.

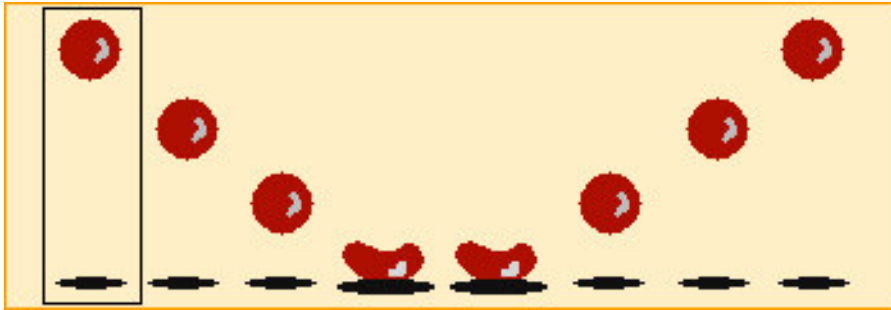


Prenez un conteneur (Holder) dans le tiroir Supplies et déposez-le dans le monde. Faites apparaître l'icône jaune «change size» (changement de taille) et agrandissez le conteneur.

# CREATION D'ANIMATIONS : UNE EXPLORATION

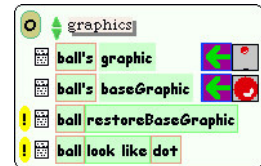
Faites apparaître les icônes. Cliquez sur icône verte «duplicate» (copie) pour faire une copie de votre dessin. Déposez cette copie dans le conteneur. Faites bien attention à ce **que votre curseur soit dans le conteneur** avant que vous ne relâchiez le bouton de la souris.

Déposez chacun des dessins *originaux* qui restent dans le conteneur. Une fois que tous les dessins sont placés dans le conteneur, copiez et placez-les comme montré dans l'exemple. Cette séquence va rendre l'animation de la balle bondissante plus souple. La balle va tomber sur le sol vers son ombre et rebondir.



Le rectangle noir autour de la première balle représente le curseur de l'animation.

Faites apparaître le panneau de contrôle de la première balle que vous dessinez ou d'un autre objet graphique. Dans la catégorie **graphics** de la balle, sélectionnez les briques `ball look like dot` et déposez-les dans le Monde.

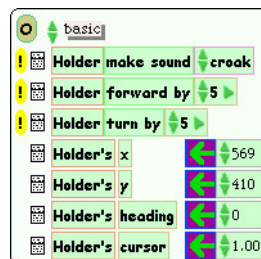
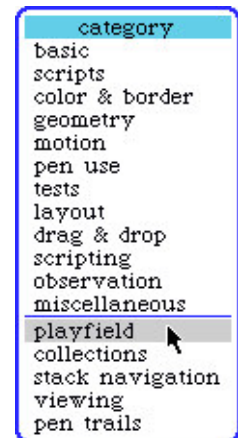


Faites apparaître le panneau de contrôle pour le conteneur. Sélectionnez le menu «playfield» depuis le menu de catégories. Parmi les briques qui apparaissent, sélectionnez les briques `Holder's playerAtCursor` et déposez-les sur le mot «dot» de votre script dans le monde.



De la catégorie **basic** du conteneur, sélectionnez les briques `Holder's cursor`. Déposez ces briques en cliquant sur la flèche verte d'affectation.

Placez ces briques dans votre script.

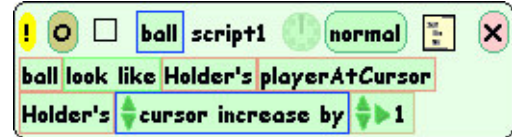


## CREATION D'ANIMATIONS : UNE EXPLORATION

Cliquez sur la flèche montante qui se trouve à droite du mot «cursor»(curseur).



Continuez de cliquer sur la flèche jusqu'à ce que vous ayez changé la brique pour lire increase by (augmente de).



Cliquez sur la petite montre située en haut du scriptor pour lancer votre animation. Essayez de renverser l'ordre des les briques dans le script. Que se passe-t-il ?



## MESURE : UNE EXPLORATION

A ce moment du cours, il est important de mettre l'accent sur le rôle des activités non-liées à l'utilisation d'un ordinateur. C'est le moment d'un transfert entre les idées élaborées avec l'ordinateur et de nouvelles activités.

Les gens sont souvent distraits quand ils cherchent une réponse précise à un problème. L'idée importante ici est de mettre les élèves dans une situation leur permettant de réaliser qu'un concept est plus important que d'essayer de trouver une réponse précise. La science crée des cartes de l'univers. Comme Einstein le faisait remarquer, plus les mathématiques font référence à la réalité moins elles sont certaines, et plus elles sont certaines moins elles font référence à la réalité.

Une partie de ce cours demande aux participants de réfléchir à la distance entre des points ou des traces laissées par des objets. Le point important est qu'ils doivent voir que la distance entre les points augmente ; il n'est pas important de savoir exactement de combien cette distance augmente. Cette activité sans ordinateur peut aider les participants à comprendre qu'il est difficile d'obtenir des mesures précises, que nos mesures peuvent changer suivant les différents outils de mesure et que chaque réponse peut ne pas être exacte mais simplement une approximation. Il est important que les élèves pensent à la notion d'erreur, qu'ils réfléchissent à la précision qu'ils souhaitent obtenir et à partir de quelle erreur ils doivent changer d'outil de mesure.

La suite de ce chapitre est une excursion sans ordinateur qui permet d'obtenir une meilleure compréhension de ce concept.

Matériel :

- pneus de bicyclettes (tous du même type et de même taille),
- petits cubes d'un centimètre,
- ruban,
- fil,
- mètre ruban.

Les participants peuvent être divisés en groupe. On donne à chaque groupe un outil différent pour mesurer la circonférence d'un pneu de bicyclette :

**Groupe 1** cubes d'un centimètre.

**Groupe 2** ruban.

**Groupe 3** fil et mètre ruban.



## MESURE : UNE EXPLORATION

Chaque participant d'un groupe doit mesurer la circonférence du pneu en utilisant l'outil de mesure associé au groupe. Une fois leur tâche accomplie, ils doivent comparer leurs résultats avec les résultats obtenus par les membres de leur groupe ainsi qu'avec les autres groupes utilisant d'autres outils de mesure.

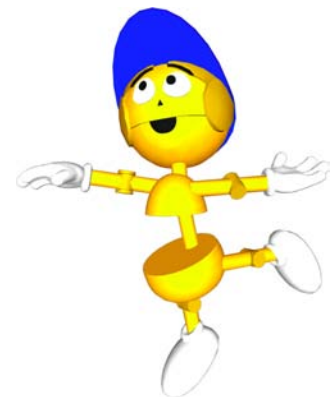
Comme tous les pneus sont tous les mêmes, une riche discussion doit avoir lieu. En effet comment-est-il possible que tout le monde mesure le même objet et obtienne une telle variété de réponses? Quelles sont les variables qui rentrent en jeu lors des mesures? Quelle place peut-on laisser aux erreurs? Est-ce qu'une mesure exacte est importante? Peut-on faire une approximation de la circonférence basée sur toutes les mesures prises?

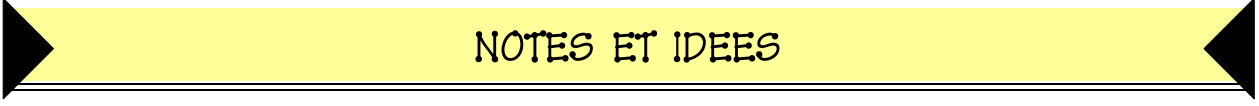
Une autre discussion peut aussi aboutir à savoir si tous les pneus sont effectivement de la même taille. Il se peut qu'ils soient étiquetés comme étant de la même taille mais sont-ils vraiment de la même taille? Quelle est la marge d'erreur du fabricant? Est-il important pour un pneu d'être exactement d'une taille donnée?

Il est important que les élèves se rendent compte que le monde n'est pas exact et que l'on ne peut pas toujours obtenir des réponses exactes à un grand nombre de questions. Une approximation peut aussi être une bonne réponse. Il existe de nombreuses variables qui peuvent affecter le résultat d'un événement. Lorsque l'on crée des simulations, il est impossible de recréer toutes les variables qui peuvent influencer le résultat d'une expérience. Une approximation est alors une réponse correcte que l'on déduit d'une expérience.



Mon pneu est grand comme ça...



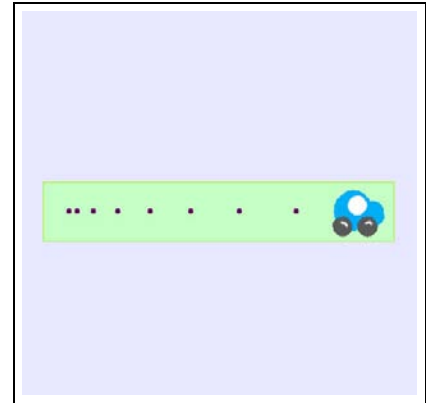


NOTES ET IDEES



## PROJET 8

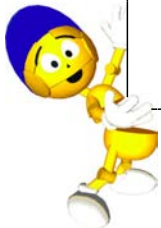
# REFLECHISSONS A LA VITESSE



Dans ce projet, les élèves vont dessiner leur voiture de côté. Cette vue de la voiture de côté va aider à visualiser la vitesse de la voiture.

Ce projet offre un autre exemple de l'utilisation de l'idée puissante d'augmentation («increase by»). La voiture est scriptée pour bouger suivant son axe x, au lieu de la direction («forward by», avance de). Dans cet exemple, les deux briques `car's x` (x de la voiture) et `car's speed` (vitesse de la voiture) vont augmenter soit par un nombre constant soit par un nombre changeant de valeur.

Avec l'utilisation de la trace d'un stylo sous forme de points dans ce projet, les élèves peuvent voir les traces laissées par leur voiture lors de leur mouvement. On peut ainsi leur demander ce qu'ils pensent des traces et comment elles changent lorsque la vitesse est constante ou augmente. Ceci permet une exploration de la notion de mouvement et d'accélération.





## Prérequis Squeak

- Faire des boutons.
- Changer les étiquettes des boutons.
- Faire un script pour un bouton.
- Changer la façon dont les scripts s'exécutent.
- Renommer les aires de jeux.

## Concepts associés Math

- La différence entre un nombre constant et un nombre qui augmente de manière constante.
- Le concept d'accumulation.
- Mouvement suivant l'axe x.
- Utilisation de `increase by` pour créer une accélération.
- Mesures.
- Utilisation des variables.

## Objectifs d'apprentissage

- Utilisation de la brique `increase by` pour montrer l'accélération.
- Utilisation des outils autres que des règles comme outil de mesure.
- Utilisation d'une information visuelle pour formuler une hypothèse.

Pour ce projet on doit dessiner une nouvelle voiture. Cette fois-ci la voiture doit être dessinée de côté au lieu du dessus comme cela a été jusqu'à présent.

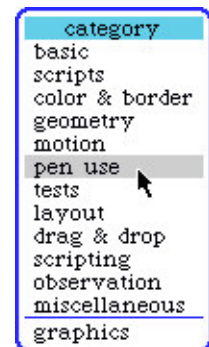
Une fois la voiture dessinée, prenez et déposez une aire de jeux du tiroir **Supplies** et nommez-le **home**. Placez la voiture dans cette aire de jeux.



Home

Prenez une autre aire de jeux du tiroir **Supplies** et nommez-la **TestArea**. Faites apparaître les briques de la voiture. Dans la catégorie **pen use**, changez les briques suivantes :

- Mettez `pen down` à `true`;
- Mettez `pen size` à 3;
- Mettez `trailStyle` à `dots`.




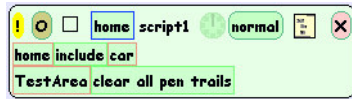
Par exemple :



L'aire de jeux **home** va servir de place dans laquelle la voiture va revenir une fois que nos tests se seront exécutés dans l'aire de jeux **TestArea**.

Faites apparaître les briques pour l'aire de jeux **home**. Dans la catégorie **collections** sélectionnez les briques `home include` (est-ce que l'aire de jeux inclut...) et placez-les sur le monde. Faites apparaître les briques pour l'aire de jeux **TestArea**. De la catégorie **pen use** (utilisation du stylo), sélectionnez les briques `TestArea` et `clear all pen trails` (effacer toutes les marques de stylos) et déposez-les dans le script que vous venez de créer.

Faites apparaître les icônes de la voiture. Cliquez sur l'icône orange  pour obtenir une brique représentant la voiture. Placez-la à côté des briques `home` et `include` dans le script.




Chaque fois que vous cliquez sur le point d'exclamation de ce script, la voiture va retourner au départ (**home**) et toutes les points de la zone d'expérience (**TestArea**) seront effacés. C'est une façon de faire revenir le projet dans son état initial.

Faites apparaître les briques de script de la voiture. Voici les éléments qui doivent être créés pour la voiture :

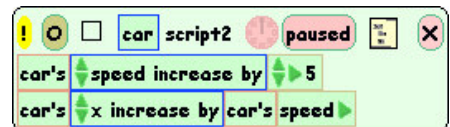
- La voiture a besoin d'une variable nommée vitesse (`speed`).
- On doit associer un nombre à la vitesse.
- La variable `x` de la voiture doit augmenter de la valeur de la vitesse de la voiture.

*Au lieu d'avoir une voiture roulant simplement devant elle, la voiture se déplace maintenant suivant son axe `x` et avec une vitesse.*

Placez la voiture dans la zone de test (**TestArea**). Prenez et déposez les boutons **stop** **step** **go** du tiroir Supplies. Cliquez sur le bouton . Remarquez que les points laissés derrière la voiture lorsqu'elle se déplace sont espacés de la même distance.



L'étape suivante est de trouver un moyen de faire *accélérer* la voiture au lieu de la faire se déplacer à vitesse constante. Pour cela, il est seulement nécessaire de changer une brique du script. Changez la brique `car's speed` du script en **increase by**. C'est une bonne idée de commencer avec un petit nombre comme par exemple 5.



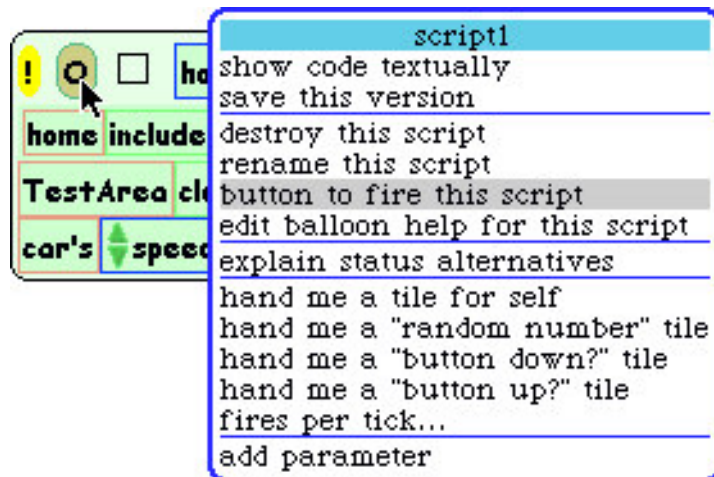
Allons plus vite!!!



Maintenant, nous allons ajouter une brique au script **home**. En effet, comme `increase by` a été ajouté à la variable vitesse (`speed`), la valeur de vitesse va augmenter chaque fois que le script sera exécuté. C'est un phénomène *cumulatif*. La première fois que le script est exécuté, la valeur de vitesse devient 5, mais les fois suivantes, elle va devenir 10, 15, 20 etc... Il est donc important de s'assurer que chaque fois que la simulation est lancée, la valeur de la variable vitesse est remise à zéro.




Voici une autre façon de créer un bouton qui va exécuter ce script. Cliquez sur le mot «home» situé en haut de la fenêtre du script. Dans le menu qui apparaît, sélectionnez le menu «button to fire this script». Une fois le bouton créé, enlevez le script en cliquant sur le cercle orange pâle.



**Home script1** Un bouton pour lancer le script.

Placez la voiture dans la zone de jeux (**TestArea**).



Cliquez sur le bouton  et observez ce qu'il se passe. Discutez de la différence entre la ligne de points créée par cette voiture et celle laissée par la voiture la première fois.



La vitesse augmente (la voiture accélère).



La vitesse est constante (pas d'accélération).

On peut aider à la discussion en utilisant un indicateur pour observer la variable vitesse. Il est important de voir le lien entre la valeur de la vitesse et l'effet qu'elle a sur la variable  $x$  de la voiture et sur l'espacement entre des points laissés par la voiture. Remarquez que la vitesse de la voiture augmente de 5 chaque fois que le script est exécuté. Cela a pour conséquence que la voiture se déplace d'une distance qui augmente à chaque pas. La voiture va donc de plus en plus vite à chaque exécution du script. La voiture accélère.



L'indicateur met en évidence la vitesse de la voiture à chaque exécution du script.

car's speed = 5

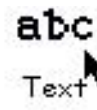
car's speed = 10

car's speed = 15



## Remarque

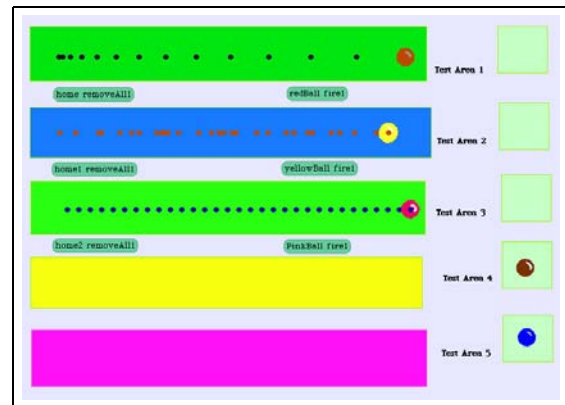
- Vous pouvez laisser les participants écrire leurs remarques et observations directement dans le projet Squeak. Cela peut être fait en prenant un **Livre (Book)** du tiroir Supplies et en le déposant dans le monde. Ensuite prenez un **texte (Text)** depuis le tiroir Supplies et déposez le dans le livre.





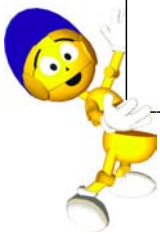
## PROJET 9

# DES VOITURES AUX BALLEES



Ce projet sert de modèle pour provoquer un transfert de connaissance lors du passage de l'utilisation des voitures aux balles qui tombent et à l'exploration de la gravité. Pour chaque élève, il doit renforcer la compréhension des concepts de vitesse *constante*, *aléatoire* et *accélérée*. La plupart des élèves vont facilement faire la transition en pensant que leur objet, précédemment une voiture, maintenant une balle, est un objet portant un costume. Pour les autres élèves il sera peut être nécessaire de rappeler que cette balle n'a rien de nouveau et que les scripts qu'ils ont fait pour la voiture doivent pouvoir s'appliquer.

Demander aux élèves de créer des scripts pour les exemples proposés et leur demander de discuter au sujet de chaque exemple peut servir d'évaluation afin de déterminer le niveau de compréhension de ces concepts.



### Prérequis Squeak

- Création et nommage de scripts.
- Création de variables.
- Utilisation de briques aléatoires.
- Création de boutons.
- Création de scripts de remise à zéro.
- Création d'animation.
- Utilisation et renommage de zone de jeux.
- Création de traces de points.

### Concepts associés Math

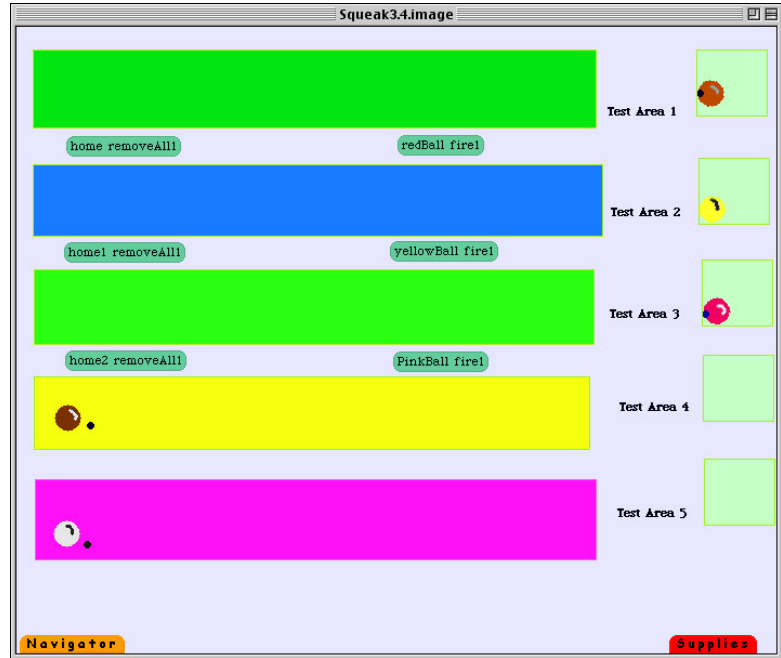
- Comprendre que la différence entre des nombres peut être :
  - constante ;
  - augmentée d'une quantité fixe ;
  - changée de manière aléatoire.

### Objectifs d'apprentissage

- Donner aux participants l'opportunité d'utiliser et d'articuler ce qu'ils ont appris afin d'identifier, résoudre et recréer les problèmes proposés dans ce petit test.



Ce projet peut être préparé en avance par les élèves pour les mettre au défi et afin qu'ils démontrent ce qu'ils ont appris lors des précédents projets. Une fois préparé, le projet peut être installé sur l'ordinateur des élèves.

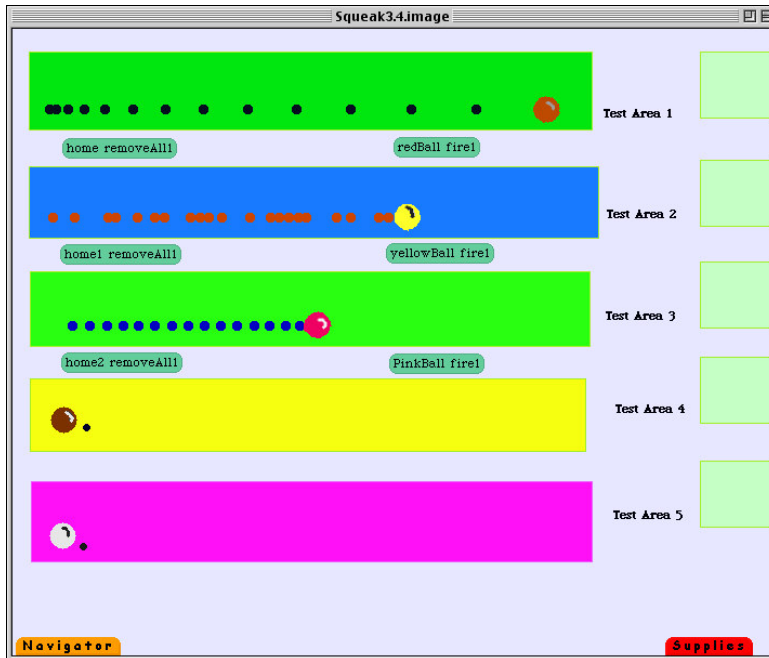


Commencez par créer 10 zones des jeux. Les cinq de gauche, comme montré dans l'exemple ci-dessus, servent de zones de tests. Les cinq de droite servent de point de départ dans lesquelles les balles doivent retourner après avoir été enlevées des zones de tests au moyen de script de remise à zéro.

`redBall fire1`

`home removeAll1`

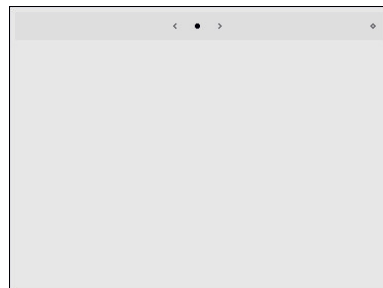
Les boutons ci-dessus sont des exemples de boutons qui ont été créés et placés sous chacune des zones de tests. Les boutons **fire** permettent d'exécuter une action donnée dans la zone de test située juste au dessus d'eux. Chaque fois que le bouton est pressé, la balle devra bouger selon son axe x en suivant les instructions définies dans le script. Le bouton **removeAll** enlèvera la balle et tous les points de la zone et placera la balle dans sa zone de départ.



Les balles bougent horizontalement. La première balle bouge suivent l'axe x et accélère, la seconde se déplace de manière aléatoire et la dernière se déplace d'une distance constante. Les points laissés sont une aide visuelle pour comprendre ce qui se passe dans chacune des simulations.

On peut demander aux participants de regarder chaque exemple et de recréer n'importe laquelle des trois actions qui prennent place dans les zones de tests. Ils doivent créer des boutons qui exécuteront chaque script et des scripts qui les remettront à jour. Ils peuvent aussi créer leurs propres simulations dans les deux dernières zones de tests en bas du projet. Comme dernière étape, les participants peuvent prendre du tiroir Supplies un livre (**Book**) et l'utiliser pour présenter leurs solutions aux problèmes. (N'oubliez pas de déposer un texte (**Text**) dans le livre). Ils peuvent aussi partager leurs solutions avec les autres participants.

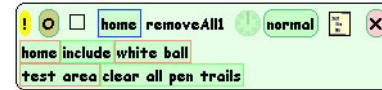
Livre (Book)



## Exemples de solutions

home removeAll1

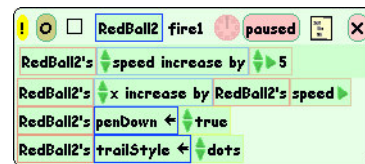
Enlever Tout bouton et script :



redBall fire1

Exécute boutons et scripts :

Accéléré



Aléatoire



Constant





## APPRENDRE LA GRAVITE : UNE EXPLORATION

Les enfants ont aimé participer à tous les aspects de de cette activité.



### Matériel nécessaire :

- Une masse pour le lancer du poids.
- Une pomme.
- Une éponge sphérique.
- Une boule de croquet.
- Une masse percée.
- Des chronomètres.
- Une caméra vidéo.
- Un dispositif pour digitaliser l'enregistrement vidéo et le stocker sur ordinateur.
- Un lieu extérieur avec un toit de plus de 2 mètres de hauteur à partir duquel les balles peuvent être lâchées.

Avant de commencer cette activité, qui sera filmée au caméscope, toutes les balles et la pomme devraient être passées de mains en mains pour que chacun puisse ressentir le poids de chaque objet. Une boule pour le lancer du poids pèse typiquement plus de 5 kg. Son poids surprend les participants qui ont souvent du mal à tenir la boule d'une seule main.

Une fois que tous les participants ont pu se familiariser avec les objets, ils peuvent commencer à essayer de prédire quels objets tomberaient éventuellement au sol le plus vite. Dans notre classe par exemple, les enfants ont classé les objets en cinq catégories en fonction de la durée qu'ils estiment nécessaire à la chute de chaque objet.

Ils ont noté leurs prédictions avec une explication de leur raisonnement.



## APPRENDRE LA GRAVITE : UNE EXPLORATION



Maintenant allez dehors et lâchez chacun des objets en filmant leur chute. Les participants devraient avoir un chronomètre pour mesurer les durées nécessaires pour atteindre le sol.

Par la suite, les objets peuvent être lâchés par paire pour vérifier l'hypothèse selon laquelle certaines boules tomberaient plus vite. Le chronomètre permet aux enfants de réaliser que la durée de la chute de l'objet ne dépend pas de son poids.

Une fois que tous les objets ont été lâchés et filmés (il est sage de le faire plusieurs fois), vous pouvez utiliser ces données de différentes façons. Vous pouvez par exemple choisir que les enfants voient l'enregistrement immédiatement après l'activité. Ceci peut initier une riche discussion sur ce qu'ils ont cru voir et ce qui s'est effectivement passé. Une observation attentive de la vidéo pourra également montrer que certains facteurs affectent aussi le résultat de chaque lâcher.

Cette activité tire également avantage des capacités multimédia de Squeak. Squeak vous permet en effet d'importer les images contenues dans la vidéo dans un projet et d'animer ensuite chaque image pour recréer la vidéo de la chute (ou de manipuler les images comme bon vous semble). C'est une partie importante de cette expérience et bien-sûr cette méthode peut être utilisée dans de nombreux cas. Notez que la séquence est aussi disponible sur le site <http://www.squeakland.org>.

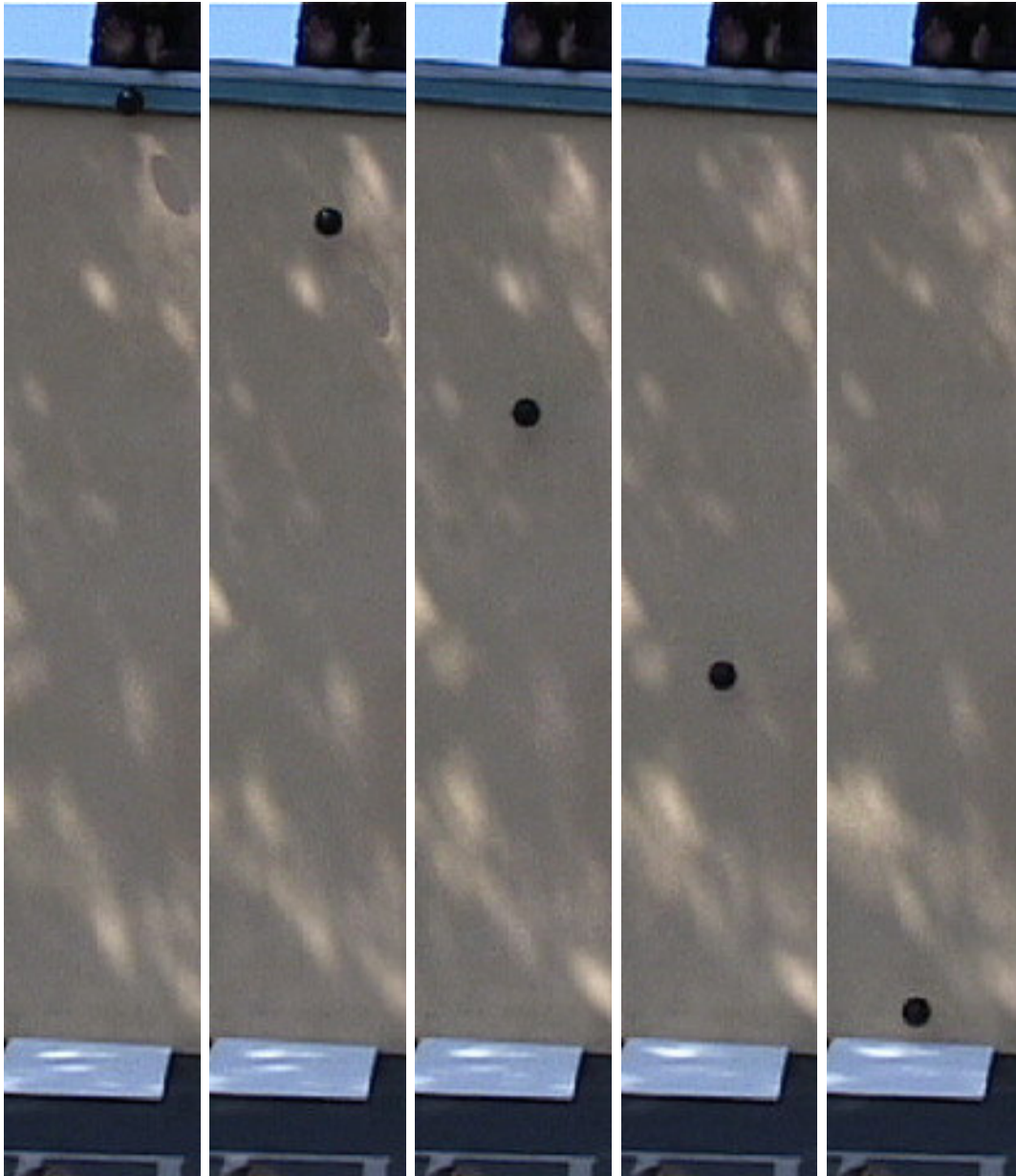




## APPRENDRE LA GRAVITE : UNE EXPLORATION

Vous devrez importer la vidéo dans votre ordinateur sous la forme d'images fixes au format jpeg. A la fois *iMovie* et *Windows Movie Maker* pour les PC peuvent transformer une vidéo importée en une série d'images jpeg.

Comme la chute ne prend environ qu'une seconde, la vidéo est composée d'une trentaine d'images jpeg. Pour intégrer maintenant les images dans Squeak, cliquez sur l'onglet «Navigateur» et cliquez sans relâcher sur le bouton «chercher» («find» dans la version anglaise de Squeak). Sélectionnez «trouver un fichier» («find any file» en anglais) à partir du menu qui s'affiche. Chargez les images jpeg dans votre projet Squeak. En chargeant une image sur six dans l'ordre chronologique vous obtiendrez quelque chose qui ressemble à ceci :

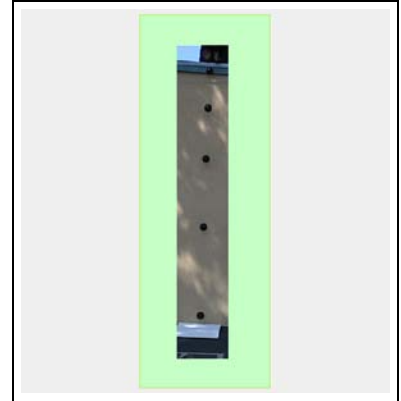


C'est tout pour cette activité, sachez cependant que les participants utiliseront ces images dans les prochains projets pour continuer notre exploration de la gravité.



## PROJET 10

# UN AUTRE REGARD SUR LA CHUTE DES BALLES

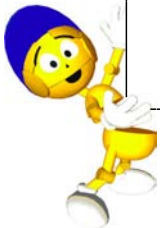


Ce projet EToy découle des excursions décrites précédemment. Idéalement, les élèves devraient avoir déjà entrepris diverses activités de mesure pour comprendre combien il est difficile de trouver une valeur «exacte» ou de répondre à n'importe quel problème, et de pratiquer l'art de l'estimation et de l'approximation.

Les élèves doivent aussi avoir une expérience concrète en ayant observé une variété d'objets lâchés depuis le toit après avoir étudié la théorie de ce qui doit se produire. Des fois, nous insistons sur le fait que «ce que nous avons vu avec nos propres yeux» n'est pas ce qui s'est réellement produit. C'est simplement ce que nous croyons avoir vu, basé sur une fausse croyance.

Ce projet permet aux élèves d'avoir un autre regard sur ce qui s'est produit quand les balles ont été lâchées. Il fournit une approche pour permettre à la fois la décomposition et la reconstruction de l'événement. Le projet offre aussi un moyen «d'arrêter le temps». Ce que nous pouvons voir quand le temps est arrêté peut être différent de ce que nous pensions avoir vu lors d'un court instant du «temps réel».

L'apport des vignettes vidéo dans le projet permet aux étudiants de manipuler une visualisation de l'événement. Fournir une vue impossible dans le monde réel offre aux élèves plus de chances de comprendre le phénomène.



## Prérequis Squeak

- Utilisation des outils d'édition graphique.

## Concepts associés Math

- Mouvement suivant l'axe y.

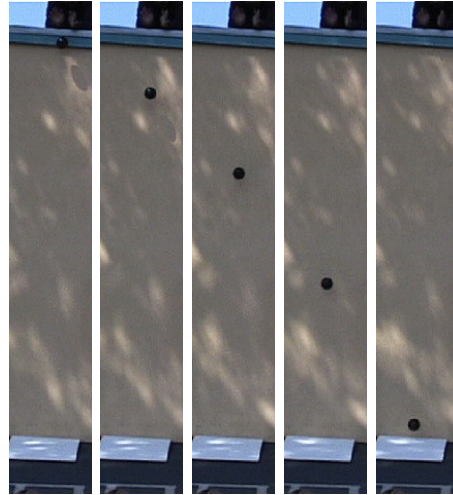
## Objectifs d'apprentissage

- Donner au participant l'occasion d'utiliser ce qu'ils ont appris à propos de l'accélération dans le projet «conduire une voiture» et l'appliquer à ce que vous voyez dans le projet gravité.

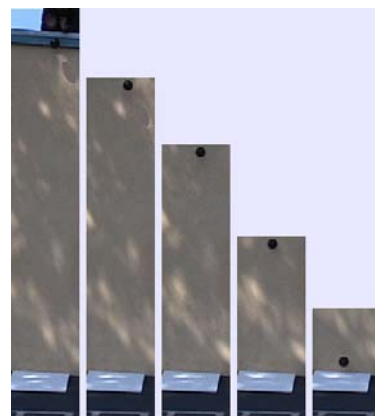
Je tombe!



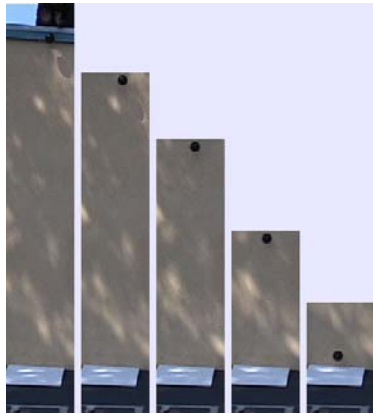
Démarrer ce projet en chargeant les vignettes vidéos dans un nouveau projet, ou en rechargeant un projet sauvegardé.



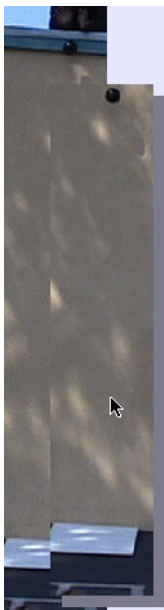
Lorsqu'il verront ces vignettes, certains élèves vont immédiatement faire un «aha!» et voir le schéma. D'autres, auront plus de difficultés avec cette représentation. Pour recréer ce qui a été vu lors de la réalisation de l'expérience dehors et obtenir une balle semblant tomber en ligne droite, les vignettes jpeg peuvent être coupées et empilées. Faites apparaître les poignées de la seconde vignette. Sélectionnez la poignée «repaint» (repeindre). Laissez la balle dans la vignette et effacez tout ce qu'il y a au dessus. Répétez l'opération pour chacune des autres vignettes.



Quand vous aurez fini, les vignettes doivent ressembler à ceci.

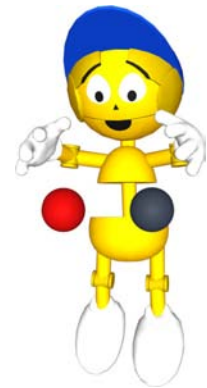
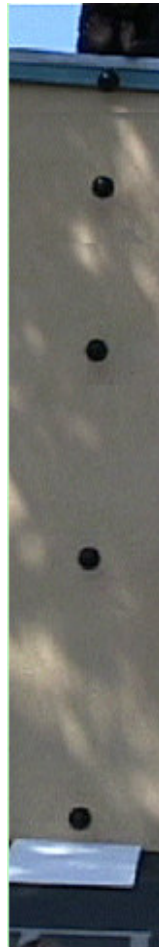


Une fois que toutes les vignettes ont été «élaguées», l'étape suivante consiste à les empiler soigneusement, les unes sur les autres.



Démarrez avec la seconde vignette et déplacez-la au dessus de la première vignette. Utilisez le bas de chaque vignette comme un repère pour placer les vignettes les unes sur les autres.

Quand toutes les trames ont été empilées les unes au dessus des autres, votre image ressemblerait à ceci. A ce stade, les élèves peuvent voir que la vitesse de l'objet qui tombe s'accroît.



### Note :

A ce moment, les élèves de votre cours seront prêts à partager ce qu'ils pensent avoir vu. Plusieurs élèves ont tout de suite vu que la balle paraît accélérer en tombant vers le sol.

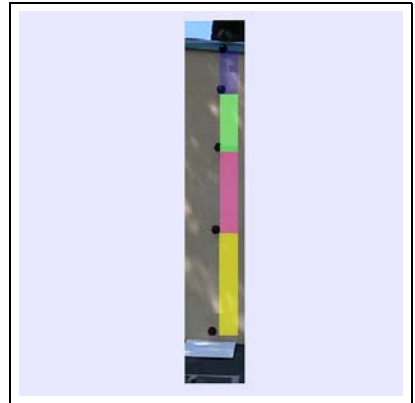
La distance entre les balles paraît s'agrandir. Des étudiants sont même retournés à leur projet voiture et lui ont fait subir une rotation pour comparer les deux. Un transfert de ce qu'ils ont appris a eu lieu.





## PROJET 11

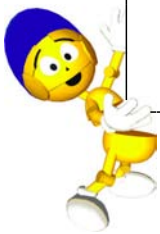
# MESURER LA DISTANCE



Ce projet vise à poursuivre la manipulation et l'analyse des vignettes vidéo coupées et empilées. Il s'agit d'aider les élèves à acquérir une compréhension plus profonde de la signification du schéma produit par la balle tombante.

Pour certains élèves, voir le schéma découvert dans le projet 10 n'est pas une preuve suffisante que la balle accélérât en tombant. L'utilisation d'étiquettes autocollantes (*sticky notes*) transparentes comme outil de mesure pour chercher la distance entre des «photographies» successives de la balle constitue une autre visualisation du comportement de celle-ci.

Après avoir utilisé les étiquettes autocollantes (*sticky notes*) pour mesurer, les élèves peuvent les empiler pour faire un graphe — créant ainsi une autre visualisation pour voir que la vitesse de chaque balle était effectivement croissante pendant la chute vers le sol.



## Prérequis Squeak

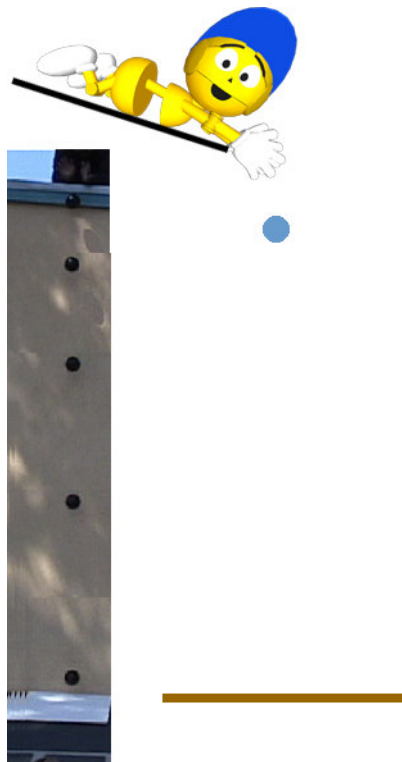
- Utilisation des rectangles comme instrument de mesure.
- Utilisation d'observateurs simples et détaillés.

## Concepts associés Maths

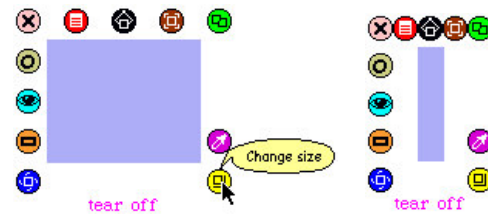
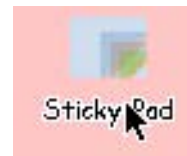
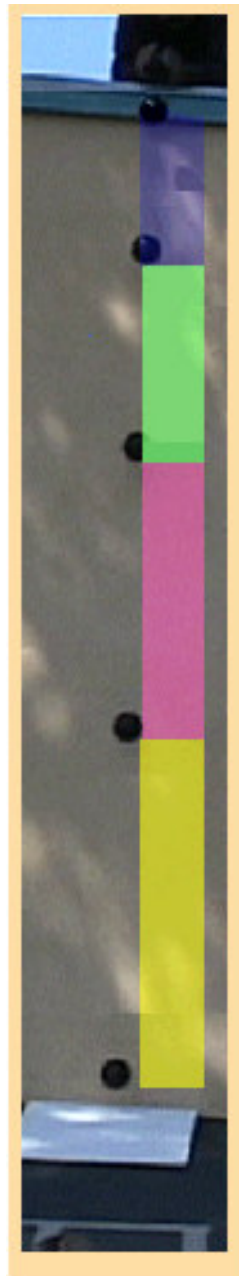
- Mouvement suivant l'axe y.
- Compréhension de la puissance de «increase by» utilisé avec des nombres négatifs.

## Concepts scientifiques

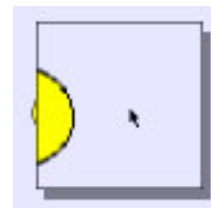
- Comment l'accélération devient un facteur clef pour comprendre la force exercée par la gravité sur les objets tombant.



La pile de vignettes vidéo créée dans le projet 10 est le point de départ de ce projet. Maintenant que la balle tombante est alignée en une colonne bien droite, des étiquettes autocollantes (*sticky notes*) rectangulaires du Sticky Pad (disponible dans le tiroir Supplies) servent à mesurer la distance entre chaque balle. Chaque rectangle sera d'une couleur différente.



La mesure doit être prise depuis le bas de la première balle jusqu'au bas de la balle suivante, ou le plus près possible. Certains participants peuvent choisir d'utiliser, pour plus de précision, une loupe (depuis le catalogue d'objets (*Object Catalog*) dans le tiroir Supplies).

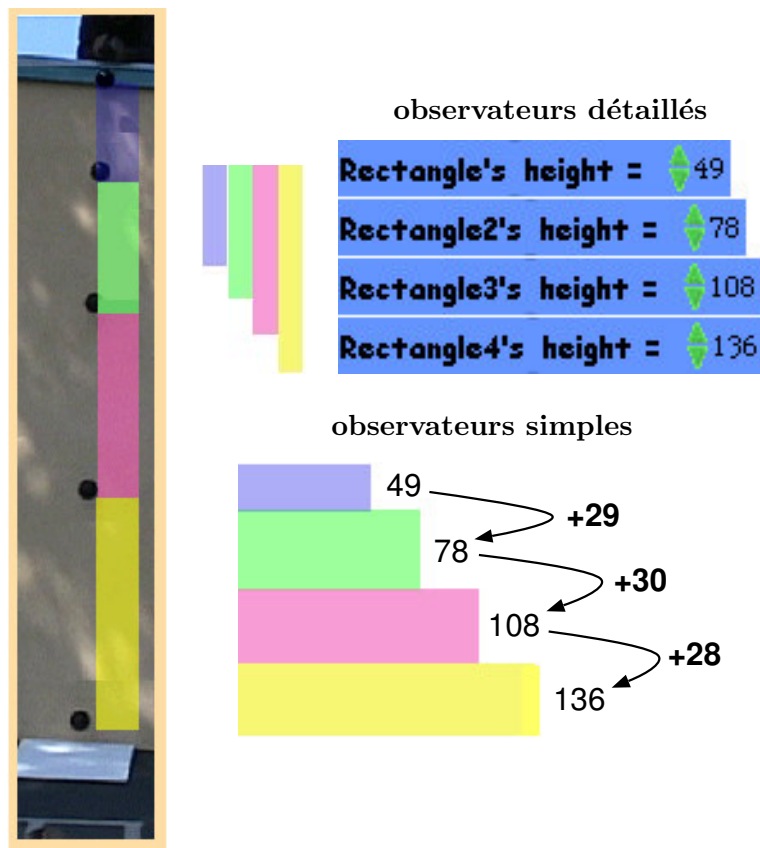


Il est important pour les élèves de comprendre ce qui arrive à la distance entre chaque balle. Nos élèves ont exprimé de diverses manières ce qu'ils ont vu.

Avant de poursuivre, il pourrait être utile de rappeler aux élèves l'ancienne activité de mesure où ils ont appris l'importance de l'approximation.



Certains élèves ont aligné leurs rectangles pour former divers graphiques à barres, afin de visualiser l'accélération. Certains d'entre eux ont choisi d'utiliser les observateurs (*watchers*) pour obtenir la valeur numérique de la hauteur ou de la longueur de chaque rectangle. Le nombre croissant a confirmé le taux de changement de la vitesse de la balle.



Le rappel de la précédente exploration dans les mesures (rappeler aux élèves ce qu'ils ont appris à propos de l'approximation et la tolérance) pourrait les aider à voir que la longueur de chaque barre (mesurée en pixels) n'est pas seulement croissante, mais qu'elle paraît croître presque à un taux constant – autrement dit, une accélération constante.

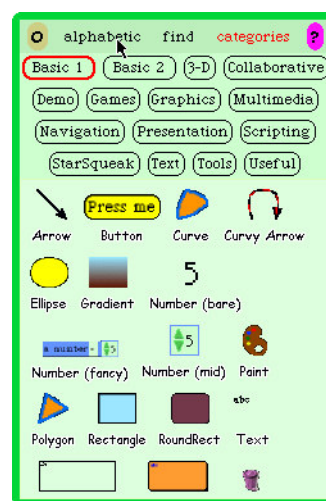
Certains élèves ont choisi d'effectuer un calcul qui devrait leur dire si la balle accélérât en tombant et, le cas échéant, de combien. Ils ont soustrait la hauteur d'un rectangle de la hauteur du rectangle suivant pour déterminer une approximation de l'accélération. A ce point, ils pouvaient voir qu'effectivement, la balle accélérât à un taux constant.

## Trouver la loupe (magnifier)

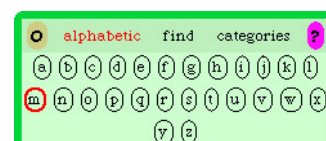
Glisser le catalogue d'objets (*Object Catalog*) depuis le tiroir Supplies.



Choisir **alphabetic** depuis le catalogue.



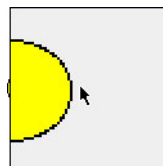
Choisir la lettre **m**.



Choisir la **Magnifier** (loupe) et la glisser dans le projet.



Cet outil va agrandir n'importe quelle partie de l'écran où se situe la souris.



## CREER DES FILMS : UNE EXPLORATION

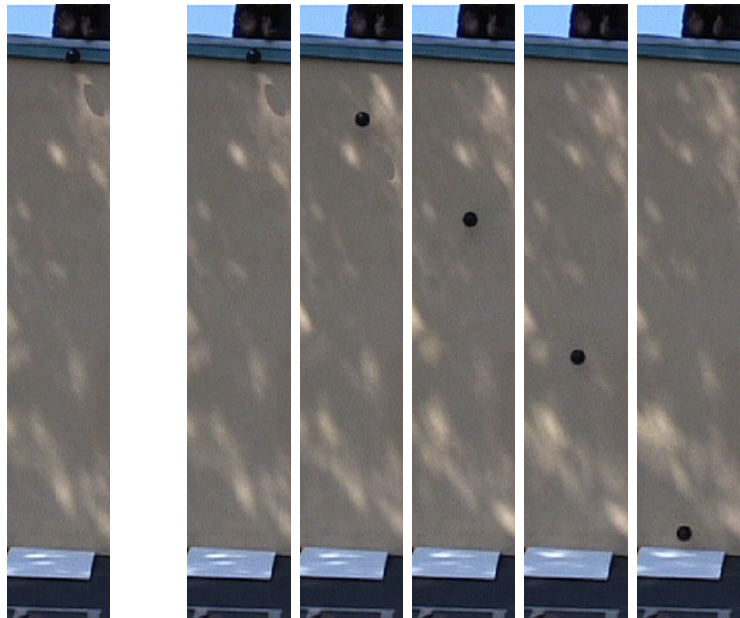
Dans ce projet, une animation sera créée depuis les jpegs de la séquence de lâcher de balle.

Rechargez la séquence jpeg dans un nouveau projet Squeak. Nommez la première vignette de la séquence **dropMovie**. Puis, glissez un **Holder** (conteneur) depuis le tiroir Supplies et redimensionnez-le pour contenir toutes les vignettes de la séquence de lâcher de balle.

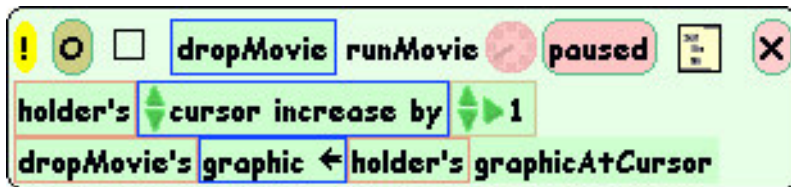
Glissez une copie de la première vignette dans le conteneur (*holder*). Puis, glissez toutes les vignettes restantes directement dans le conteneur.



Nous faisons un film !!



Faites apparaître les briques de script (*scripting tiles*) pour le cadre appelé **dropMovie**. Créez un script. Nommez-le **runMovie**.



Ce script créera un film animé de la balle tombante depuis le toit.

```
runMovie
show code textually
save this version
destroy this script
rename this script
button to fire this script
edit balloon help for this script
explain status alternatives
hand me a tile for self
hand me a "random number" tile
hand me a "button down?" tile
hand me a "button up?" tile
fires per tick...
add parameter
```

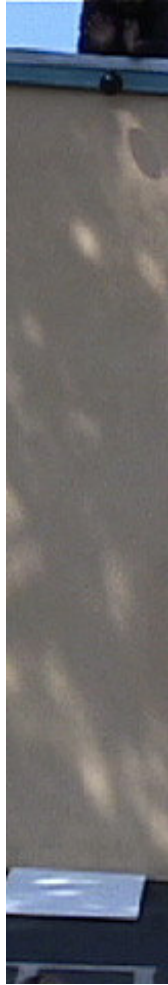
Une fois que vous aurez créé le script, faites un bouton pour l'exécuter. Renommez le bouton en **stepMovie**.

**stepMovie**

Si vous démarrez le script, il s'exécutera de manière continue, mais si vous cliquez sur le bouton, le script ne sera exécuté qu'une seule fois. A chaque clic supplémentaire, le script fera une passe sur la séquence de vignettes et animera la chute de la balle.

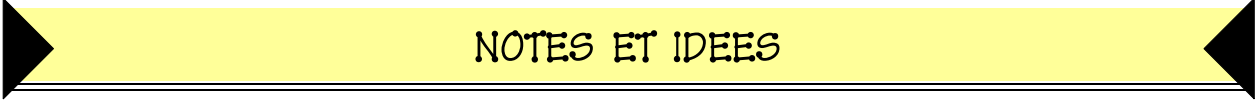
Une fois que le bouton a été réalisé, assurez-vous de bien sauvegarder votre projet. Vous en aurez besoin plus tard.

Nous sommes prêts à passer au projet final : une simulation de la gravité à l'aide d'une balle dessinée. Le défi ultime sera de faire «tomber» une balle dessinée au même taux d'accélération que la balle dans le film. Nous voulons que la balle dessinée et la balle du film atteignent le bas au même moment.



Eh ben, ceci nécessitera une vraie réflexion. Ca, *c'est* un bon défi!



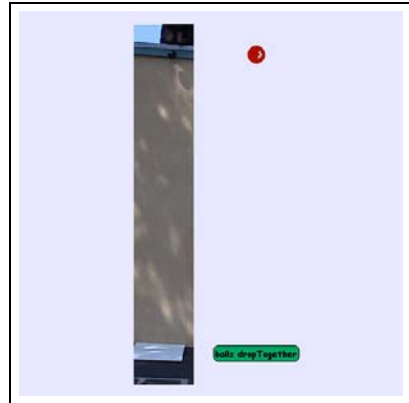


NOTES ET IDEES



## PROJET 12

# UNE SIMULATION DE CHUTE DE BALLE

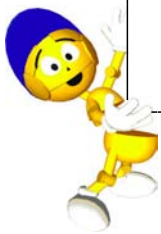


Ceci est le projet ultime de cette série explorant l'accélération et la gravité. Dans ce projet, les élèves seront mis au défi de dessiner une balle et de la munir d'un script qui modélise l'accélération des objets du monde réel du fait de la gravité.

L'animation de la chute de la balle sera ensuite importée dans le projet afin que les élèves puissent comparer la chute de leur balle avec celle de la balle dans la vidéo.

La création de cette simulation nécessite que les élèves pensent à plusieurs concepts utilisés dans les précédents projets — la création de variables, l'utilisation de *augmenter* (*increase by*) et ce que veut dire *augmenter* d'un nombre négatif.

Le plus important est probablement que cette animation encourage les élèves à réfléchir à la création de théories, à les tester pour valider leurs hypothèses à l'égard de ces théories et à réaliser des modèles simulant les phénomènes du monde réel.



## Prérequis Squeak

- Créer et nommer des scripts.
- Créer des variables.
- Créer des scripts de réinitialisation.
- Créer des animations.
- Créer des boutons.
- Créer et utiliser des observateurs.

## Concepts associés Math

- Comprendre les variables.
- Comprendre la puissance de «augmenter de »(*increase by*) avec les nombres négatifs.
- Mouvement selon l'axe y.
- Comprendre le concept d'accélération.

## Concepts associés Science

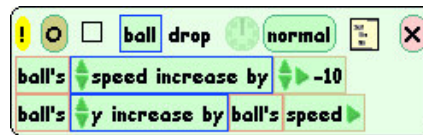
- Poser des hypothèses.
- Voir comment l'accélération devient un facteur clé pour la compréhension de la force exercée par la gravité sur un objet tombant.
- Simuler la force de la gravité sur un objet tombant, en utilisant un modèle comme guide.

Démarrez un nouveau projet Squeak. Dessinez une balle et nommez-la.



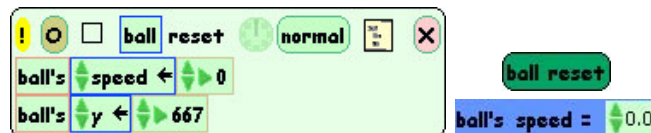
Faites apparaître la brique de script pour la balle. Créez une variable et nommez-la *speed* (vitesse).

Créez un script dans la balle nommée **drop**. Ce script va augmenter *ball's speed* (la vitesse de la balle) d'un nombre négatif, qui reste à déterminer. L'exemple ci-dessous va tester ce qui se produit quand la vitesse est augmentée de  $-10$ . La seconde ligne du script **drop** (*lâcher*) va augmenter *ball's y* (le y de la balle) avec sa vitesse.



A partir de ce point, vous avez un modèle d'une balle tombante à accélération constante !

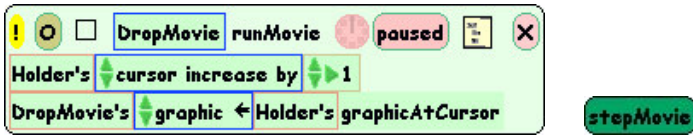
L'étape suivante est de créer un script qui réinitialise *ball's speed* (la vitesse de la balle) et *ball's y* (le y de la balle) à leurs valeurs initiales. Nommez le script **reset**. Créez un bouton qui va lancer ce script. Récupérez un observateur détaillé (*detailed watcher*) pour la variable *speed*. Le script **reset** va simplifier l'initialisation de la position et de la vitesse de la balle.



Expérimentez et jouez avec le nombre duquel la vitesse est augmentée. Que se passe-t-il ? Qu'est-ce qui se passe quand le y de la balle n'a pas augmenté avec sa vitesse ?

Puis, ouvrez le projet dans lequel vous avez sauvegardé le film de la chute de balle. Dessinez une nouvelle balle et placez-la à côté de la balle du cadre **dropMovie**.



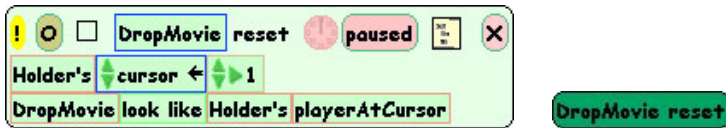


Lancez l'animation en boucle en activant le script en mode «ticking». Un click sur le bouton **stepMovie** fera un pas dans l'animation.

Le dernier défi est de «scripter» la balle dessinée pour qu'elle tombe avec le même taux qu'une balle dans l'animation. En utilisant ce que vous avez précédemment appris, recréez les scripts pour la balle dessinée afin qu'elle tombe et qu'elle se réinitialise. *Note : Lors de la création du bouton **reset** (remise à zéro), souvenez-vous que comme le film commence avec la balle en mouvement, sa vitesse initiale ne peut pas être zéro.*



Puis créez un script *dans le cadre **dropMovie*** qui va réinitialiser l'animation à la première vignette. Nommez le script **reset**. Créez un bouton qui va lancer ce script.



Finalement, créez un script *dans la balle* qui fasse tomber la balle et démarrer le film en même temps. Nommez ce script **dropTogether** (tombe ensemble). Quand vous créez un bouton qui va démarrer ce script, renommez le bouton en **balls drop together**.



Quand le script **dropTogether** est lancé, il sera nécessaire de tester (et retester) et de changer la valeur de la vitesse. balle tombe au même taux que la balle dans l'animation. Dans notre classe, quand les élèves ont réussi, nous avons entendu des cris de joie. Non seulement, ils ont rempli avec succès le défi, mais ils avaient une meilleure compréhension du concept de gravité que celle qu'ils auraient acquise à la lecture d'un livre.

## Défi

- Dans ce livre nous avons utilisé Squeak pour créer à la fois des *animations* et des *simulations*. Pensez à la différence entre eux. Pourquoi n'est-ce pas la même chose?

## Quelques notes finales

Les projets dans ce livre sont juste le début de ce qui pourrait être créé avec ce puissant outil.

Nos solutions sont seulement des suggestions, puisqu'il peut y avoir plusieurs voies pour résoudre ces problèmes intéressants.

Squeak, en tant qu'outil pour l'enseignement des maths et des sciences, possède des possibilités illimitées!



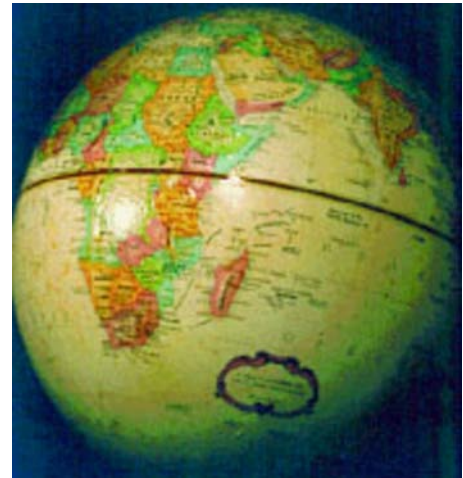
A bientôt sur **Squeakland!!**



### Notre condition humaine «depuis l'Espace»

Alan Kay

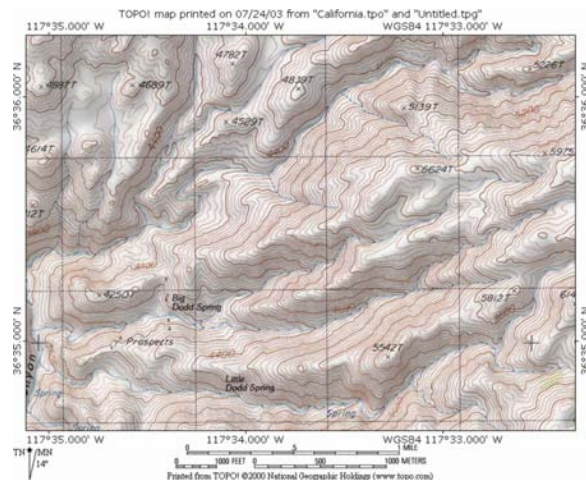
Mes exemples préférés des débuts de la science, et une merveilleuse métaphore de ses mécanismes, sont les tentatives de cartographies précises initiées par les grecs, perdues durant un millénaire et reprises au 15ème siècle. A la fin du 16ème siècle, les gens étaient ravis de pouvoir acheter des globes terrestres de poche «Le monde comme vu de l' Espace». Deux siècles plus tard nous sommes allés dans l' Espace, y avons observé notre monde, pris des clichés, et vus ce que les cartographes du 18ème siècle avaient déjà découvert.



Tous les procédés de la connaissance scientifique ont ce caractère : des tentatives pour «voir» et se représenter le monde très précisément depuis des points de vue privilégiés qui ne font pas partie de notre vécu intuitif - «rendre l'invisible quelque peu visible». Durant la majeure partie de notre histoire, nos théories sur nous-même et notre monde étaient essentiellement composées de croyances fragiles exprimées sous forme d'histoires réconfortantes. Il y a quelques siècles nous avons appris à changer de points de vue pour percevoir le monde physique «comme vu de l'espace» avec bien moins de préjugés sur le chemin. Au 21ème siècle nous avons besoin non seulement de le faire pour notre monde matériel mais aussi pour la compréhension de l'intégralité de la condition humaine «comme vu de l'espace», sans histoires réconfortantes mais avec une meilleure compréhension de notre développement et en harmonie avec la nature.

## EPILOGUE

Les cartes, ainsi que toutes nos représentations d'idées, sont plutôt arbitraires et ne se réclament pas automatiquement exactes. Par exemple, voici 3 cartes. La première nous vient du Moyen Age, la seconde est une carte du livre de Tolkien «Le seigneur des anneaux», et la troisième est une carte du désert Mojave. La carte médiévale montre le monde comme les gens pensaient qu'il devrait être et qui inclut le jardin d'Eden (à l'extrême Est en haut). La méditerranée est au milieu, Jérusalem est au centre du monde et la botte de l'Italie n'est qu'une petite excroissance. La carte de Tolkien a été réalisée avec beaucoup de détails pour aider les lecteurs (et probablement l'auteur) à mieux visualiser le monde imaginaire des hobbits et du seigneur des anneaux. La carte du désert Mojave date d'une année et elle a été conçue à l'aide de techniques sophistiquées de survol et d'imagerie satellite pour guider le positionnement des indications.



Il est important de remarquer que du point de vue de la logique traditionnelle, aucune de ces cartes n'est «vraie», dans le sens où aucune d'entre elles n'établit de correspondance point à point avec tous les détails qu'elles seraient censées représenter. A l'intérieur d'une carte nous pouvons appliquer une logique parfaite : par exemple si Rome est au nord d'Alexandrie, et Paris au nord de Rome, alors Paris est au nord d'Alexandrie. Cette logique interne fonctionne parfaitement pour toutes les cartes. Les mathématiques sont aussi une sorte de carte, conçues pour être totalement consistantes à l'intérieur d'elles-mêmes : en fait les mathématiques incluent l'élaboration de telles cartes ( «Mesurer la terre» en grec se dit «Géométrie»). C'est lorsque nous essayons de nous

rapporter aux cartes pour ce qu'elles sont sensées représenter «à l'extérieur» que l'on se heurte à des difficultés et que l'on découvre qu'aucune n'est «vraie» au sens de la vérité qui peut être obtenue à partir d'une carte. Mais si vous étiez parachuté dans le désert du Mojave, quelle «fausse» carte choisiriez-vous ? Plusieurs saveurs de «faux» font vraiment la différence dans la pensée moderne !

De notre point de vue, la raison d'enseigner «la nouvelle pensée» qui s'est épanouie ces derniers 400 ans n'est pas de fournir d'avantage de métiers techniques, ni «de garder un pays fort», ni même de former de meilleurs citoyens. Ceux-ci sont tous des sous-produits de la nouvelle pensée, mais les vraies raisons concernent le fait d'être sain et civilisé. Si les cartes dans nos têtes sont différentes «de l'extérieur» alors nous sommes au mieux ce que Alfred Korsybski qualifie de «malsain». Notre définition d'être malsain se produit simplement lorsque les cartes dans nos têtes, quelques en soient les raisons, deviennent tellement différentes «de l'extérieur» (incluant les cartes des autres) que c'est remarquable et parfois dangereux. Puisqu'on ne peut pas réaliser de cartes exactement vraies, nous sommes toujours quelque peu malsain en ce qui concerne le vrai monde physique. Comme nos cartes internes ne sont pas directement partageables, nous sommes encore plus malsain par rapport à la cartographie du monde de chaque personne (nous y compris). Parce que nous pensons par rapport à nos cartes internes une sorte de représentation théâtrale de nos propres croyances ce n'est pas trop exagéré de dire que nous ne vivons pas dans la réalité, mais dans le rêve éveillé d'un somnambule que nous aimons appeler «réalité». Nous voulons certainement construire la version «la moins fausse» possible !

La civilisation n'est pas un état d'être, ni le trajet, mais une façon de voyager. Pour moi la chose la plus intéressante et la plus remarquable voire incroyable à propos de la science est qu'elle est faite par nous bien que nous ne soyons que des créatures ayant des histoires de différents types dans nos têtes et que nous soyons d'avantage enclins au combat contre le tigre-sabre que par le changement climatique au travers des siècles. Mais la méthode de la pensée scientifique est capable de prendre compte de nos propres incapacités à penser et autres déviations d'une manière suffisamment efficace pour produire une cartographie de plus en plus précise de phénomènes de plus en plus complexes de notre Univers. C'est pour cette raison que nous devons aider tous les enfants dans le monde à apprendre comment le faire.

Mais pourquoi les sciences et leur langage de cartographie, les mathématiques, sont-elles condamnées à être si difficiles à apprendre ? Je crois que ce n'est pas parce qu'elles sont intrinsèquement si compliquées, mais plutôt parce qu'elles sont incroyablement simples mais cependant très différentes de notre sens commun et de notre façon de réfléchir. Réussir à accéder à ce «point de vue» inhabituel de ce qui est «en dehors», et ce que cela signifie de chercher et d'en déduire des connaissances, est le processus principal de la science. Une façon voir cela est de comprendre qu'une partie de ce qui est à apprendre est un nouveau type de bon sens. Alan Cromer appelle cela le «uncommonsense».

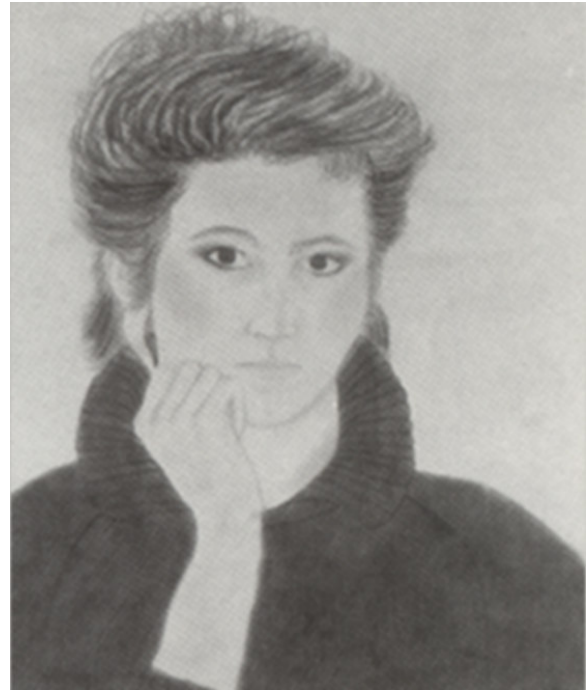
Et, de la même manière que l'on n'a pas besoin de plus qu'un esprit normal pour apprendre ces idées, il n'est pas nécessaire non plus de fonds financiers importants, bien que beaucoup de gens aiment à expliquer que l'enseignement des sciences ne fonctionne pas en raison du manque d'ordinateurs, d'équipements scientifiques, de livres, etc. Les sciences ont aujourd'hui près de 400 ans d'existence, nous avons des ordinateurs personnels depuis un peu plus de 20 ans, il y a ainsi 380 années durant lesquelles les sciences et les mathématiques ont été apprises sans outils high-tech. Certaines des découvertes les plus importantes ont été réalisées avant la révolution industrielle avec très peu d'équipement.

Je crois que ce qui manque principalement ce sont des adultes qui comprennent les sciences et qui veulent travailler avec des enfants et des enseignants malgré les financements. Honte à ma profession ! La plupart d'entre nous reste dans les laboratoires et loin des enfants, parents,

enseignants et écoles.

Comment peut-on apprendre les sciences sans argent ? En premier lieu, nous devons apprendre comment observer et être intéressé par des phénomènes de façon non catégorique, c'est-à-dire sans écarter les choses une fois que nous avons à peine appris leurs noms - la plupart des choses deviennent presque invisibles une fois qu'on les reconnaît et qu'on se rappelle leur nom. Nous devons donc trouver des moyens pour «rendre l'invisible visible», pour éviter «la reconnaissance prématurée». Les sciences sont tout autour de nous et beaucoup peut être révélé en étant simplement plus attentif à ce que nous pensons voir.

Une façon de faire cela est d'apprendre à dessiner. Comme Betty Edwards (dans «Dessiner du côté droit du cerveau») le montre, apprendre à dessiner consiste essentiellement à apprendre à voir (par apposition à apprendre à reconnaître). Nous devons trouver des façons de neutraliser une reconnaissance rapide en faveur d'une perception plus lente. Les portraits typiques «avant et après» des étudiants d' Edwards montrent une amélioration remarquable.



C'est quelque peu différent de la «partie artistique» des arts visuels dans le sens où nous essayons essentiellement d'exprimer des détails visuels de «ce qui est là» plutôt que ce qu'on en ressent (mais ils ne sont pas du tout exclusifs). Comme mon grand-père l'avait remarqué dans un article qu'il écrivait pour le Saturday Evening Post en 1904 au sujet de la photographie en tant que possible forme d'art : «L'art intervient lorsque nous construisons avec attention et avec un objectif en tête ; c'est à dire, lorsque nous nous libérons des actions qui sont essentiellement mécaniques». Nos sentiments apparaissent dans chacune de nos créations réalisées avec attention.

Un autre bon exemple d'«observation précise à bas prix» est la mesure de la circonférence du pneu de vélo de la page 43. Une bonne partie de la quintessence de la philosophie scientifique peut se reconnaître dans cette activité d'observation attentive.

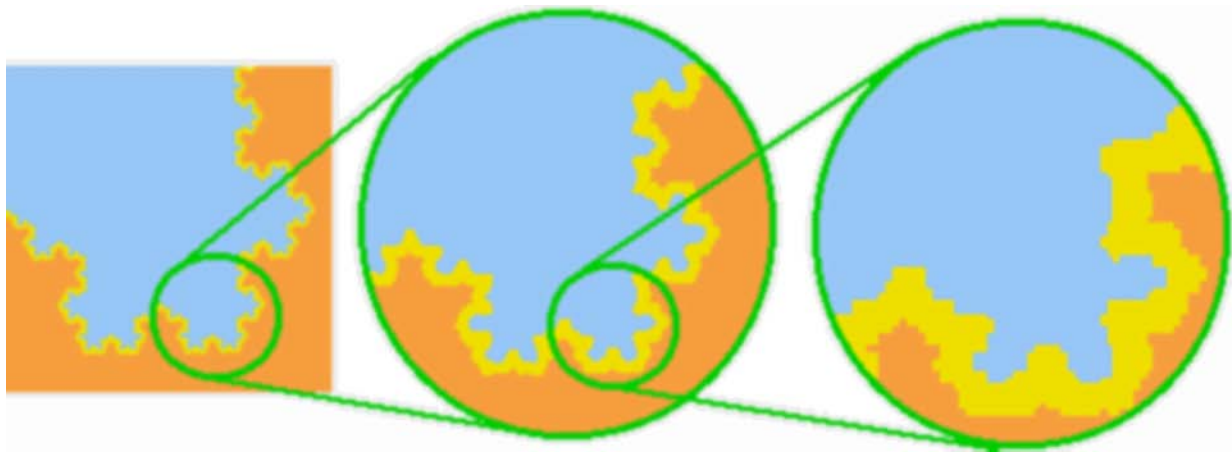
Les étudiants ont utilisé différents matériaux et ont trouvé différentes réponses, mais ils étaient certains qu'il y avait une réponse exacte en centimètres (pour partie parce que l'école les encourage

pour obtenir des réponses exactes plutôt que réelles). Un des enseignants le pensait aussi car sur un côté du pneu on pouvait lire que son diamètre était de 20 pouces. L'enseignant «savait» que la circonférence était  $\text{Pi} \times \text{diamètre}$ , que  $\text{Pi} = 3.14$  et que «des pouces fois 2.54 donnent des centimètres», etc., et multiplier le tout pour obtenir «la circonférence exacte» du pneu = 159.512 cm. J'ai suggéré qu'ils mesurent le diamètre et ils ont trouvés que c'était d'avantage 19.75 pouces (le pneu n'était pas gonflé)! Cela a été un choc, puisque ils étaient tous habitués à croire pratiquement tout ce qui était écrit, et l'idée de faire un test indépendant de quelque chose d'écrit ne les a pas effleurés.

Ceci a mené à se poser la question du gonflage à différentes pressions, etc. Mais malgré tout certains pensaient encore qu'il existait une circonférence parfaite. Un d'entre nous a alors contacté le fabricant du pneu (qui était en fait Coréen) et il y a eu plusieurs échanges par mail intéressants et divertissants jusqu'à ce qu'un ingénieur de cette compagnie soit contacté et nous écrivit «Nous ne connaissons pas en réalité la circonférence ou le diamètre du pneu, nous les extrudons et les coupons à une longueur qui est de 159.6 cm plus ou moins un centimètre de tolérance!».

Ceci a réellement étonné et impressionné les enfants le constructeur du pneu ne connaissait même pas le diamètre ou circonférence! - cela leur a permis de penser de façon beaucoup plus profonde. Peut-être qu'on ne peut pas mesurer les choses exactement. N'y a-t-il pas des «atomes» ci-dessous? Ne s'agit-ils pas? Les atomes ne sont ils pas composés de choses qui vibrent? Et ainsi de suite.

L'analogie avec «quelle est la longueur de la côte?» est intéressante. La réponse est partiellement liée à l'échelle et à la tolérance des mesures. Comme Mandelbrot et d'autres intéressés par les fractales l'ont montrée, la longueur d'une côte mathématique peut être infinie, et la Physique nous montre que la mesure physique pourrait «presque» être aussi longue (c'est à dire très longue).



Il y a plusieurs façon d'utiliser le puissant concept de «tolérance». Par exemple, lorsque les enfants font leurs projets sur la gravité et en tirent un modèle de ce que fait la gravité aux objets près de la surface de la Terre, c'est très important pour eux de réaliser qu'ils peuvent seulement mesurer jusqu'au pixel de leur écran et qu'ils peuvent aussi faire des erreurs. Une mesure littérale stricte peut les empêcher de voir qu'il s'agit d'une accélération uniforme. Ils ont donc besoin d'être tolérants avec de petites erreurs. D'un autre côté, ils ont besoin d'être plutôt vigilant face aux divergences qui résident en dehors des erreurs typiques de mesure. Historiquement, c'était important pour Galilée de ne pas être capable de mesurer vraiment précisément la façon dont les boules roulaient le long du plan incliné, et pour Newton de ne pas connaître l'orbite de la planète Mercure lorsqu'on la regarde réellement de près.

## EPILOGUE

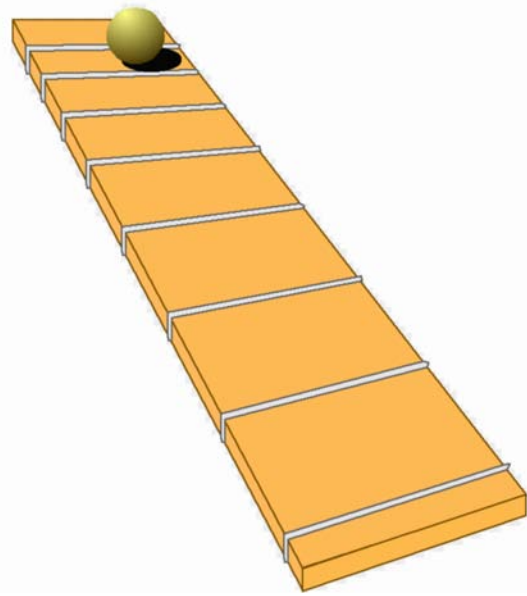
L'année prochaine (2004) sera le 400ème anniversaire où pour la première fois dans l'histoire un bon modèle a été créé pour décrire la chute des corps proches de la surface terrestre sous l'influence de la gravité. Galilée n'avait pas de caméscope et d'ordinateur avec squeak pour construire un modèle. Il a fait sa découverte «sans argent» en étant très appliqué dans ses observations jusqu'à qu'il trouva une façon d'épingler fermement toutes ses observations dans une carte grâce aux mathématiques.

Comment a-t-il fait ? Il ne semble pas qu'il y ait de réponse définitive mais il y a plusieurs histoires qui proviennent des notes et écrits de Galilée. Le père de Galilée était un musicien professionnel et Galilée avait la réputation d'être un excellent amateur concernant plusieurs instruments dont la flute et le luth.

Il a fait de nombreuses expériences avec des plans inclinés avec des boules de tailles identiques fabriquées dans différents matériaux et ayant différents poids. Il découvra que les boules de même tailles mais de poids différents semblaient rouler le long du plan incliné avec le même taux d'augmentation de la vitesse quelque soit l'angle du plan.

Un jour il a peut-être paresseusement laissé rouler une balle ou deux depuis le manche de son luth. Vous pouvez voir que les touches du luth et de la guitare ne sont pas espacées de façon régulière. A un certain moment il s'est peut-être rendu compte que les claquements de la balle sur les touches étaient presque réguliers et a réalisé que les écartements grandissants entre les touches étaient compensés par l'augmentation de la vitesse de la balle. Une intéressante caractéristique des luths est que contrairement aux guitares leurs touches sont constituées du même boyau que celui qui est utilisé pour les cordes et sont simplement ficelées. Ainsi Galilée pouvait les bouger. Il commença à les bouger jusqu'à qu'il puisse entendre des claquements parfaitement réguliers (il a probablement fixé des boyaux identiques tout au long de son plan incliné). Lorsqu'il obtint des claquements parfaitement réguliers, il mesura les distances et trouva que l'augmentation de la vitesse (l'accélération) était constante !





L'un des enseignements les plus importants ici est qu'il y a de nombreux moyens d'investigation intéressants qui peuvent être réalisés avec des matériaux quotidiens si l'enseignant comprend vraiment les sciences. C'est une façon de faire cette investigation «sans argent», et laisser rouler un camion-jouet le long du plan incliné en y fixant une cartouche d'encre qui goutte en est une autre.

Ne laissez pas le manque d'ordinateurs ou d'équipements vous ralentir. Les sciences et les maths sont tout autour de nous. Le monde dans lequel nous vivons est un vaste laboratoire rempli d'équipements s'il peut être remarqué. Il y a des bibliothèques gratuites et publiques même dans les recoins les plus reculés des USA qui contiennent des livres qui expliquent tout cela : la connaissance ne coûte pas d'argent mais du temps, de l'intérêt et de la concentration.

Vous lisez ce livre car vous êtes intéressés par ces problèmes – peut-être vous l'avez trouvé dans une librairie publique – que vous puissiez ou non vous offrir un ordinateur aujourd'hui. Si vous ne pouvez pas, il y a cependant encore beaucoup de choses que vous pouvez faire, comme il y a tellement de vraie musique qui peut être faite sans instrument formel. Si vous pouvez vous offrir l'instrument – musique ou ordinateur – alors vous avez un magnifique amplificateur de vos influx musicaux, mathématique ou scientifique.

L'ordinateur assez naturellement transforme les mathématiques en un phénomène, permettant ainsi de compléter le cercle et de remettre ensemble ce qui a été décomposé par la nature même de la science. C'est un des usages les plus importants en science et ingénierie des adultes et ainsi les enfants et les adultes se rejoignent fortement dans le même art et sport, juste comme la musique des enfants est de la vraie musique et le baseball ou le tennis pratiqué par les enfants sont de vraies versions de ces sports.

Une autre considération à prendre en compte est que l'éventail et la qualité des réalisations que

les enfants peuvent entreprendre est largement facilité par l'utilisation d'un ordinateur adéquat. Plusieurs chercheurs ont montré que les enfants ont des pensées bien plus riches que ce qu'ils peuvent facilement construire classiquement : par exemple, ils peuvent précisément réfléchir à la façon dont des robots ou des animaux peuvent trouver leur chemin et créer des programmes réellement élaborés qui donnent vie à des idées qui dépassent largement leurs capacités à faire la même chose avec des modèles physiques.

Dans quelques années les ordinateurs eux-même seront presque gratuits et feront réellement partie d'un réseau global de communication. Ainsi, toutes les idées décrites dans ce livre sont presque à portée de tous les enfants de la planète. Mais il faudra toujours trouver des façons de se rappeler ce qui est important.

Il est capital de garder en tête la distinction entre «faire de la vraie science» et «apprendre ce que les scientifiques ont découvert». Cette distinction est similaire à celle de la «musique» et de «la compréhension de la musique». La dernière est utile dans les deux cas, mais les deux nécessitent l'apprentissage du fonctionnement réel pour comprendre ce que «comprendre la connaissance» signifie vraiment. Par exemple, il n'y a pas de différences entre recevoir un «livre religieux» rempli d'affirmations et nous demander d'y croire et de les mémoriser, et, recevoir un «livre scientifique» rempli d'affirmations et nous demander d'y croire et de les mémoriser. Comme avec la différence entre deux valeurs logiques (vraie et fausse) et les différentes valeurs logiques de la science la différence entre ce que signifie la science lorsqu'elle dit «nous savons que ...» et ce qu'un système de connaissances précédent signifie par cela ne pourrait être plus grand. Quand les sciences affirment «connaître», c'est si différent des usages précédents qu'ils n'auraient pas dû réutiliser le mot «savoir» pour cela, parce que cela signifie est : «nous avons une très bonne cartographie du problème qui fonctionne bien avec cette tolérance mais qui ne cartographie pas aussi bien qu'on le voudrait ici et là, et au fait, voici comment vous pouvez nous aider à vérifier ceci et donner vos propres critiques, etc.»

J'espère que les projets présentés dans ce livre et ce que vous avez déjà lu jusqu'à présent vous convaincront que ces activités ne sont pas seulement «des maths et des sciences», mais des aspects réels et importants des mathématiques et des sciences. Qu'en serait-il si davantage de problèmes étaient pensés ainsi plus lentement, avec une perception réfléchie, sceptique, attentive et de façon cartographique ? Si vous pensez que les choses seraient largement différentes et améliorées pour le bénéfice de tous, alors s'il vous plait aidez les enfants à penser bien mieux que la plupart des adultes ne le font aujourd'hui.



Can you speak "Squeak?"



## La flèche d'affectation

La flèche d'affectation est la flèche verte sur fond violet que l'on trouve dans le panneau de contrôle. Quand on utilise cette flèche dans un script, elle affecte une valeur à une propriété spécifique d'un objet.



## Etoy

Un eToy est un projet électronique qui peut être une simulation, un modèle ou un jeu créé par un enfant ou adulte. Un eToy aide à illustrer un concept ou à donner un aperçu dans une idée puissante.



## Un Tiroir

Les commandes pour la navigation, les objets prédéfinis et les briques de script sont tous contenus dans des tiroirs. Les tiroirs sont partiellement transparents et peuvent être utilisés pour garder sous la main de nouveaux objets. Le plug-in Squeak a deux tiroirs en plus : Navigator et Objets prédéfinis.



## Les nuages

Quand on sélectionne un objet (Alt-Click ou Command-Click), un nuage d'icônes apparaît.

## Les icônes

Une icône est une petite image colorée qui entoure un objet. Chacune de ces icônes permet d'exécuter des actions sur l'objet. Une présentation détaillée des icônes est disponible sur la page [www.squeakland.org/etoys/tutorials](http://www.squeakland.org/etoys/tutorials). Chaque icône a aussi un ballon d'aide décrivant sa fonction.



## Navigator

Le tiroir orange situé en bas à gauche du monde. Ce tiroir contient des options pour naviguer dans Squeak, publier (sauver), peindre et créer des nouveaux projets.



## Objet

Un objet est l'unité de création en Squeak. En Squeak tout est un objet. On communique avec les objets en leur envoyant des messages (commandes ou scripts).

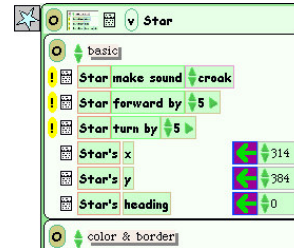
## Le catalogue d'objets

Le catalogue d'objets peut être trouvé dans le tiroir Objets prédéfinis. Il s'agit d'un outil pour trouver des objets. Il contient plusieurs catégories et offre aussi une liste de tous les objets qu'il contient.



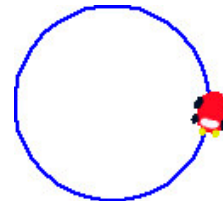
## Pane

Un pane est une partie ou une catégorie d'une panneau de contrôle. Lorsque l'on clique sur le mini-viewer on peut choisir d'avoir une vue ou quatre. On peut détruire un panneau de contrôle en cliquant sur le bouton **remove**.



## Les stylos

Comme en Logo, n'importe quel objet peut écrire à l'aide d'un stylo sur l'écran et y laisser une trace. Les stylos se trouvent dans leur propre catégorie nommée **Pen Use** dans un panneau de contrôle. C'est en changeant la valeur de **pen down** à "true" que l'objet peut laisser une trace. Attention : Les stylos écrivent sur le monde et ne fonctionnent pas sur un fond d'écran paint.



## Une aire de jeux

Une aire de jeux est un objet que l'on trouve dans le tiroir Objets prédéfinis. Une aire de jeux peut servir pour créer un sous-projet ou illustrer un projet. Plusieurs aires de jeux peuvent être placées sur le monde et faire partie d'un projet.



## Plug-In

Un plug-in est un logiciel utilisé pour ajouter des fonctionnalités à un système existant lors de son exécution. Une fois que le plug-in Squeak est installé sur votre ordinateur, les projets deviennent complètement interactifs. Les plug-ins n'ont pas besoin d'être chargés ou déchargés après utilisation. Ils s'exécutent d'eux-mêmes. Il existe plusieurs plug-ins qui offrent différentes fonctionnalités comme par exemple : des lecteurs pour QuickTime, Shockwave et Flash...



## Un projet

Un projet est un hyper document en Squeak. On peut créer, publier (sauver), partager et échanger des projets. Avec un éditeur de texte, nous créons des documents, en Squeak on crée des projets.



## PUBLISH IT!

### Publier

Sauver en Squeak s'appelle publier un projet. Un projet peut être publié sur votre disque dur ou sur un serveur pouvant ainsi être partagé avec d'autres utilisateurs. Pour obtenir les différentes options pour publier un projet, appuyez sur le bouton **Publish It!** du tiroir doré nommé Navigator.



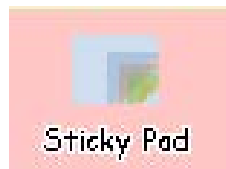
### Script/Scriptor

Les objets peuvent recevoir des messages et exécuter des instructions en combinant des briques à l'aide d'un script. Avant que des scripts soient créés pour un objet, la catégorie "Scripts" dans le panneau de contrôle contient seulement un script vide. Notez qu'il est de bon usage de nommer vos scripts quand vous les créez. Les scripts peuvent être cachés ou détruits en cliquant sur le cercle à droite du point d'exclamation.



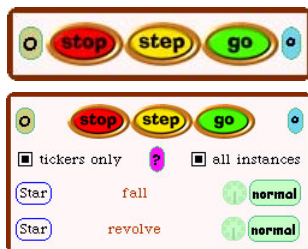
### Un dessin

N'importe quel objet que vous avez peint est un dessin avant que vous lui donniez un nom. C'est une bonne pratique de donner un nom à vos dessins une fois que vous les avez dessinés.



### Les notes collantes

Une note collante est une note de couleur pastel et transparente. Une note peut être décollée du paquet et utilisée dans un projet pour écrire des remarques ou d'autres utilisations. Certains projets de ce livre les utilisent comme règles pour mesurer des distances.



### Les boutons Stop-Step-Go (All Scripts tool)

L'ensemble de boutons **stop step go** que l'on trouve dans le tiroir Objets prédéfinis exécutent et arrêtent tous les scripts d'un projet. Si vous cliquez sur le petit bouton bleu situé à côté du bouton **go**, les boutons **stop step go** s'agrandissent afin de montrer tous les scripts du projet. Cet outil est intéressant pour comprendre comment les scripts ont été construits.

## Les oreilles

Les oreilles sont utilisées pour ouvrir et fermer les tiroirs comment pour les panneaux de contrôle les tiroirs Objets prédéfinis et Navigator. Les oreilles sont créés une fois qu'un dessin est "gardé" et son panneau de contrôle est créé en cliquant sur l'icône turquoise «eyeball». Une oreille a une petite représentation du dessin auquel il est associé. Une oreille peut être cachée (mais non perdu) en cliquant sur le petit cercle situé en haut du panneau de contrôle juste à droite de l'oreille. Lorsqu'une oreille est cachée, il suffit de cliquer sur l'icône turquoise «eyeball».



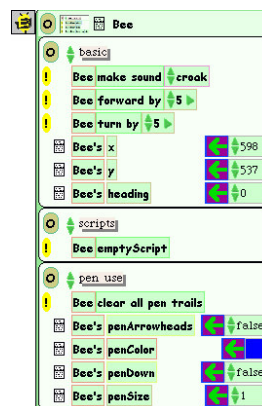
## Les briques

Les briques sont les éléments de base utilisés pour construire des messages et scripts qui sont envoyés aux objets. Les briques se trouvent principalement dans un panneau de contrôle. Mises bout à bout des briques forment une "chaîne".



## Un panneau de contrôle

Le panneau de contrôle d'un objet apparaît quand on clique sur l'icône turquoise «eyeball» de cet objet. Un panneau de contrôle montre les catégories groupant les propriétés et instructions d'un objet sous forme de briques. Il est possible d'exécuter une instruction en cliquant sur le point d'exclamation jaune qui se trouve à gauche d'une instruction (l'exécution est répétée si l'on reste appuyé sur le point d'exclamation). Les valeurs des propriétés comme x, y et heading sont aussi montrées dans un panneau de contrôle. Un panneau de contrôle contient plusieurs catégories. On peut changer de catégorie en cliquant sur les triangles verts. Il est possible d'ajouter des panes à un panneau de contrôle en cliquant sur l'icône "mini-viewer". Nous vous suggérons d'explorer les différentes possibilités.



## Les indicateurs (ou observateurs)

On obtient des indicateurs (simples et détaillés) en cliquant dans le petit menu à gauche d'une propriété dans un panneau de contrôle. Un indicateur peut être utilisé dans un projet afin de suivre l'évolution d'une propriété. La figure montre un indicateur détaillé. Un indicateur simple montre uniquement des valeurs numériques.



### Un monde

Un monde est la base des projets en Squeak. Les simulations sont directement exécutées dans un monde. Si vous n'aimez pas la couleur grise par défaut d'un monde, vous pouvez la changer en utilisant l'icône grise «repaint» du monde.



Cette section est une petite sélection de livres écrits par des penseurs «d'idées puissantes» dont les noms ont été mentionnés dans ce livre ou dont le travail a influencé de manière significative le développement de ce cours.

Jerome S. Bruner. *Toward a Theory of Instruction*. Harvard University Press, 1974.

Alan Cromer. *Uncommon Sense : The Heretical Nature of Science*. Oxford University Press, New York, 1993.

Stillman Drake. *Galileo at Work : His Scientific Biography*. University of Chicago Press, 1981.

Yasmin Kafai and Mitchel Resnick (editors). *Constructionism in Practice : Designing, Thinking and Learning in a Digital World*. Lawrence Erlbaum Associates, 1996.

Alan C. Kay. *A Personal Computer for Children of All Ages*. Xerox Palo Alto Research Center, August 1972.

Alan C. Kay. *Computers, Networks and Education*. Scientific American, September 1991.

[http://www.squeakland.org/school/HTML/sci\\_amer\\_article/sci\\_amer\\_01.html](http://www.squeakland.org/school/HTML/sci_amer_article/sci_amer_01.html)

Maria Montessori. *The Secret of Childhood*. Ballantine Books, 1992.

Philip and Phylis Morrison. *The Ring of Truth : an inquiry into how we know what we know*. Random House, Inc., 1987. ISBN 0-394-55663-1.

Pat Murphy, Ellen Klages and Linda Shore. *The Science Explorer : An Exploratorium-at-Home Book*. The Exploratorium, San Francisco, CA, 2003.

Seymour Papert. *Mindstorms : Children, Computers and Powerful Ideas*. Basic Books, New York, 1980.

Seymour Papert. *Situating Constuctionism*. In I. Harel and S. Papert (editors). *Constructionism*. Ablex Publishing, Norwood, New Jersey, 1991.

Jean Piaget and Barbel Inhelder. *The Psychology of the Child*. Basic Books, New York, 2000.

Jean Piaget. *The Child's Conception of Number*. Norton Press, New York, 1965.

Mitchel Resnick. *Xylophones, Hamsters and Fireworks : The Role of Diversity in Constructionist Activities*. In I. Harel and S. Papert (editors). *Constructionism*. Ablex Publishing, Norwood, New Jersey, 1991.

Mitchel Resnick. *Turtles, Termites and Traffic Jams : Explorations in Massively Parallel Micro-worlds*. MIT Press, Cambridge, Massachusetts, 1994.

R. P. Taylor (editor). *The Computer in the School : Tutor, Tool, Tutee*. Teachers College Press, New York, 1980.

Si vous aimez un traitement plus technique des ces idées les ouvrages suivants constituent un bon point de départ.

H. Abelson and A. diSessa. *Turtle Geomety : The Computer as a Medium for Exploring Mathematics*. MIT Press, Cambridge, Massachusetts, 1980.

Michael Travers. *Agar : An Animal Construction Kit*. Master's Thesis, MIT, 1988.

<http://xenia.media.mit.edu/~mt/agar/agar.html>



## REMERCIEMENTS

Les auteurs souhaitent remercier notre ami et collègue Peter Maguire. Peter est le talentueux artiste qui a donné vie à Adam Link, le personnage qui est apparu tout au long des pages de ce livre et des aventures Etoy. Peter pris une extrême attention à ce que chacun des graphiques que nous avons choisi apparaisse précis et propre dans sa forme finale.

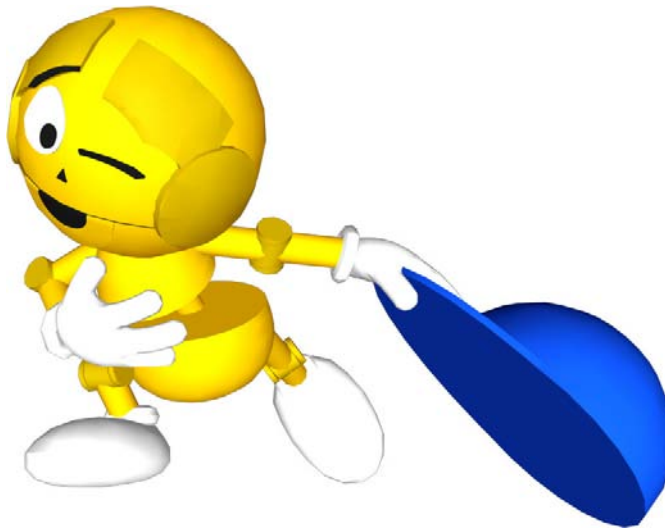
Nous voulons aussi remercier Ian Piumarta, notre éditeur et le cerveau derrière la dernière édition et mise en page de ce livre. Nous sommes reconnaissantes à Ian pour ces efforts lors de cette tâche difficile et laborieuse.

Finalement nous voulons remercier nos collègues — les enseignants et amis qui ont relu et testé les projets de ce livre, pour leurs commentaires et encouragements.

BJ & Kim

Ce livre a été traduit bénévolement par Stéphane Ducasse, Alexandre Bergel, François Schnell, Noury Bouraqadi et Serge Stinckwich. Des parties de cette traduction ont été relues par Florence Ducasse, Clément Ducasse, Hilaire Fernandes, Isabelle Rondel, Rémi Sussan...

Vous avez entre les mains la version 1.2 de la traduction n'hésitez pas à nous contacter pour des améliorations.







BJ Conn est enseignante à Los Angeles Unified School District depuis 36 ans et elle enseigne actuellement à l'Open Charter School. Cela fait vingt ans qu'elle travaille à la création de moyens pour l'intégration d'outils technologiques dans les parcours scolaires et spécifiquement en mathématiques et sciences. Elle a reçu de nombreux prix et bourses pour son travail. Elle a été nommée enseignante de l'année 1988. Elle a été contactée pour partager son expérience et succès dans le cadre de conférences et d'ateliers à travers les Etats-Unis.

BJ a eu le plaisir de travailler avec Alan Kay depuis 1985, lorsque son école (Open Charter School) a été choisie comme site de recherche pour le projet Vivarium mené par Apple.

BJ a obtenu son Bachelor of Arts Degree de California State College de Los Angeles en 1968. Elle a travaillé comme enseignante consultante pour Apple Computer et Walt Disney Imagineering Research & Development. Elle est membre de l'académie internationale des arts digitaux et sciences.



Kim Rose est la co-créatrice et directrice de Viewpoints Research Institute, Inc., une organisation à buts non lucratifs servant le développement de Squeak et d'autres environnements offrant de nouveaux médias. Elle est aussi Systems Program Manager chez Hewlett-Packard. Kim est un critique de média et travaille en science cognitive. Elle travaille avec Alan Kay et son équipe depuis 1986.

Kim travaille avec les enfants et les enseignants de nombreuses écoles, universités et centres d'apprentissages dans le monde entier qui développent et expérimentent avec Squeak et autres environnements basés sur Squeak. Elle explore comment des idées puissantes peuvent devenir plus accessibles aux enfants.

Kim a obtenu un Bachelor of Arts Degree de l'University of California, Los Angeles (UCLA) en 1979. En 1995 elle a reçu une bourse d'étude du département NYU's Media Ecology. En 2000, Prentice Hall a publié un livre édité par Kim et Mark Guzdial : "Squeak Open Personal Computing and Multimedia."

Kim connaît BJ Conn depuis 17 ans et a passé de nombreuses heures en collaborant avec BJ.