

Portfolio – Aurélien Defossez

Ce document vise à montrer brièvement les projets personnels les plus importants et les plus intéressants que j'ai eu l'occasion de faire. Si vous désirez plus d'information, n'hésitez pas à me contacter à cette adresse : defossez.aurelien@gmail.com.

<u>JEUX « GAME JAM »</u>	2
EAT’EM ALL!	2
MONORAIL CAT	3
 COD’OPEN WORLD	 4
LA PLATEFORME COW	4
PHEROWARS	5
FRUIT SALAD	6
 ABELIA WARS	 7
 JEUX DIVERS	 9
AVALIA	9
BUNBYX	10
KURU	11
 DIVERS	 12
WIIPROJECT	12
MAPPING	13

Jeux « Game Jam »

Eat'em All!

Eat'em All est un jeu développé en 24h (étalées sur deux jours) lors de la deuxième Game Dev Party (<http://gamedevparty.fr/>) en 2012.

Dans ce jeu, chaque joueur contrôle une poupée vaudou qui va diriger des zombies dans le but de détruire la forteresse de l'autre joueur. Les zombies sortent du cimetière vaudou à intervalle de temps régulier et la poupée vaudou va placer des panneaux sur le chemin de ces zombies pour les rediriger. Entre les deux forteresses, il y a des villes de différentes tailles, que les zombies peuvent prendre d'assaut pour y trouver de la nourriture (des humains), afin de se reproduire et d'augmenter la taille de son armée. Le premier joueur qui détruit la forteresse ennemie a gagné.

Pour ce jeu, j'ai adossé le rôle de game designer et de chef de projet à temps plein lors de la première moitié de l'événement, puis je me suis mis au développement pour accélérer la création du jeu pour la suite. Après la Game Dev Party, nous avons continué le développement du jeu, afin de le rendre agréable à jouer, équilibré, et plus performant.

À l'heure actuelle, il est toujours en développement. Il est tout de même possible d'y jouer mais quelques problèmes de performances subsistent (notamment sous Firefox Linux et Mac OS) et empêchent d'apprécier le jeu à sa juste valeur. Je vous conseille de le tester sous Chrome, sur lequel le jeu tourne bien. Il est possible que le jeu soit long à se charger car c'est une version de dev. Et n'oubliez pas de mettre le son ! <http://aurelien-defossez.github.com/Eat-them-All/src/index.html>

Vous pouvez aussi visionner la vidéo de démonstration qui a été faite à la fin de la Game Dev Party (Lien dans cet article : <http://gamedevparty.fr/gdp2-eat-them-all/>).

Repository GIT : <https://github.com/aurelien-defossez/Eat-them-All>



Monorail Cat

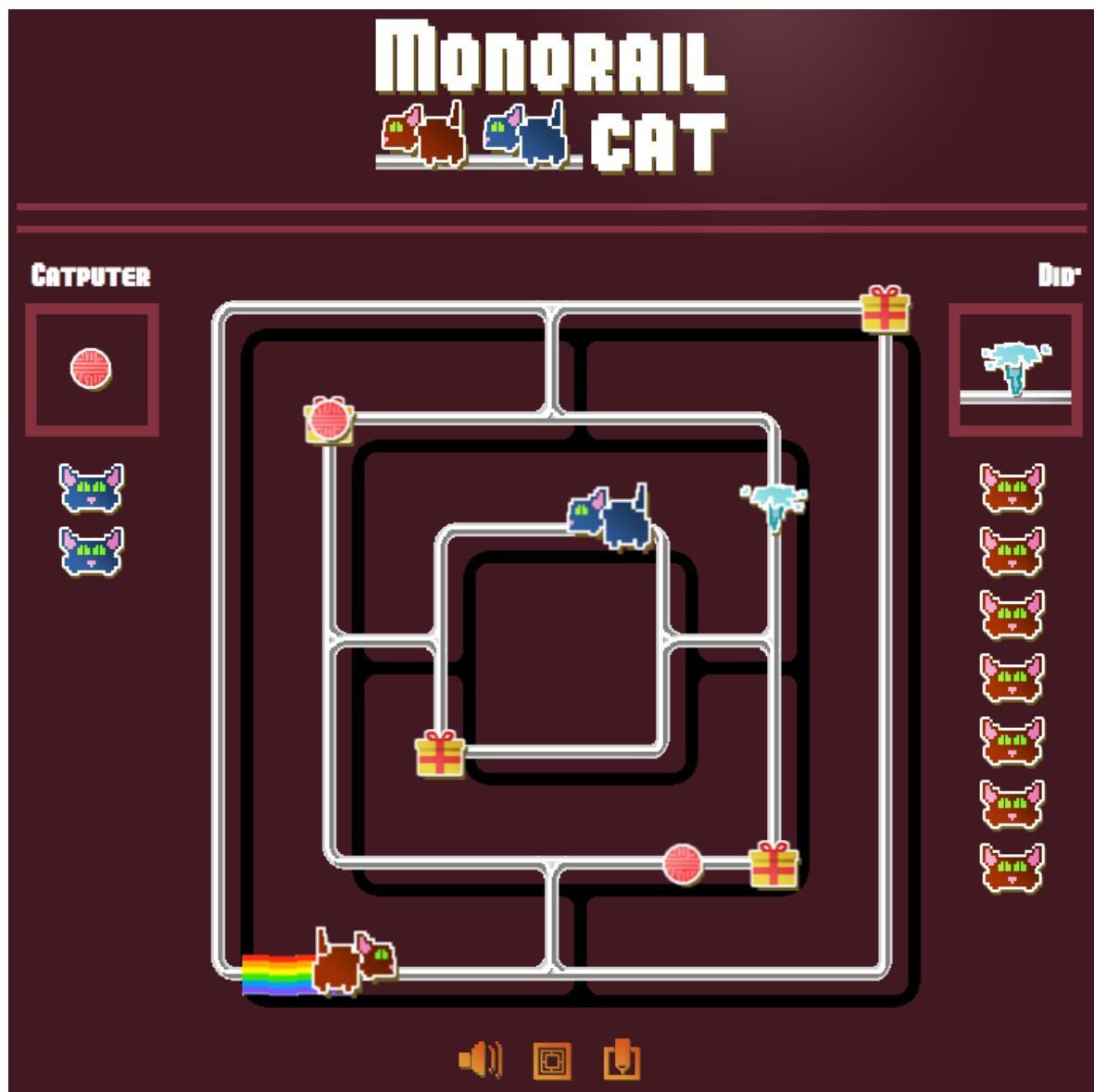
Monorail Cat est un autre jeu développé en 24h lors de l'événement Game Dev Party, pour l'édition 2011.

Dans ce jeu, on contrôle un chat sur rail qui peut ramasser des objets et les envoyer sur son adversaire à la manière d'un Mario Kart. Trois objets ont été développés pendant l'événement, un pic à glace permettant de percer le monorail et créer un simili de mine, une pelote de laine se baladant aléatoirement sur la carte, et un arc-en-ciel, permettant de se transformer en Nyan Cat et d'être invincible pendant quelques secondes.

Pour ce jeu, j'ai endossé le rôle de game designer, chef de projet et co-développeur du jeu. Après la Game Dev Party, certains d'entre nous ont continué de développer le jeu, pour améliorer notamment la réactivité des contrôles et les graphismes.

Le jeu est jouable à cette adresse : <http://monorail-cat.esion99.info/>

Repository GIT : <https://github.com/esion/Monorail-Cat>



Cod'Open World

La plateforme COW

Cod'Open World, abrégé COW, est le nom de projet d'une plateforme Java permettant de faire s'affronter des intelligences artificielles autour d'un jeu. Le but premier de cette plateforme est d'être utilisée par l'association Cod'INSA, dont je fus le chef de projet, afin de faire tourner la finale, réunissant 5 équipes de 5 personnes qui ont pour mission de développer une IA en moins de 24h autour d'un jeu développé pour l'occasion.

Cette plateforme a pour deuxième but d'être exportée sur internet et de permettre à quiconque de développer des IAs, voire des jeux. Pour le moment, les langages Java, Python et C++ sont utilisables pour développer des IAs et des jeux, mais il est prévu de supporter autant de langages que possible (C#, ADA, PHP, Lua, Befunge...).

J'ai entièrement développé cette plateforme. C'est un projet qui me tient beaucoup à cœur, car la compétition d'IAs possède beaucoup de problématiques (par exemple lier des IAs C++ dans une JVM java) qu'il a fallu résoudre et de contraintes (notamment de performances) qu'il a fallu gérer, et c'est très gratifiant de voir qu'aujourd'hui elle marche bien et qu'elle est toujours utilisée par l'association.

Le gros défi technique de COW est son aspect cross-langage. En effet, pour des raisons de performance lors du développement local des IAs, ces dernières sont dynamiquement chargées dans le processus local et doivent donc communiquer directement avec Java sans passer par un moyen de communication interprocessus comme les pipes ou les sockets. Par conséquent, il faut étudier les spécificités de chaque langage, afin de bien l'intégrer avec Java.

Cette plateforme permet de développer des concepts de jeux très novateurs, car le fait que le joueur soit une intelligence artificielle et non un joueur humain ouvre de grandes portes au niveau gameplay.

Repository Mercurial : <http://code.google.com/p/cod-open-world/>

Voici les jeux des finales de Cod'INSA des deux dernières années :

Pherowars

Pherowars est le jeu pour lequel les participants de l'édition Cod'INSA 2010 ont développé leur IA.

C'est un jeu multi-agent dans lequel chaque entité possède sa propre « instance » d'intelligence artificielle. Il se déroule sur une planète peuplée d'insectes, et le joueur possède un nid qui peut créer des fourmis, chaque nid et chaque fourmi possédant sa propre intelligence indépendante.

Le but est de se développer et de survivre, en trouvant des sources de nourriture et en ramenant cette nourriture au nid, tout en se protégeant des prédateurs et des autres joueurs. La difficulté est que les fourmis n'ont conscience que de leur environnement proche (10cm autour d'elles), et ne peuvent mémoriser que de minuscules informations (un « pointeur » vers un autre objet et un entier 16 bits). Malgré tout, elles doivent communiquer entre elles, en utilisant des phéromones, afin de tracer des chemins et divulguer des informations.

Pour ce jeu, j'étais le game designer et gameplay programmer. Il n'était pas sur la plateforme COW mais sur son précurseur, une plateforme en ADA, qui n'était pas assez robuste et générique.

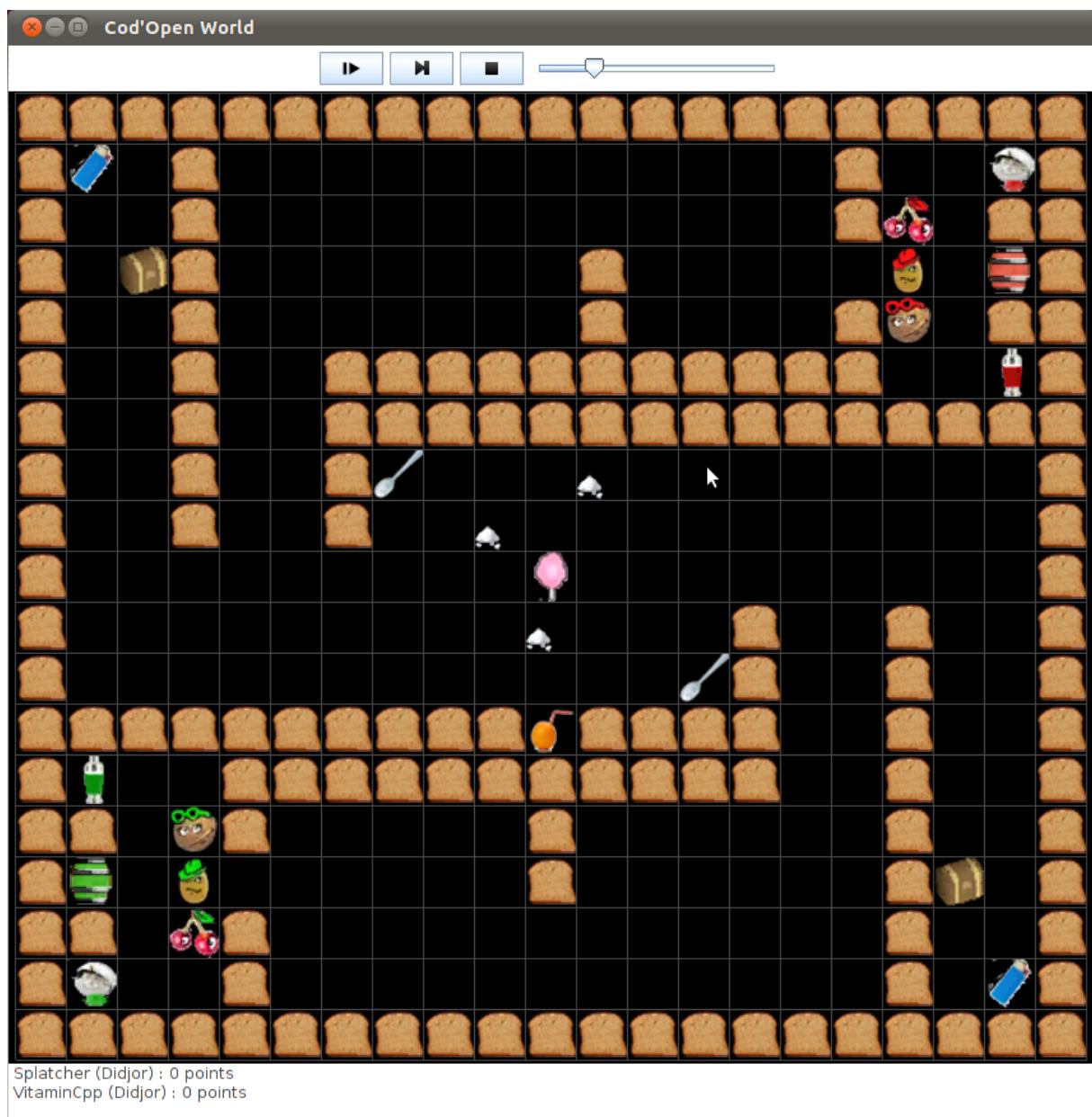
Ce jeu fut très bien accueilli et plusieurs équipes ont réclamé le code source pour pouvoir continuer à développer leur IA.

Fruit Salad

Fruit Salad est le jeu pour Intelligences Artificielles de l'édition 2011 de Cod'INSA.

C'est un jeu se rapprochant plus du jeu de stratégie en temps-réel, dans lequel le joueur contrôle une escouade de fruits qui doivent récolter du sucre et des vitamines pour gagner. Évidemment, ce sont des ressources critiques et les joueurs se battront pour les obtenir. La particularité de ce jeu est qu'il est beaucoup plus axé stratégie, car l'IA unique connaît tout et doit faire effectuer deux actions à ses fruits chaque tour. De plus, les fruits peuvent acheter et revendre des équipements avec le sucre récolté, ce qui ajoute un autre aspect stratégique fort.

Ce jeu utilise la plateforme COW pour communiquer avec les IAs, mais aussi pour visualiser l'état du jeu, en utilisant la vue générique inclue dans COW. Pour ce jeu, j'ai surtout travaillé sur la plateforme, mais j'ai également beaucoup participé au game design.



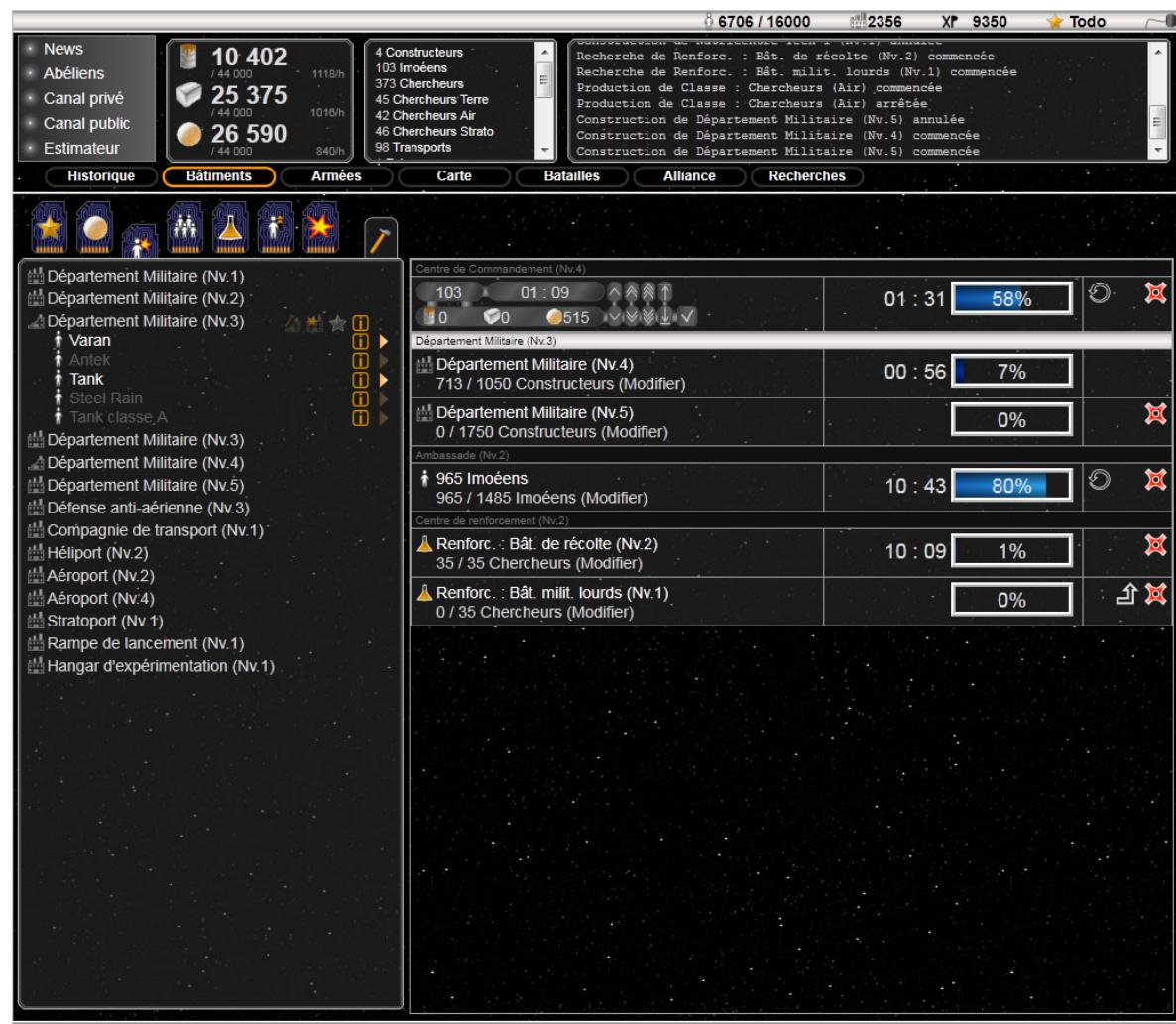
Abelia Wars

Abelia Wars est un jeu de stratégie web se déroulant dans un univers futuriste où l'homme est obligé de coloniser des planètes lointaines pour subvenir à ses besoins.

C'est un jeu que j'ai développé seul pendant 4 ans. Malheureusement, le temps a manqué dans ma vie étudiante et le projet a dû s'arrêter pour diverses raisons.

Ce jeu se démarquait de ses principaux concurrents (comme OGame) surtout par le fait qu'il était possible de perdre, et il y avait un vrai contrôle de territoire, grâce à la présence d'une carte du monde et de points stratégiques. Il poussait à faire des alliances mais également des trahisons au sein de son alliance, pour arriver à ses fins. De plus, il était plutôt difficile, car complet, et nécessitait un investissement du joueur pour élaborer une stratégie sur le long terme. Ce jeu a connu 5 parties fermées en beta, qui comptaient chacune entre 16 et 60 joueurs et qui ont chacune duré entre 3 et 6 mois.

Dans un deuxième temps, le projet fut changé du simple jeu vers un moteur de jeu de stratégie générique permettant à n'importe qui de créer un jeu de stratégie dans l'univers de son choix. Ce moteur utilisait la technologie jQuery, afin d'avoir une interface dynamique et intuitive, ainsi que Smarty, pour le côté multilingue.





Jeux divers

Avalia

Avalia est un jeu réalisé dans le cadre d'un projet universitaire optionnel, avec 6 autres personnes. Il est basé sur la réalité augmentée et se joue à deux joueurs, autour d'un plateau constitué de cases. Les joueurs vont successivement poser des cartes sur le plateau afin de créer un environnement duquel des créatures vont surgir. Le but est de les rediriger, afin de récupérer le plus de créatures bleues et d'éviter le plus de créatures rouges.

Dans cette équipe, j'étais le game designer et gameplay programmer.

Ce plateau est filmé, redressé, les cartes sont décodées, puis le moteur du jeu déplace les créatures en fonction des cartes sur le plateau, et le tout est envoyé à un moteur de rendu qui affiche des entités sur le plateau en réalité augmentée.

Notre vidéo de présentation (de piètre qualité) est sur internet à cette adresse :
<http://vimeo.com/5029851>

Elle est un peu longue et je conseille d'aller directement à 5:00 pour aller directement à l'explication des différentes phases, du redressement du plateau à l'incrustation d'entités, ou encore à 9:00 pour la démonstration et l'explication des règles.

Bunbyx

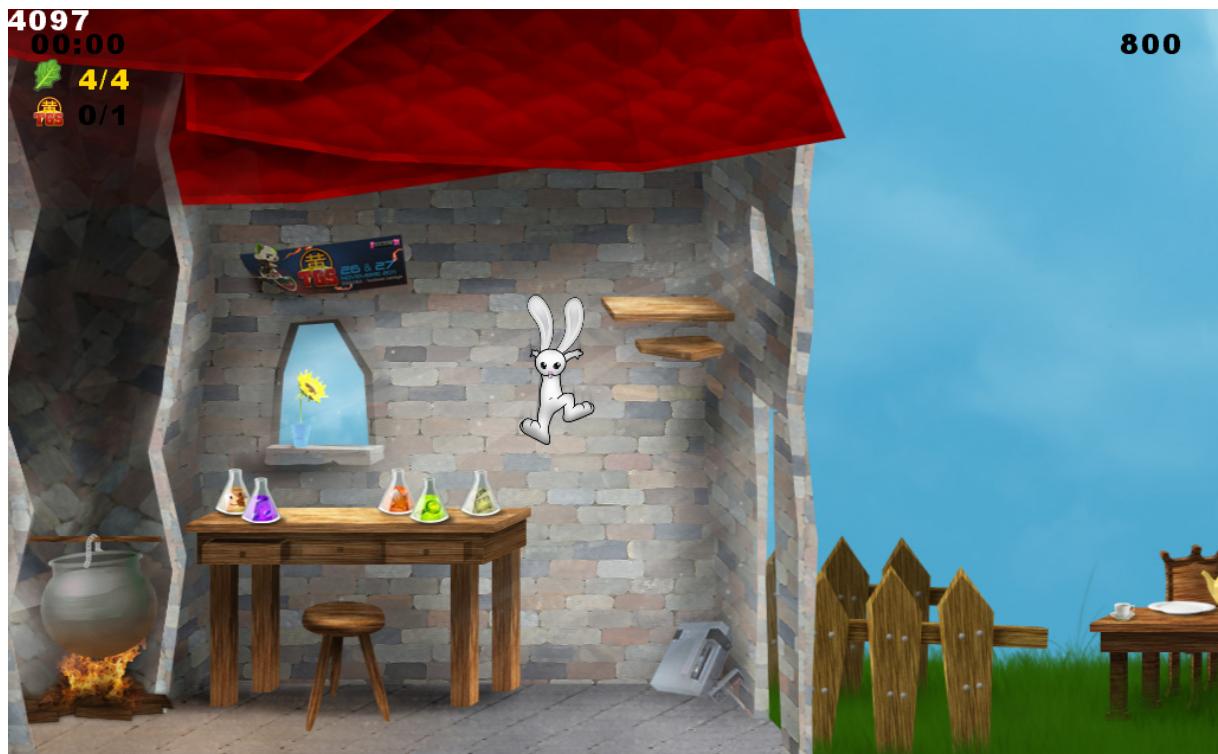
Bunbyx est un jeu réalisé en l'espace de quelques semaines pour le concours de développement de jeu organisé pour le Toulouse Game Show 2012.

Nous étions trois personnes pour réaliser ce jeu, un graphiste et deux développeurs, dont moi.

C'est un jeu de plates-formes classique dans lequel on dirige Bunbyx, un lapin, qui doit aller à la fin du niveau. Pour ce faire, il devra récupérer des items qui vont lui permettre de courir plus vite, sauter plus haut, faire des wall-jumps, afin d'accéder à des endroits auparavant non-atteignables. Le jeu n'est donc pas trop linéaire et demande au joueur de revenir parfois sur ses pas pour récupérer des objets auparavant inaccessibles.

Il a été développé en C#, grâce à la technologie XNA. L'intérêt premier de ce jeu était de découvrir XNA, voir ses capacités et surtout ses limites.

Repository : <https://github.com/aurelien-defossez/GameANiaque>



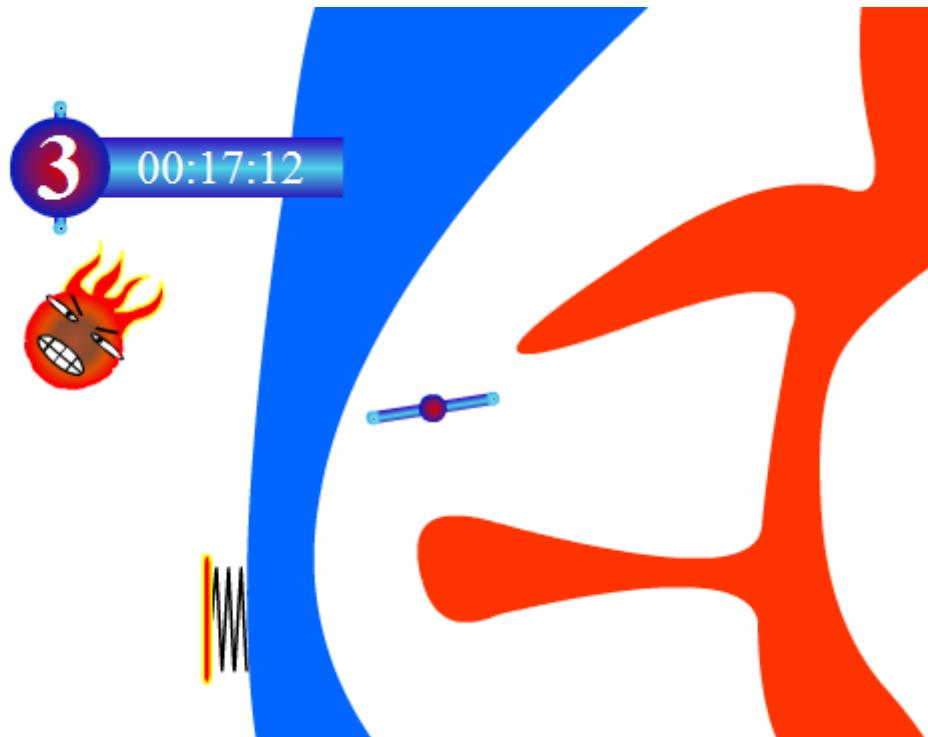
Kuru

Kuru est un jeu que j'ai développé lors de mes débuts de développeur pour apprivoiser Flash et Action Script 2 en 2006.

Le principe du jeu est le même que celui de [Kuru Kuru Kururin](#), jeu dans lequel on dirige une barre en perpétuelle rotation. Le but est de faire traverser cette barre dans un niveau sans toucher les parois. Des ennemis et items parsèment la carte, dynamisant un peu le jeu.

Ce jeu a été pour moi l'occasion de m'essayer au graphisme vectoriel et aux sons, faits à la bouche avec un micro et Audacity, ainsi qu'à plusieurs aspects de la programmation d'un jeu-vidéo, avec notamment la réalisation d'un moteur de collision maison.

Le jeu n'est pas jouable sur internet, ne brille pas par ses graphismes, et se limite à un simple tutorial avec que deux vrais niveaux, mais si vous désirez tout de même y jouer, je peux fournir l'exécutable du jeu.



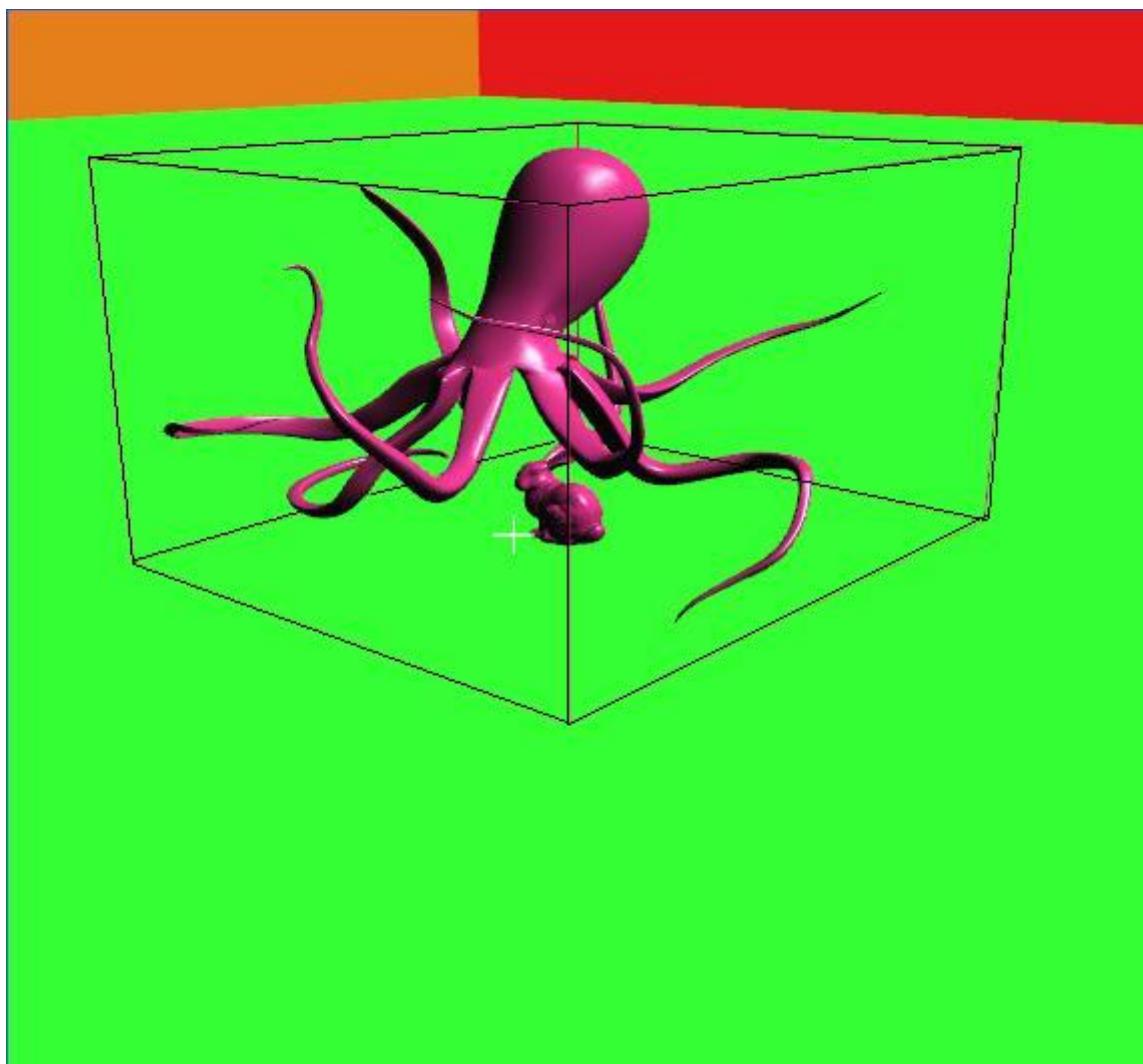
Divers

WiiProject

WiiProject est un projet étudiant réalisé en binôme, ayant pour but de déplacer des objets 3D dans une scène à l'aide de la Wiimote de Nintendo. Il a été réalisé en C++, grâce à la librairie Wiifyourself.

Je me suis surtout occupé de la reconnaissance de mouvements et la prise en main du logiciel, mais aussi un peu du déplacement et de la rotation des objets dans la scène.

Ci-dessous une capture d'écran de l'application. Imaginez déplacer ces objets en visant à l'écran, en tournant la Wiimote ou encore en avançant et reculant :



Mapping

Avant de me mettre au développement, je me suis tout d'abord mis au mapping, notamment pour la série de jeux de Valve Counter Strike.

J'ai créé plusieurs maps pour ces jeux, de 2003 à 2005, qui ont eu des succès plus ou moins grands, mais deux sont sorties du lot à l'époque et ont eu un succès plus grand que prévu car les maps ont été jouées sur d'autres serveurs que celui d'origine sur lequel je jouais, alors que je ne les avais pas personnellement partagées.

Voici quelques screenshots de ces maps :

