

EASY TO LEARN, HARD TO MASTER

EASY TO LEARN, HARD TO MASTER

Une analyse des outils de création de jeux vidéos destinés à un public néophyte.

Mémoire de Master, rédigé, illustré et mis en page par Ivan Gulizia,
étudiant de la filière Media Design de la Haute École d'Art et de Design de Genève.

La fonte utilisée est l'Akkurat, dessinée en 2004 par Laurenz Brunner et distribuée par Lineto.

Imprimé et relié à Venise en décembre 2018.

Table des matières

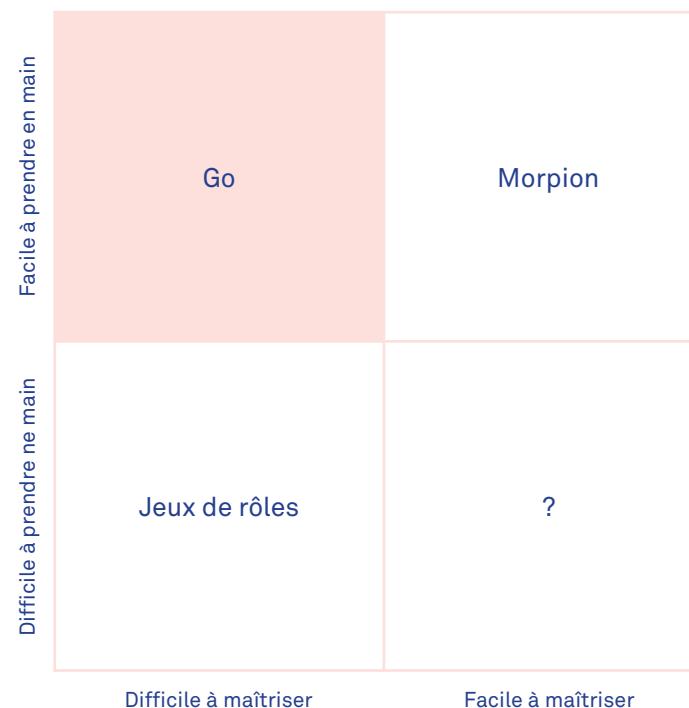
Avant propos	8
Easy to Learn, Hard to Master	-
Introduction	10
Notions	-
Néophytes	-
Expression vidéoludique	14
Contexte	16
Analyse	24
Critères d'analyses	36
Accessibilité	-
Interface et interaction	40
Personnalisation et représentation	41
Communauté et jeux produits	41
Outils sélectionnés	42
Dreams	-
Bitsy Game Maker	50
Scribblenauts	56
Sketch Nation	64
Discussion	74
Contraintes et contextes	-
Un outil idéal	75
Références	79
Bibliographie	-
Crédits iconographiques	81

Avant propos

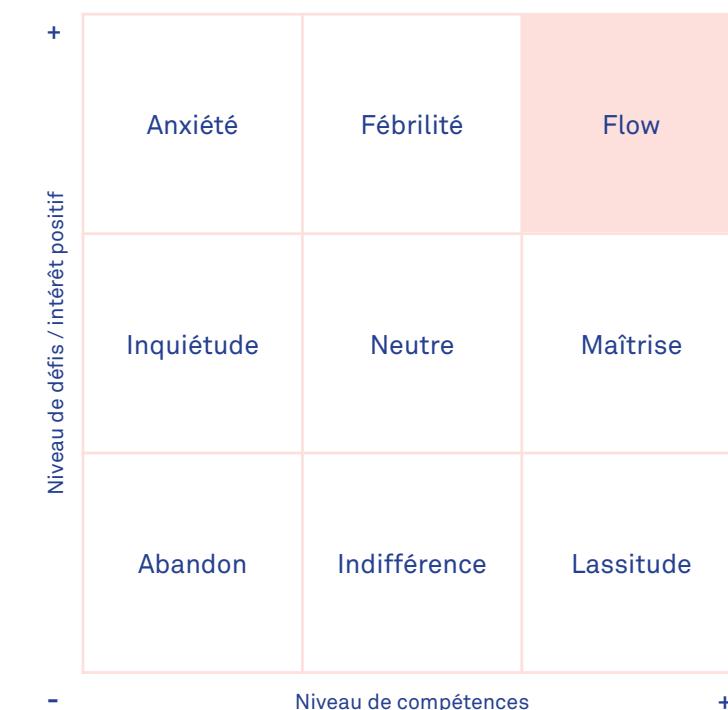
Le titre de ce mémoire “Easy to Learn, Hard to Master” ou “Facile à prendre en main, difficile à maîtriser” en français, fait référence à *la loi de Bushnell*. Cette loi porte le nom de l'auteur de cette phrase, Nolan Bushnell, créateur du jeu *Pong* et fondateur d'*Atari*¹. À l'origine, elle sert à définir la courbe de progression idéale d'un joueur par rapport aux mécaniques d'un jeu. Elle fait d'ailleurs écho à la théorie du flow énoncé par Mihály Csíkszentmihályi², ou *la zone proximale de développement* de Lev Vygotski³. Deux notions importantes que partagent les professionnels des sciences de l'éducation et les game designers.

“ALL THE BEST GAMES ARE EASY TO LEARN AND DIFFICULT TO MASTER. THEY SHOULD REWARD THE FIRST QUARTER AND THE HUNDREDTH.”

Nolan Bushnell



J'ai choisi dans ce mémoire d'appliquer la signification que m'évoque cet aphorisme aux outils de création de jeux vidéo dans le but de découvrir comment définir ou concevoir un outil “idéal” selon des critères que j'identifie au fur et mesure de ma recherche. Quels sont les moyens mis à disposition de l'utilisateur par ces logiciels pour qu'ils soient rapides et simples à prendre en main, de manière autonome dans l'idéal et suffisamment exhaustif pour permettre à de nouveaux utilisateurs de s'exprimer ou se positionner à travers le langage propre au médium vidéoludique, voir à permettre l'émergence de mécaniques de jeux inédites.



¹ Crockett, Larry J. *Easy to Learn, Difficult to Master* [Livre]

² Csikszentmihalyi, Mihaly, & Csikszentmihalyi, Isabella. *Beyond boredom and anxiety* [Livre]

³ Brossard, Michel. *Apprentissage et développement: tensions dans la zone proximale* [Livre]

Fig. 1 Schéma illustrant la loi de Bushnell.

Fig. 2 Représentation schématique de la théorie du flow

Introduction

Notions

Suite au grand boom de la scène des jeux vidéo indépendant durant les années 2000, de nombreux outils gratuits et relativement accessibles ont fait leur apparition, les plus populaires à ce jour étant *Unity* et *l'Unreal Engine*. Leur but est de démocratiser la pratique du game design en permettant aux amateurs et aux studios de développement plus modestes de fabriquer des jeux en utilisant des moyens de production professionnels.

Il existe en effet un réel intérêt de la part du grand public à réaliser ses propres jeux. Cela s'observe, entre autres, par la diversité des jeux disponibles sur des plateformes telles que *Steam* ou *Itch.io* et le nombre toujours croissant de débuteurs prenant part à divers événements axés sur la fabrication de jeux vidéos tels que des ateliers ou des game jams. Malgré cela, cette activité reste encore majoritairement réservée à un public restreint, suffisamment passionné pour avoir investi du temps dans l'apprentissage de connaissances techniques dans au moins l'un des domaines de compétences "requis" pour produire un jeu vidéo.

En effet, la multiplicité des facettes de la création d'un jeu représente une barrière qui peut en décourager plus d'un. Cette activité nécessite des connaissances techniques préalables qui la rendent peu accessible: programmation, manipulation d'interfaces complexes, nombre d'étapes dans le processus de production, variété dans la nature du contenu et gestions des propriétés de ces éléments. La liste n'étant pas exhaustive, il ne s'agit là que de quelques-uns des composants qui complexifient l'élaboration d'un jeu. Ainsi, sans des méthodes et des outils adaptés aux néophytes, cette pratique resterait malheureusement inaccessible pour un grand nombre de personnes. Il s'agit d'autant d'auteurs potentiels se trouvant hors des circuits habituels, qui selon moi, pourraient apporter un regard nouveau sur le médium vidéoludique hors de toute logique commerciale.

Ces réflexions m'ont mené alors naturellement à me pencher sur l'exploration de ces outils dans le but de découvrir quels sont les moyens mis en oeuvre par leurs créateurs pour les rendre accessibles. Mais également d'identifier des opportunités quant à leurs utilisations par un public néophyte dans des contextes définis plus loin dans ce mémoire.

Néophytes

Un néophyte désigne par essence une personne n'ayant aucune connaissance particulière dans un domaine donné. De façon plus explicite, un néophyte est défini ici comme une personne étant capable de comprendre et d'utiliser une interface informatique basique telle qu'un navigateur web. Elle n'a a priori aucune connaissance en programmation, mais est tout à fait en mesure d'apprendre et de s'approprier des notions de bases comme la manipulation de déclencheur, de variables et la création de conditions, qui dans un jeu, servent à déterminer l'état et les événements qui peuvent se produire. Elle sait aussi exprimer une idée par le dessin, produire et manipuler des images et des sons avec ou sans assistances d'outils informatiques, et a les aptitudes pour raconter une histoire.

Joueuse passionnée ou n'ayant quasiment jamais joué, elle saisit le concept général de ce qu'est un jeu vidéo et a probablement déjà des connaissances de certains codes liés au médium. En effet, les deux dernières décennies ont vu un changement important dans la perception des jeux vidéo au sein du grand public. Ils font partie intégrante de la culture populaire et sont devenus un facteur d'intégration sociale important chez les jeunes notamment en favorisant les échanges et la communication en dehors des pratiques de jeu.

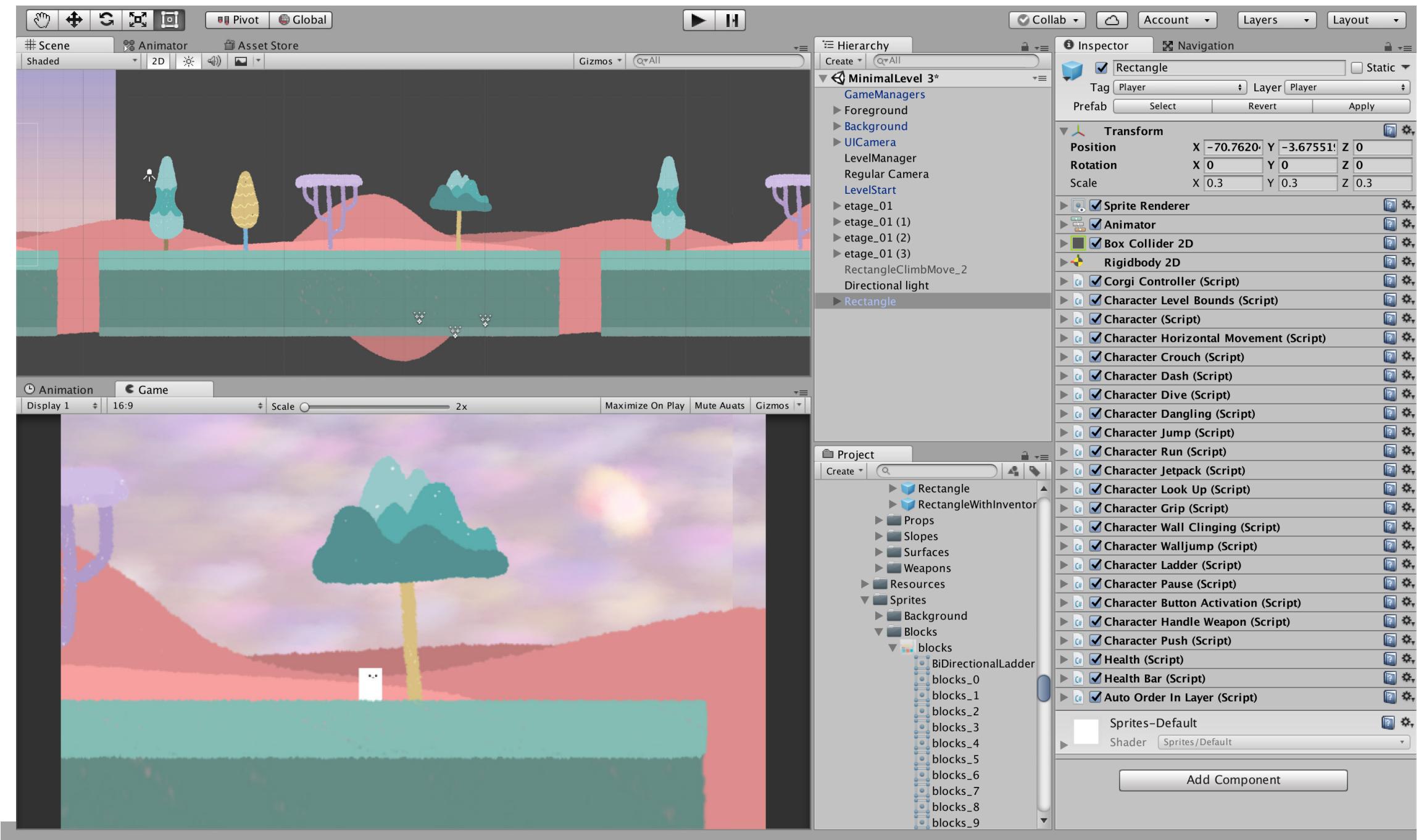


Fig. 3 Capture d'écran de l'interface d'Unity.

Expression vidéoludique

Les jeux vidéo sont susceptibles aussi bien de promouvoir des valeurs culturelles et sociales, que de les critiquer⁴. Mais selon Sébastien Genvo, à qui l'on doit le terme “expression vidéoludique” ce médium est aussi un moyen d’expression à part entière, vecteur d’émotions capables de nous sensibiliser en nous proposant de nous mettre à la place d’autrui et d’explorer ses problèmes psychologiques, sociaux, culturels en faisant l’expérience des dilemmes moraux et éthiques qui en résultent, avec leurs conséquences⁵. Ces caractéristiques ne sont pas uniquement liées au contenu esthétique d’une oeuvre. Le message que l’auteur souhaite véhiculer est davantage transmis à travers le système de règles et de contraintes, définies par les représentations et les interactions que son jeu propose.

Un exemple de jeu expressif connu est *Lieve Oma*, créé par Florian Veltman, qui nous invite à incarner une jeune fille lors de promenades aux champignons en forêt avec sa grand-mère. Ce jeu a pour origine une expérience autobiographique⁶ et peut faire écho à un article de Gonzalo Frasca écrit en 2001, “The Sims : Grandmothers are cooler than trolls”⁷, dans lequel l'auteur raconte son désintérêt pour les univers sur-représentés comme celui de l'héroïque fantasy et les thématiques prédominantes présentes dans les jeux vidéos à ce moment-là. Il propose au contraire de pouvoir vivre le quotidien et d'affronter les défis que rencontrent des personnages tels que celle d'une mère monoparentale grâce à la modification d'un jeu comme les *Sims*.

Les dynamiques et formes de rhétoriques propres au médium qui en émerge peuvent notamment s’appliquer dans des contextes pédagogiques, thérapeutiques, culturels ou politiques. C'est notamment pour ces raisons que je souhaite explorer ce potentiel expressif lors d'ateliers de création de jeux vidéo ayant lieu dans des contextes particuliers que je développe dans la partie suivante. Il s'agit donc de l'un des motifs qui m'ont amené à m'intéresser aux outils de création à destination d'un public néophyte.

Par exemple, Pierre-Yves Hurel, doctorant et assistant au Département des Arts et Sciences de la Communication de l'Université de Liège et membre du Game Lab de “La Cité Ardentes”, présente dans son blog personnel⁸ un atelier qu'il a mené dans un quartier multiculturel de Bruxelles, où il proposait à des jeunes migrants de raconter leurs parcours à travers la création d'un mini jeu en utilisant des outils adaptés à chaque étape: Du papier pour écrire leurs histoires, le jeu/éditeur de niveaux *Mario Maker* pour prototyper une première version, et le moteur de jeu *Construct 2* pour donner vie à leurs récits.

⁴ Grace, Lindsay. *Critical games: Critical design in independent games* [Pdf]

⁵ Genvo, Sébastien. *Comprendre et développer le potentiel expressif* [Livre]

⁶ Grine, Esteban. *Lieve Oma, les grands-mères sont plus cool que les troll* [Article]

⁷ Frasca, Gonzalo. *The Sims: Grandmothers are cooler than trolls* [Article]

⁸ Hurel, Pierre-Yves . *Faire créer des jeux sur la migration à des jeunes concernés* [Article]

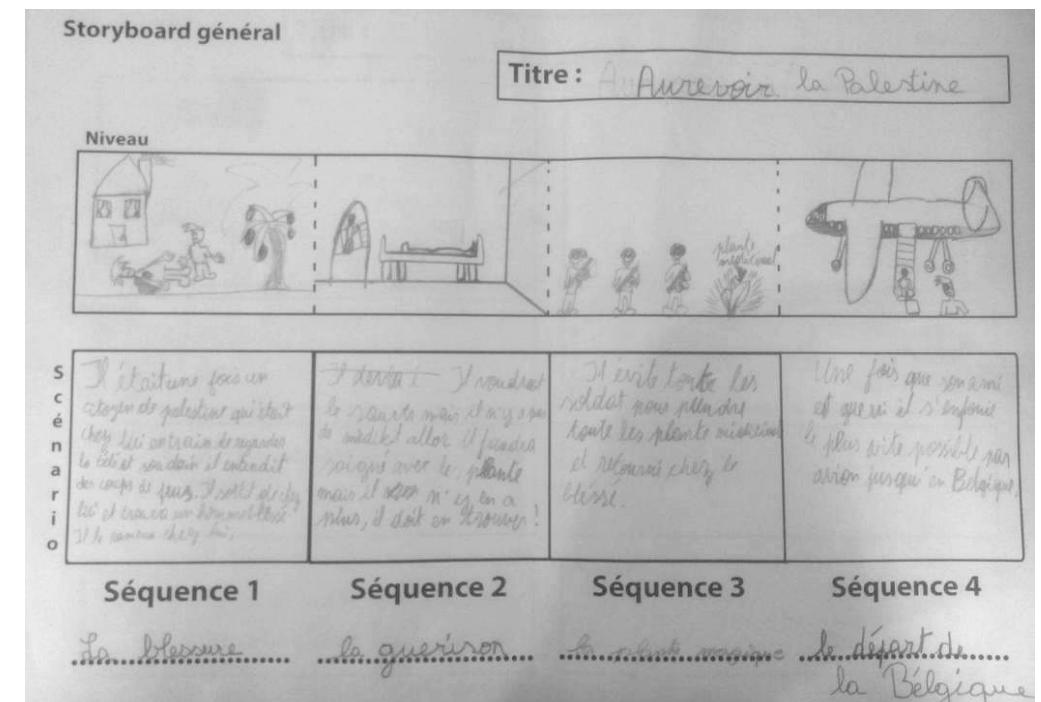


Fig. 4 Image tirée du jeu *Lieve Oma*.

Fig. 5 Photo issue de l'atelier dirigé par Pierre-Yves Hurel.

Contexte

Dans ce mémoire, j'ai fait le choix de mettre en perspective les réflexions liées aux outils de création de jeu vidéo avec un contexte d'utilisation particulier, celui d'ateliers dédiés à l'expression vidéoludique.

En effet, parallèlement à l'écriture de ce mémoire, j'ai eu l'occasion d'organiser de nombreux ateliers avec des publics à chaque fois différents, mais étant néanmoins tous néophytes. Par exemple, avec un petit groupe d'enfants âgés de 11 à 13 ans encadré par une art-thérapeute et des éducateurs dans un foyer pour jeunes en difficultés, les Écureuils Doret. Où depuis plus d'un an, le suivi hebdomadaire d'Elias, un jeune autiste passionné de jeux "point and clic" surréalistes, particulièrement ceux créés par le studio Amanita Design, que j'accompagne dans le but de l'aider à donner vie à son univers singulier, à travers la réalisation de son propre jeu.

Durant ces ateliers j'ai eu l'occasion d'essayer deux types d'outils, *Unity* pour commencer, que j'ai rapidement abandonné pour de multiples raisons telles que le temps restreint alloué aux sessions, la difficulté de prises en main pour un novice ou le matériel nécessaire à son fonctionnement. Mais surtout, car sa difficulté rendait mon rôle d'intervenant plus proche d'un opérateur veillant à ce que tout fonctionnement correctement, au détriment d'un accompagnement des participants plus axés sur la transmission de savoir lié à l'aspect conceptuel et créatif de la création de jeux. À défaut d'avoir trouvé un outil idéal, mon choix s'est alors porté depuis sur *Sketch Nation*, l'un des logiciels analysés plus loin dans ce mémoire.

Ainsi, j'ai pu profiter de ces ateliers pour notamment prendre connaissance des difficultés encourues par les participants lorsqu'ils utilisent ces outils. Par exemple, j'ai pu constater lors de sessions ayant eu lieu au foyer, une certaine aisance dans la manipulation de l'interface de *Sketch Nation* par certains jeunes et un renversement des connaissances liées à la culture vidéoludique entre les adultes et les enfants. Ce décalage est assez représentatif à mon sens, de ce qu'un atelier de ce type peut apporter dans ce contexte d'échange bien précis.



Fig. 4 - 13 Captures d'écrans de *Moon Does Stress*, un jeu réalisé lors d'un atelier à l'aide d'une méthode hybride mêlant éléments tangibles et numériques.



Fig. 14 Reproduction du support papier ayant servi à créer le personnage principal de Moon Does Stress, le corps est le résultat d'un processus de vote visant à élire chaque membre le constituant.

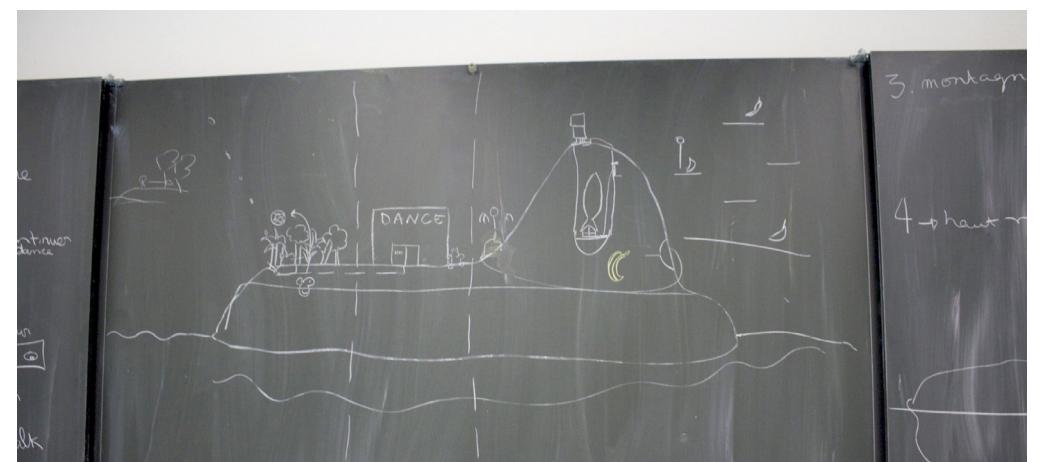


Fig. 15 - 17 Photos prises lors de cet atelier ou l'on peut observer la méthode employée basée sur l'utilisation quasi exclusive de matériel non informatique.

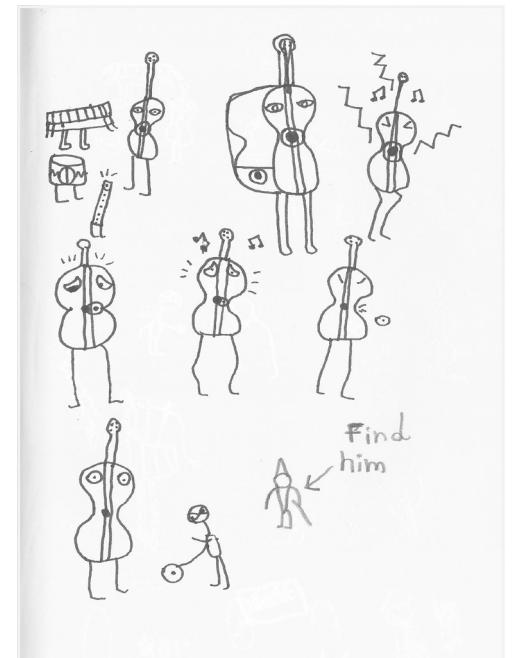
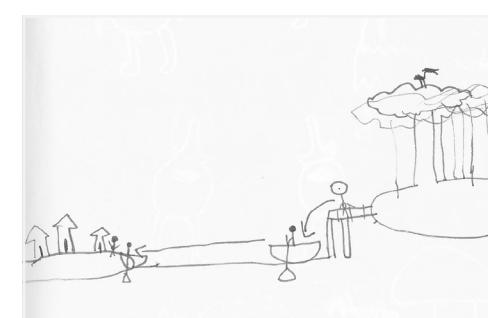
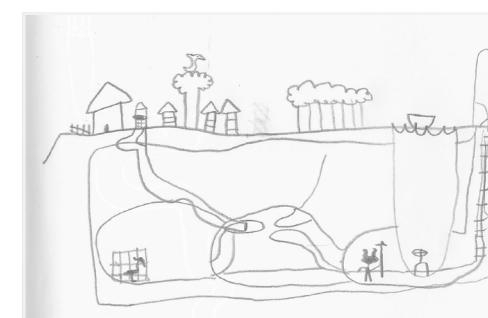
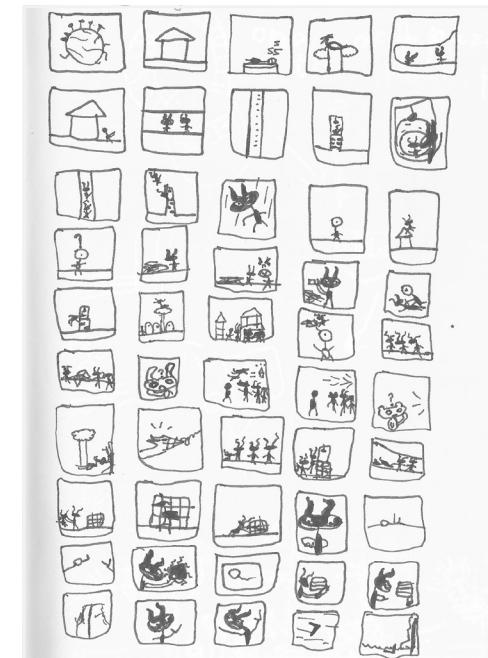
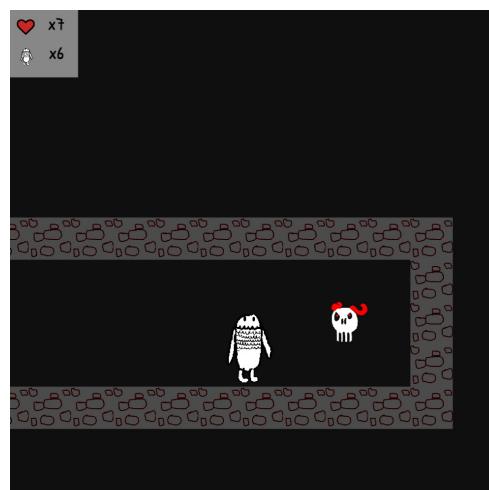


Fig. 18 - 23 Captures d'écran du prototype du jeu d'Elias réalisé à l'aide de Sketch Nation.
Fig. 24 - 28 Dessins de conception d'Elias.

Ces ateliers m'ont également permis d'observer d'autres opportunités d'apprentissage qui peuvent être développées à travers le processus de création d'un jeu vidéo.

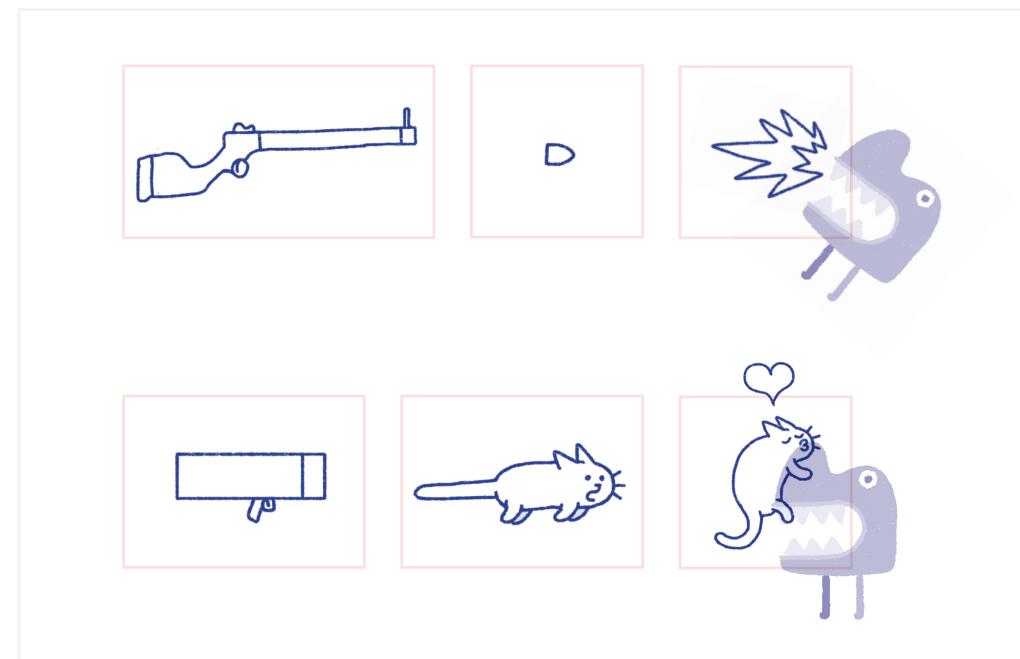
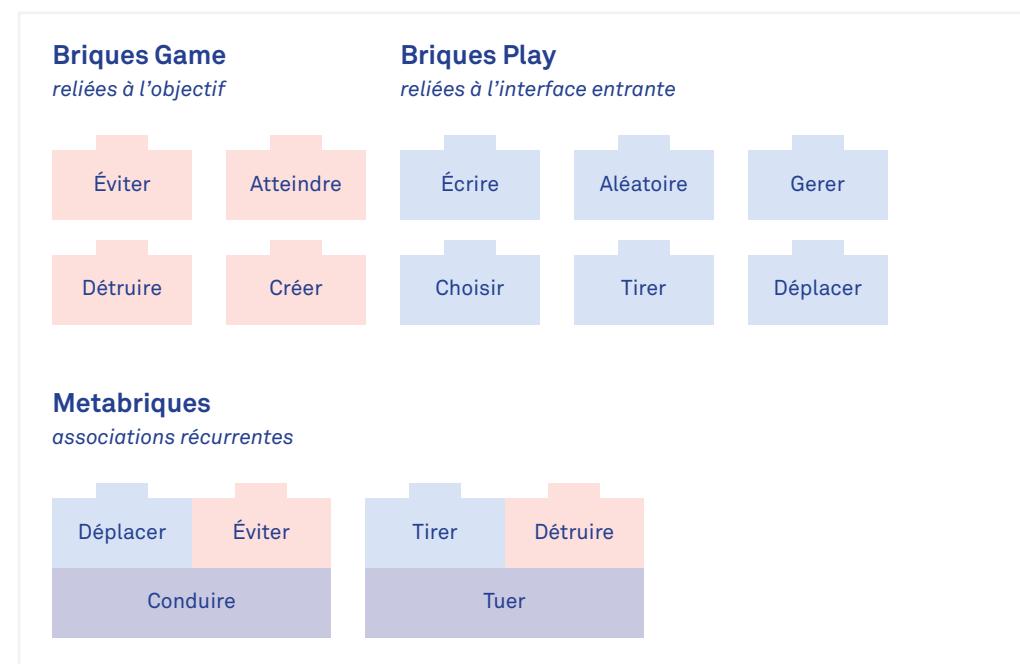
La transdisciplinarité inhérente à cette pratique favorise notamment le travail en équipe et permet d'aborder tous types de problématiques et de les traiter en utilisant le langage et les caractéristiques propres au jeu vidéo, à savoir selon moi, le game design appliquée à un environnement simulé, interactif et imprévisible.

Par exemple, (dé)construire les règles qui régissent un univers virtuel et les éléments qui le composent me semble être un moyen intéressant de consolider son esprit critique et permet également d'apporter un regard nouveau sur sa propre pratique de joueur. Ce qui a entre autres pour effet de repousser les risques liés au jeu excessif selon Niels Weber, Psychologue spécialisé dans les cyberaddictions.

Les jeux vidéos permettent donc de pouvoir traiter des sujets variés. Ainsi, on peut amener les participants à questionner, par exemple, la représentation stéréotypée des personnages et de leurs rôles dans les œuvres de fictions ou s'interroger sur la signification que peuvent avoir certaines mécaniques de jeu.

En effet, il est intéressant de constater que les règles d'un jeu et les dynamiques qu'elles produisent, lorsqu'elles sont mises en évidence, peuvent être questionnées à cause de leurs omniprésences et par leur aspect normatif (pour gagner, il faut conquérir des territoires, détruire des entités, amasser des ressources, etc..).

De même, lorsqu'on approfondit la réflexion en analysant par exemple un certain type de gameplay jugé violent comme celui d'un jeu de tir, en mettant à plat sa mécanique principale et en isolant les ingrédients qui la constituent, notamment grâce à un modèle inspiré par la proposition de Julian Alvarez et Damien Djaouti et al. "les briques de gameplay"⁹. Il devient alors facile de mettre en perspective l'impact produit par un élément en séparant simplement la couche esthétique de la couche mécanique.



⁹ Alvarez, Julian, Djaouti, Damien, Jessel, Jean-Pierre, Methel, Gilles, Molinier, Pierre. Morphologie des jeux vidéo [Article]

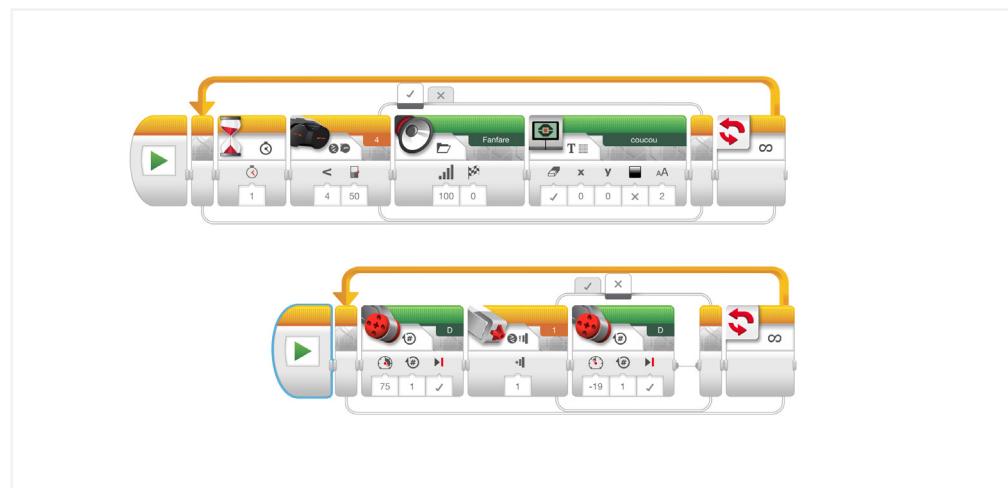
Fig. 29 Schéma des 10 briques élémentaires de gameplay.

Fig. 30 Schéma permettant d'identifier les ingrédients visuels constitutifs d'une mécanique de tir dans un jeu et leurs impacts sur le sens donné à l'action.

Analyse

Afin de proposer une analyse des outils existants, et sans avoir une ambition exhaustive, j'ai choisi dans un premier temps de rechercher et tester un maximum d'outils et de jeux (plusieurs centaines) qui intègrent une notion de création de contenu par les utilisateurs, sans que ceux-ci soient nécessairement conçus à l'origine pour réaliser des jeux vidéos. Figurent ainsi dans la liste¹⁰, un logiciel servant à prototyper des applications comme *Proto.io*, Autodesk *Sketchbook Motion*, une application d'animation novatrice inspirée des travaux sur les interfaces de Bret Victor¹¹, ou encore *Mindstorm*, un programme développé par le MIT Media Lab, basé sur les recherches de Seymour Papert¹², qui permet de donner vie à des robots Legos à l'aide d'un langage de programmation visuel extrêmement intuitif. Dans un second temps, j'ai décidé d'effectuer une première sélection en appliquant les trois critères majeurs décrits ci-dessous afin d'obtenir un corpus réduit à une trentaine de cas.

À noter que, bien qu'ayant découvert beaucoup de ces outils grâce au très précieux site web "Game Tool Classification"¹³. J'ai préféré ne pas appliquer rigoureusement les catégories de classification proposées par Damien Djaouti dans sa thèse "Serious Game Design: Considérations théoriques et techniques sur la création de jeux vidéo à vocation utilitaire"¹⁴ afin de développer une vision personnelle de ces logiciels et de réussir à définir empiriquement les caractéristiques qui me semblent importantes.



¹⁰ Liste consultable à la page 32 de ce mémoire.

¹¹ Brett, Victor. Stop Drawing Dead Fish [Vidéo]

¹² Papert, Seymour. *Mindstorms: Children, computers, and powerful ideas* [Livre]

¹³ creatools.gameclassification.com [Site web]
¹⁴ Djaouti, Damien. *Serious Game Design Considérations théoriques et techniques sur la création de jeux vidéo à vocation utilitaire* [Thèse]

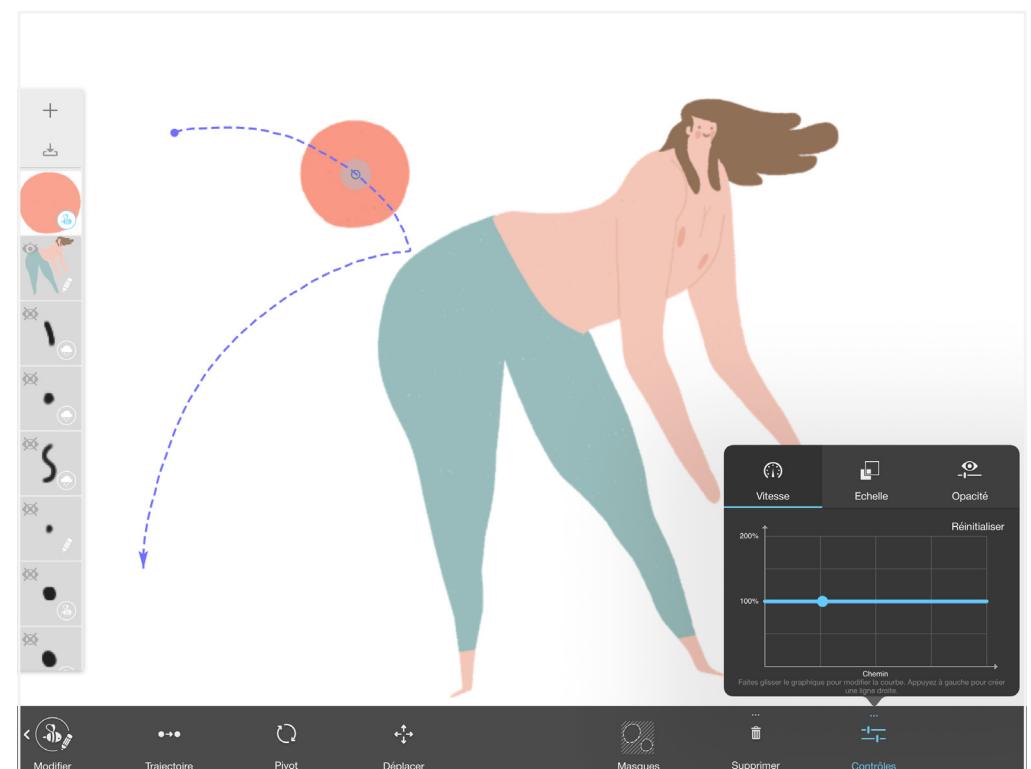
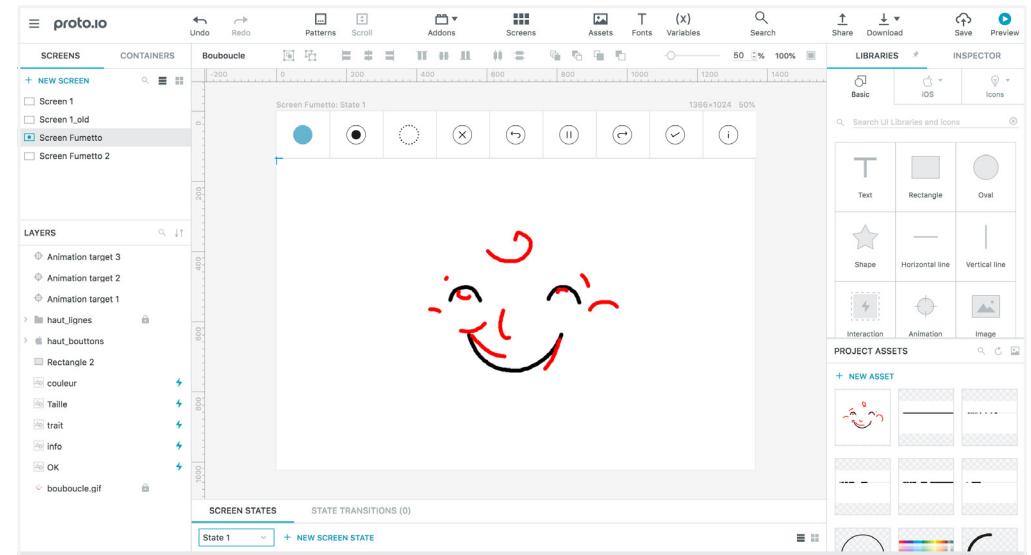


Fig. 31 Détail d'une capture d'écran du programme Mindstorm

Fig. 32 Capture d'écran du logiciel Proto.io

Fig. 33 Capture d'écran de l'application Sketchbook Motion

Le premier critère de sélection est l'existence ou non d'une interface graphique de création, et plus précisément, si l'outil intègre un langage de programmation visuel, en écartant d'ores et déjà ceux qui sont centrés sur la programmation textuelle. Une fois ceux-ci exclus, il reste parmi ces outils, majoritairement deux paradigmes de programmation qui sont les logiciels axés sur la création d'événements par bloc tel que *Scratch* ou *Construct* et ceux qui sont centrés sur une logique de circuits avec des cartes reliées entre elles, appelé "noeud" tel que *Playmaker* ou *Dreams*. Cette dernière solution à l'avantage d'être très similaire aux schémas d'une machine d'état souvent utilisé par les game designers pour illustrer par exemple le comportement d'une intelligence artificielle.

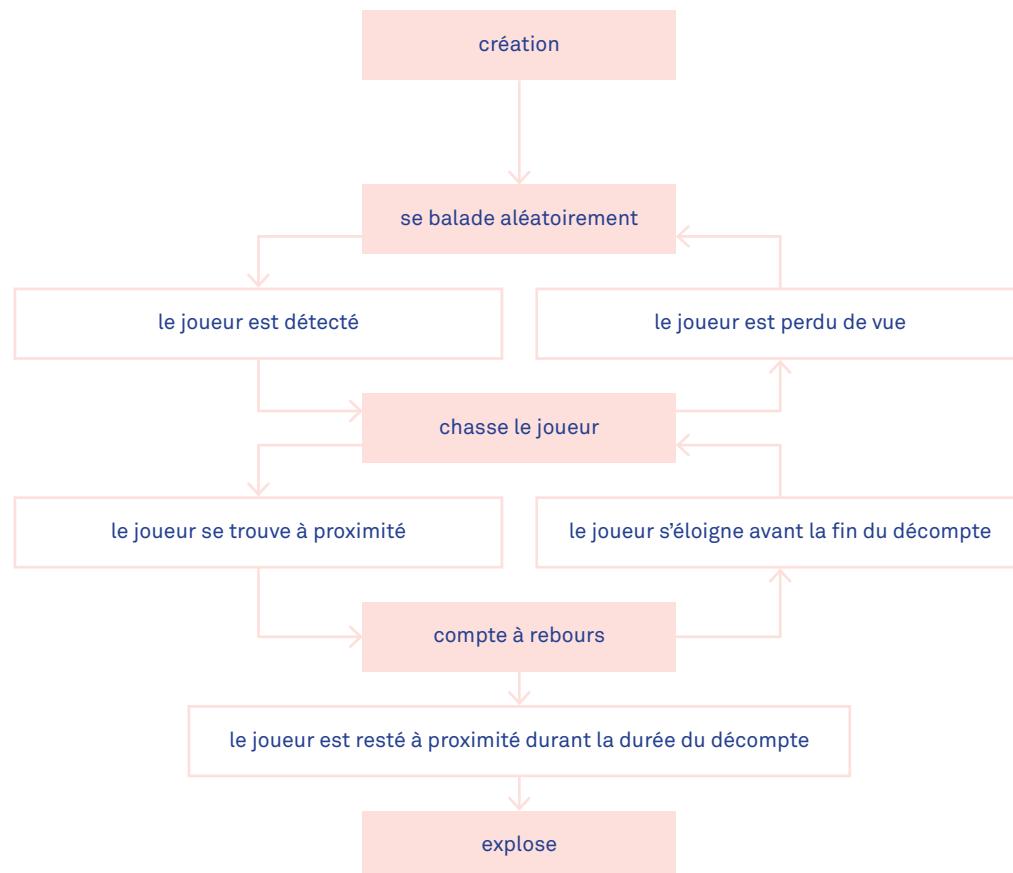


Fig. 34 Schéma simplifié du comportement d'un Creeper (ennemi iconique issu du jeu Minecraft) représenté sous la forme d'une machine d'état finie.

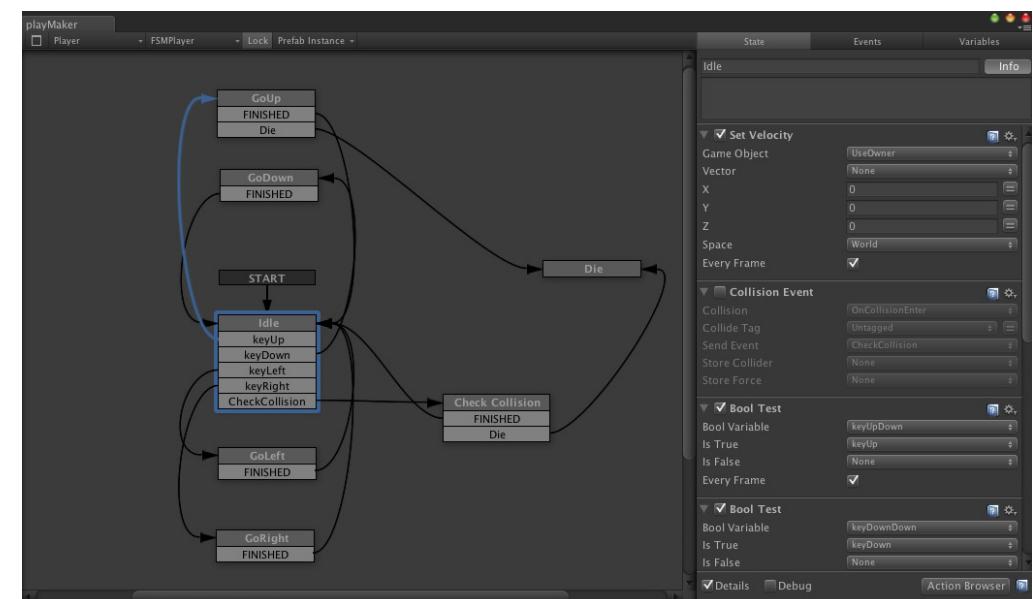
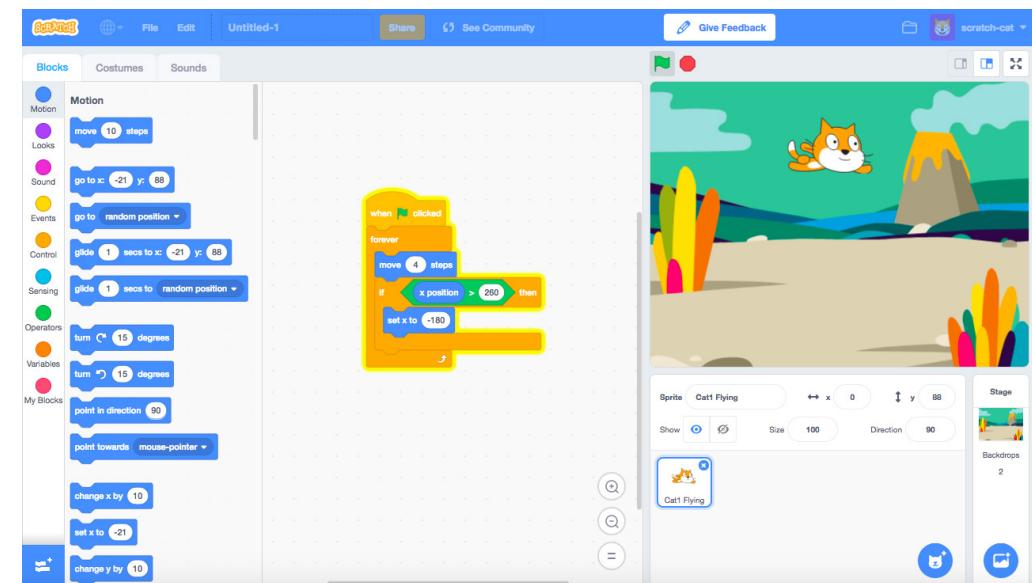


Fig. 35 Interface de Scratch 3.0.

Fig. 36 Fenêtre issue de Playmaker, un plug-in intégré à l'environnement d'Unity.

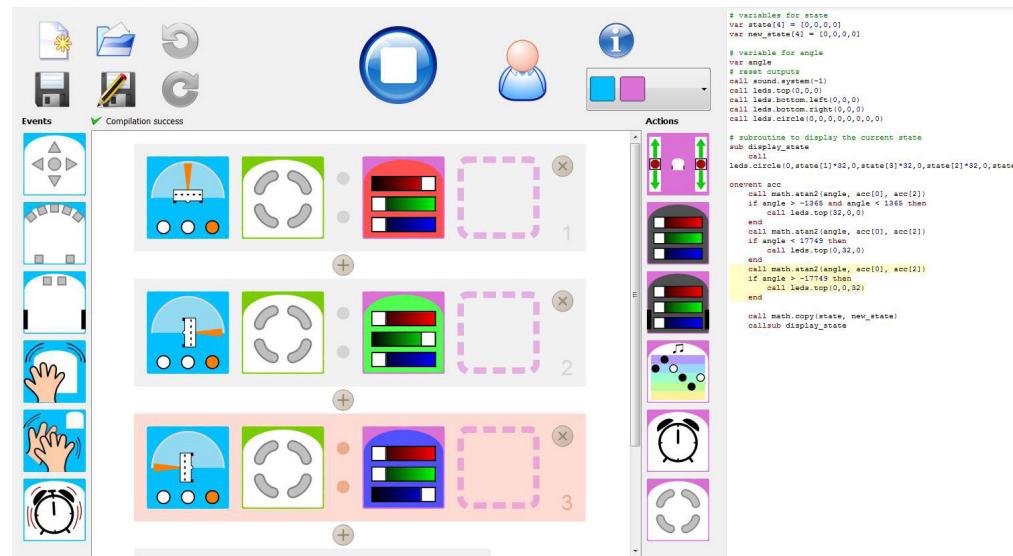
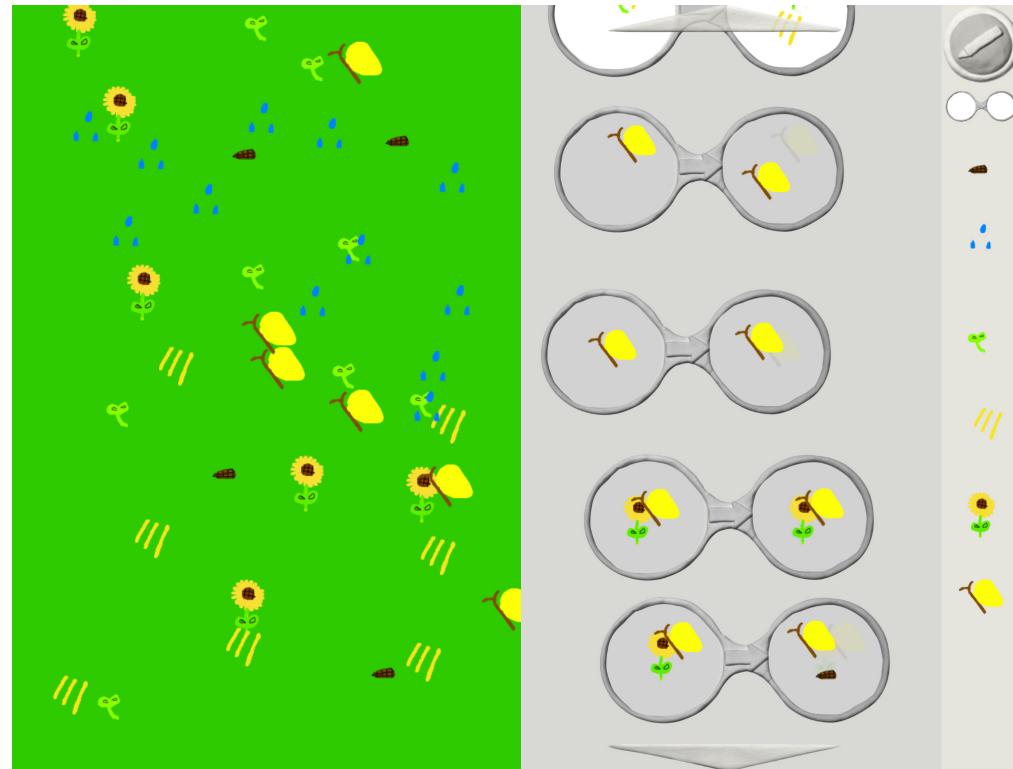


Fig. 39 Image de l'espace de création de Viscuit illustrant le paradigme particulier de cet outil ou les conditions sont créées en plaçant des images à l'intérieur des "lunettes".

Fig. 40 Image du langage de programmation visuel développé pour contrôler le robot Thymio.

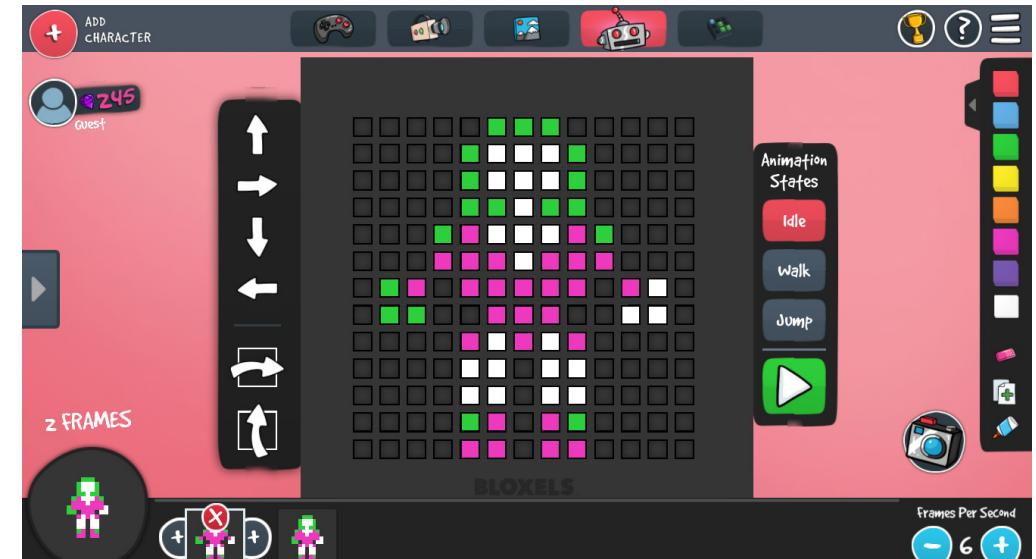
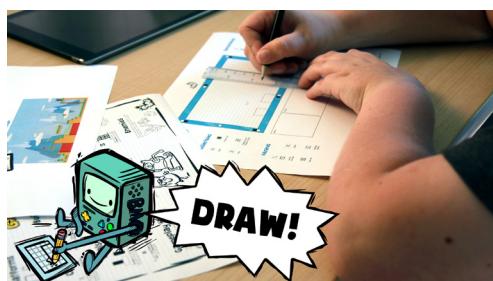


Fig. 41 Image tirée de l'application Bloxels où l'on peut observer comment les développeurs ont reproduit virtuellement le plateau physique autour duquel cet outil est centré.

Fig. 42 Image de l'environnement de programmation séquentiel proposé dans Kodu semblable à un mélange entre de la bande dessinée et la logique "IFTTT" (If This Then That).



Ensuite vient comme deuxième critère de sélection la possibilité de créer ou d'importer ses propres visuels, animations, sons et autres éléments esthétiques. L'idée est de privilégier les outils facilitant la création de son propre contenu, ce qui permet donc à l'utilisateur de s'exprimer en donnant vie à son propre univers.

À ce stade, les logiciels centrés sur un univers unique ou une licence ne sont pas retenus à l'exception d'outils se distinguant vis-à-vis du 3e et derniers critères de sélection, tels que *Mario Maker*, *Adventure Times Game Wizard* ou *Spore*.

En effet, ces 3 exemples ainsi que d'autres logiciels sélectionnés ont en commun de proposer une interface ou une expérience originale par rapport aux outils de création les plus populaires comme ceux de la *Creative Suite* d'Adobe ou *Unity* dans le cas des jeux vidéos, par exemple, *Bloxels* intègre des éléments tangibles alors que *Viscuit*, propose un environnement de programmation totalement inédit et singulier basé uniquement sur des images mises en relation entre elles pour décrire visuellement une condition et le résultat obtenu lorsqu'elle est remplie.

Ces trois critères m'ont donc permis de sélectionner des outils de création pour constituer un tableau d'analyse contenant une trentaine de cas. Au fur et à mesure de ces tests de ces logiciels, j'ai identifié de nouveaux critères pour décrire ces différents outils. Ces catégories serviront ensuite comme base pour définir les critères d'analyse que j'applique par la suite à une sélection restreinte à quatre cas. Ce corpus final a pour but de résumer quatre familles d'outils différents que le tableau permet de mettre en lumière et qui méritent selon moi une analyse plus approfondie.

Fig. 37 Image illustrant le moyen d'interaction principal dans Super Mario Maker.

Fig. 38 Images détaillant le processus de création de l'application Adventure Time Magical Wizard.

Nom	testé	cible	plateformes	model éco.	programmation	tuto
Sploader	oui	édu, amateurs	web	gratuit	mini event sheet	tour d'horizon
Bitsy	oui	amateurs	web	gratuit	ifttt	communauté
Bloxels	oui	édu, amateurs	tablettes	gratuit (plateau payant)	non	?
Mario Maker	non	joueurs	wiiu, 3ds	payant	non (oui exploit)	?
Mega Maker	oui	joueurs	win	gratuit	non	interactif
PixelPress ATMW	oui	joueurs	tablettes	payant	non	interactif
Proto lo	oui	pro	web	freemium	oui	vidéos
Sketchbook Motion	oui	amateurs	ios	freemium	non	interactif
Buildbox	non	amateurs, pro	mac, pc	payant	flow	vidéos
Construct 3	oui	édu ?, amateurs, pro	web	freemium	bloc	vidéos
Scratch	oui	joueurs, édu	web	libre	bloc	vidéos
Mindstorm	oui	édu, amateurs	... (sauf console)	gratuit (tangible payant)	flow	vidéos
Minecraft	oui	joueurs, édu	...	payant	redstone	communauté
Dreams	non	joueurs	ps4	payant	flow	?

2d/3d	animation	assets	interaction	template, modèles	communauté	prise en main	possibilités
2d	sprites	prefabs, outils, communauté	drag/drop	oui (outil, communauté)	oui	*	**
2d	sprites (2frames)	outils	paint	oui (outil, communauté)	itch.io	*	**
2.5d	sprites	prefabs, outils, communauté	paint	–	oui	*	**
2d	–	prefabs (univers Mario)	paint	oui, communauté	oui	*	**
2d	–	prefabs (univers Mega Man)	paint	–	oui	*	*
2d	–	prefabs (univers Adventure Time)	paint, tangible (papier + photo)	–	oui	*	*
2d	–	prefabs, outils, import	drag/drop	oui (outil)	–	*	**
2d	oscillations et particules	–	–	oui (outil)	?	*	*
2d	tween	?	?	oui (outil, communauté, store)	?	?	?
2d	...	outils, import	drag/drop	oui (outil, communauté, store)	oui	**	****
2d	sprites	outils, import	drag/drop	oui (commu.)	oui	**	****
–	–	–	drag/drop	oui (commu.)	non	**	***
3d	–	prefabs (univers Minecraft)	in game	oui (commu.)	oui	*	?
...	puppet, procedural	prefabs, outils, communauté	sculpture, ...	oui (outil, communauté)	oui	**	*****

Fig. 43 - 44 Reproduction “brute” du tableau ayant servi à l’analyse factuelle de la première sélection d’outils (raccourci de plusieurs colonnes pour des questions d’espace).

Nom	testé	cible	plateformes	model éco.	programmation	tuto
Spore	oui	joueurs	pc, mac	payant	non	–
ReadyMaker	oui	amateurs	tablettes	gratuit	ifttt	vidéos
Puzzle Script	oui	amateurs	web	gratuit	simple script	?
Sketch Nation	oui	édu, amateurs	... (sauf console)	gratuit	non (mini choix variables)	non
Playmaker	oui	amateurs, pro	Unity	payant	flow (state machine)	vidéos
Corgi Engine	oui	amateurs, pro	mac, pc (Unity)	payant	motif variables	vidéos
Little Big Planet	oui	joueurs	ps4	payant	node	vidéos par objets
Fungus	oui	amateurs, pro	Unity	gratuit	flow	vidéos
Gladiabots	oui	joueurs	pc, mac	payant	behavior tree	interactif
Scribblenauts Unlimited	oui	joueurs	... (sauf mac)	payant	ifttt (basique)	–
Viscuit	oui	édu	web, ios	gratuit	ifttt visuel	vidéos
Hyperpad	oui	édu, amateurs	ios	freemium	flow	videos
Flowlab	oui	amateurs	web	freemium	flow	videos
My doodle game	oui	amateurs	pc	gratuit	non (3 variables)	–

2d/3d	animation	assets	interaction	template, modèles	communauté	prise en main (*>*****)	possibilités (*<*****)
3d	–	prefabs, outils	sculpture	–	?	*	*
2d	?	prefabs, import, communauté	drag/drop	non	oui	**	***
2d	sprites	outils	texte, paint	oui (outil, communauté)	oui	***	***
2d	sprites	outils, import	paint	oui (outil, communauté)	oui	*	**
–	–	–	drag/drop	?	–	**	****
2d	...	prefabs, import, store	drag/drop	oui (package)	non	**	***
2.5d	puppet	prefabs, outils, import, communauté	drag/drop	oui communauté	oui	**	****
–	–	–	drag/drop	oui (package)	–	*	***
–	–	–	menu	oui (missions)	–	*	**
2d	–	outils, communauté	drag/drop	oui (dictionnaire)	steam workshop	*	**
2d	sprites	outils	drag/drop, dessin	–	non	*	**
2d	sprites, tween	outils, import	drag/drop	oui (outil)	oui	**	***
2d	sprites	outils, import	paint	oui (outil)	oui	**	***
2d	sprites	outils	drag/drop, dessin	–	non	*	*

Critères d'analyses

Avant de nous pencher sur l'analyse des outils, il est important de définir et de décrire les différents critères qui nous permettront de les comparer. Ces catégories proviennent en toute logique du tableau utilisé dans la précédente partie.

Accessibilité

Le terme "accessibilité" est employé ici dans sa définition la plus large et n'est pas relatif au monde du handicap. Ma recherche étant orientée vers des logiciels destinés aux néophytes, il m'a donc semblé important d'en faire mon critère d'analyse principal. De cette manière. J'ai privilégié ceux qui peuvent être pris en main en quelques minutes ou qui sont susceptibles de produire très rapidement des résultats, et ce de manière la plus autonome possible.

Afin de déterminer si un outil est accessible ou non, l'un des aspects les plus importants à étudier concerne la quantité de connaissances préalablement requises par l'utilisateur avant de pouvoir réussir à produire son jeu, aussi modeste soit-il. En effet, l'univers vidéoludique a un jargon et des mécaniques qui lui sont propres. Ainsi, une simplification du langage technique devient nécessaire.

Certains outils mettent à disposition des projets contenant toutes sortes d'éléments prédéfinis, appelés *templates*, qui correspondent à des motifs récurrents dans certains types de jeu. Il s'agit souvent d'éditeurs de niveau qui offrent un choix très limité, mais qui se révèlent parfois plus profonds et complets qu'ils n'y paraissent à l'instar de *Sketch Nation*. Utiliser une base préconstruite est une bonne manière d'apprendre en explorant et s'appropriant le travail d'une personne plus expérimentée, par exemple, en commençant par modifier certaines variables. En expérimentant ainsi, il est possible d'acquérir progressivement de la confiance et des connaissances en restant toutefois sur un chemin balisé.

Une autre notion sur laquelle il faut se pencher est la courbe d'apprentissage qui peut être définie par le temps nécessaire à un néophyte pour se familiariser avec les principes de bases d'un outil et les possibilités de progressions offertes à l'utilisateur sur le long terme. Certains dispositifs sont plus difficiles à prendre en main que d'autres, mais cette recherche m'a permis de constater que certains logiciels permettant d'aboutir sur le même type de jeux ne sont en réalité pas aussi simple l'un que l'autre à aborder. Par exemple, dans le cas d'un jeu de plateforme 2D, avec

une application comme *Scratch* il est assez simple d'obtenir rapidement un résultat gratifiant qui peut pousser un débutant à progresser. Mais en contrepartie, son utilisation se révèle laborieuse pour des projets avec un trop grand nombre de paramètres et d'éléments. Alors qu'un moteur de jeu comme *Unity*, est sans conteste beaucoup plus fastidieux à prendre en main, mais devient au final plus adapté à la réalisation de jeux plus ambitieux.

Afin que l'étape d'introduction reste accessible au plus grand nombre, la difficulté doit être progressive et adaptée au niveau de chacun. Dans l'idéal, les contraintes seraient assez souples pour encourager l'imagination de l'utilisateur, tout en préservant la satisfaction liée à sa progression.

Il est de même important de s'intéresser au matériel informatique qu'un outil nécessitera pour fonctionner, ainsi qu'à son prix. En effet, certains logiciels ne sont accessibles que sur des consoles de jeux spécifiques comme peut l'être *Dreams*, qui est disponible exclusivement sur *Playstation 4*, ce qui représente un coût non négligeable. Ceci devient particulièrement problématique si nous réfléchissons à l'utilisation de cet outil dans le cadre d'un atelier ayant lieu dans un établissement qui met déjà son propre matériel à disposition. Pour la plupart des structures, ce genre d'investissement est moins intéressant qu'une tablette ou un ordinateur, qui seront plus polyvalents.

D'autres développeurs de logiciels de création comme *Construct 3*, *Scratch*, *Flowlab*, *Sketch Nation*, *Bitsy* ou *Sploder*, ont fait le choix d'opter pour une plate-forme plus universelle comme le web. Les outils accessibles via un navigateur sont souvent gratuits et prêts à l'utilisation sans avoir besoin d'être téléchargé, cette option reste la plus économique et facile à mettre en place lors d'ateliers. Ces outils ont également l'avantage de proposer un stockage en ligne des projets, ce qui permet de les transférer et partager facilement, voire de collaborer à plusieurs sur le même projet. D'ailleurs, les jeux produits à l'aide de ces logiciels web bénéficient des mêmes avantages.

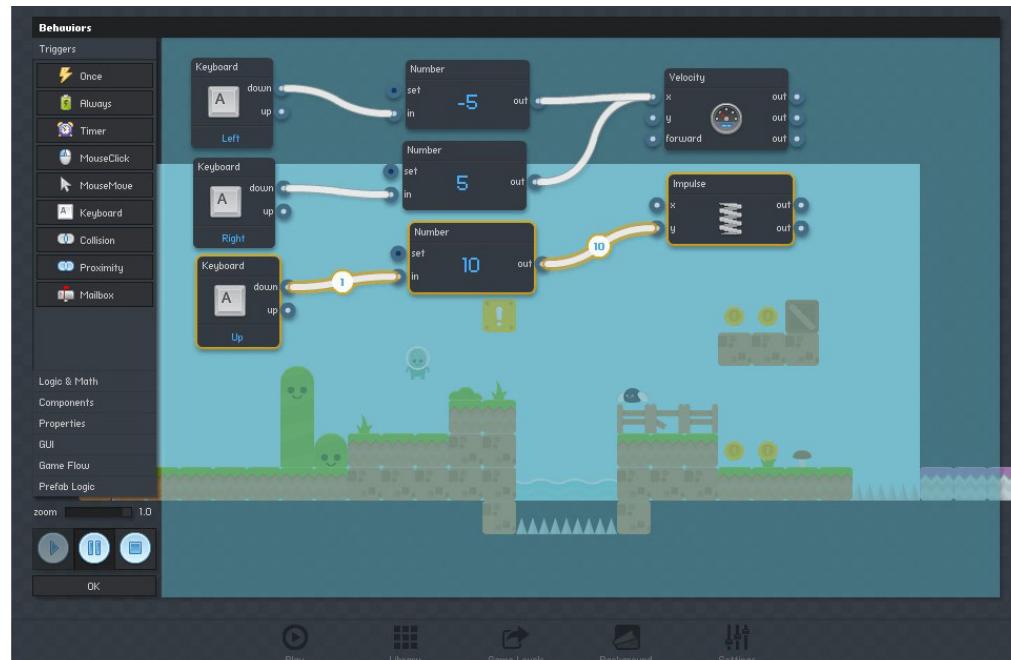
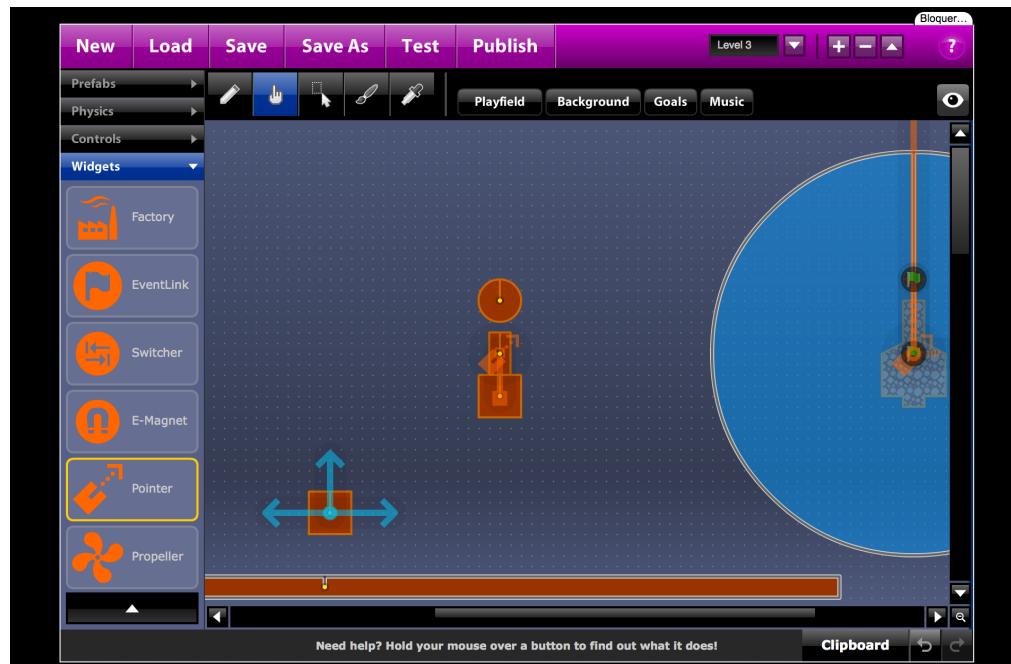


Fig. 45 Capture d'écran de l'interface de Sploder

Fig. 46 Image issue de Flowlab

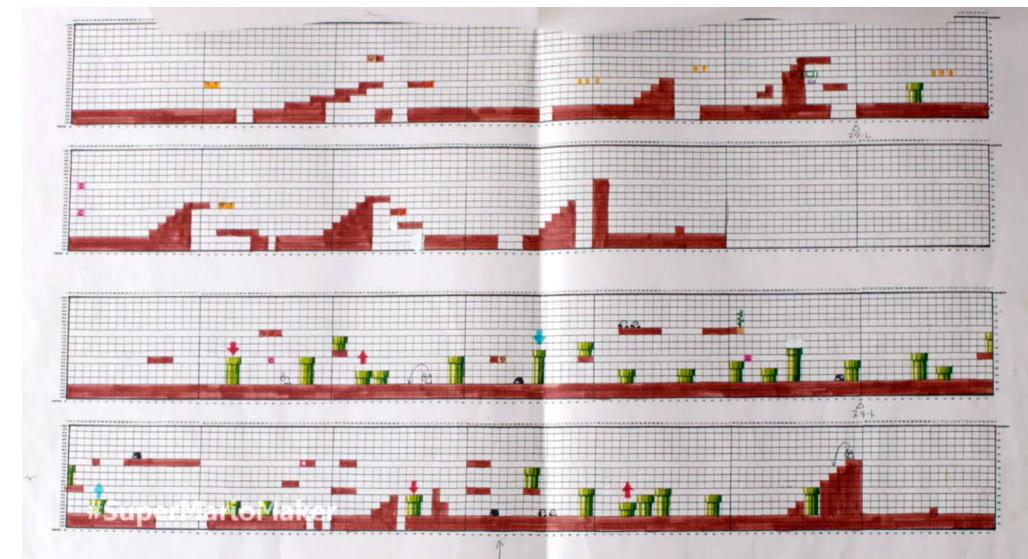


Fig. 47 Feuille utilisée pour concevoir des niveaux de Super Mario Bros.

Fig. 48 Sega Digitizer System, outil utilisé pour créer des éléments graphiques des jeux Mega Drive.

Interface et interaction

Ce critère vise à mettre en évidence les moyens mis en oeuvre par les développeurs de ces outils afin de les rendre plus abordables par un public néophyte. Cela peut se traduire par une attention particulière donnée à la hiérarchisation des éléments visibles à l'écran, la création d'un paradigme d'interaction inédite ou en s'appuyant sur des actions familières¹⁵ telle que le *drag'n'drop*.

Par exemple, les logiciels basés sur un système de *tiles* (ou tuiles en français) peuvent faire référence au coloriage. En effet, ceux-ci proposent à l'utilisateur de remplir une grille au moyen d'un "pinceau" qui dépose non pas de la couleur, mais des "objets de jeu". Cette manière de produire basée sur la composition d'éléments dans une matrice est d'ailleurs la méthode généralement employée par les professionnels du jeu vidéo pour réaliser des jeux en 2 dimensions¹⁶ sur les consoles n'étant pas assez puissante pour afficher des objets en 3 dimensions. Il est alors intéressant de voir un éditeur comme *Nintendo*, avec son jeu *Mario Maker*, mettre à disposition du grand public un outil qu'il aurait sans doute gardé en interne des années plus tôt.

Certains concepteurs choisissent quant à eux, une approche hybride, en associant des éléments tangibles et numériques. Ainsi, il est possible d'exploiter le potentiel d'une tablette en utilisant ses différents capteurs afin d'augmenter les possibilités d'un matériel familier tel qu'une feuille a4. Par exemple, les applications développées par *Pixel Press* se sont spécialisées dans la génération d'éléments virtuels en partant de dessins ou de pièces physiques grâce à l'utilisation d'algorithmes de reconnaissance d'images.

D'un point de vue pédagogique, il semble que manipuler des objets physiques soit une approche efficace pour l'apprentissage. Ce constat est d'ailleurs au cœur de la méthode Montessori. Cela permet non seulement de matérialiser le processus de production en gardant des traces de l'acte de création, mais aussi d'utiliser les écrans de manière moins intensive. Ce qui en fait un argument non négligeable pour organiser un atelier de création de jeux vidéo dans un établissement scolaire par exemple.

Personnalisation et représentation

Ce critère se concentre d'une part, sur les possibilités offertes par l'outil permettant aux utilisateurs de créer des éléments inédits qui participent à l'expérience esthétique du joueur. Et d'autre part, à la profondeur et aux limites de l'outil en termes de réalisation et quels effets peuvent produire ces restrictions, c'est à dire, jusqu'à quel point l'utilisateur, indépendamment de ses compétences, à la liberté de créer ce qu'il souhaite ou est contraint de produire des jeux "attendus" par l'outil.

Communauté et jeux produits

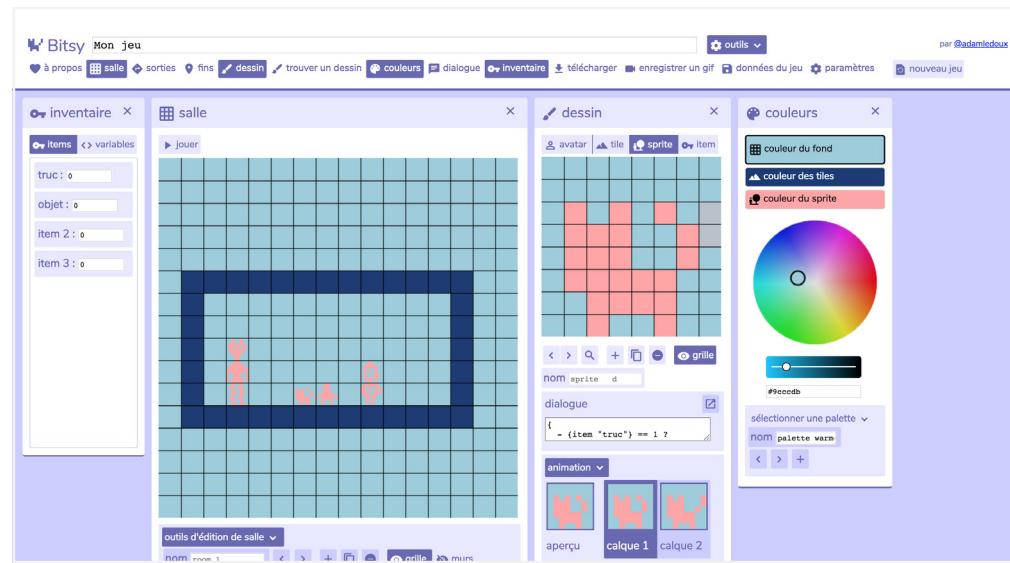
Ce point vise à répondre aux questions telles que "quel type de jeux est-il possible de produire avec cet outil" ou bien "Comment sont-ils partagés ?". Par exemple, certains outils tels que *Bitsy* ou *My Doodle Game* sont conçus pour réaliser un certain type de jeux précis, alors que d'autres permettent d'obtenir des résultats extrêmement variés comme *Unity* et *Construct*.

Une fois terminé, un créateur a selon l'outil qu'il a utilisé, plusieurs manières de partager son jeu. Avec un moteur de jeu classique comme *Unity*, il est possible d'exporter son jeu vers n'importe quel support ou navigateur web. Alors que les jeux réalisés avec certains outils tels que *Bloxel*, *Sketch Nation*, *Scratch* ou *Dreams* ne peuvent pas être joués en dehors de leur plateforme de distribution dédiée et généralement intégrée au sein de leurs logiciels. Cela peut sembler être un désavantage, mais ces plateformes de partage permettent surtout la formation de communauté de créateur se réunissant autour de certains outils. Le cas de *Bitsy* est particulièrement intéressant, car l'outil est hébergé sur *itch.io*, une plateforme de partage déjà existante et fréquentée par de nombreux utilisateurs qui sont devenus avec le temps la principale source de créateur réalisant et partageant des jeux fabriqués avec *Bitsy*.

¹⁵ Bogost, Ian . *Persuasive Games: Familiarity, Habituation, and Catchiness* [Article]

¹⁶ Video Game Densetsu. vgdensetsu.tumblr.com [Blog]

Bitsy Game Maker



Présentation

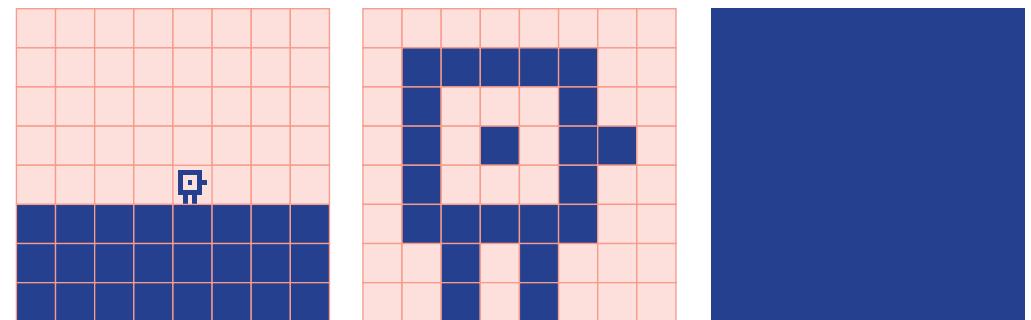
Bitsy est un mini outil de création développé par Adam Ledoux en 2017 qui permet de concevoir des mini jeux, des expériences narratives et des récits interactifs. Dans un article de Rock Paper Shotgun¹⁷, il nous explique les motivations derrière ce projet: il était frustré de perdre du temps sur les détails techniques lors de la réalisation de ses propres jeux. Il a donc décidé de créer son propre moteur de jeu afin de se concentrer sur ce qui lui paraissait essentiel, à savoir la création d'un univers et raconter une histoire.

Accessibilité

L'outil est disponible gratuitement sur n'importe quel navigateur web sans avoir à télécharger le moindre logiciel. Cela est rendu possible grâce au parti pris technique d'en faire un outil minimaliste offrant des performances graphiques extrêmement réduites, ce qui rend son utilisation possible sur un ordinateur peu puissant.

¹⁷ Dixon, Adam. *How small game makers found their community with Bitsy* [Article]

En effet, Bitsy est basé sur un système de grille composée de *tiles* ou briques. Chaque scène, est composée de 16x16 briques, et chaque brique fait 8x8 pixels. La palette de couleurs est limitée à trois choix de teintes: la première pour le fond, la deuxième pour les décors et la troisième pour les entités avec lesquelles il est possible d'interagir. Le nombre réduit de paramètres à régler dans Bitsy et le fait qu'il ne nécessite aucune connaissance en programmation ou en création graphique permettent de prendre en main l'outil en quelques minutes et le maîtriser en quelques heures.



Interface et interaction

L'interface est simple, épurée et extrêmement modulable. Il est possible de personnaliser son plan de travail en activant ou non les différentes fenêtres et les hiérarchisant dans l'espace. Cela permet de séquencer à son propre rythme l'apprentissage de ces différentes fenêtres/modules en maintenant son attention sur un nombre réduit de possibilités.

L'interaction concernant la création du contenu visuel et sa disposition dans l'espace peut être qualifiée de binaire. En effet, les pixels qui composent une brique sont soit remplis, soit vides. Cette particularité inédite par rapport à tous les autres outils que j'ai eu l'occasion d'essayer, à l'avantage de réduire les possibilités de l'utilisateur en le forçant à aller à l'essentiel, c'est à dire, créer une forme simple dans un espace restreint. Cette contrainte combinée à la limitation de la palette de couleur à 3 nuances permet de modifier très rapidement la teinte de tous les éléments présents dans une "scène".

Fig. 49 Interface de Bitsy Game Maker.

Fig. 50 Schéma explicatif du système de briques (une salle contient des blocs composés de pixels).

Les caractéristiques qui contribuent à rendre le processus de production plus immédiat sont, selon moi, très importantes dans un outil de création et bénéficie autant aux néophytes qu'aux utilisateurs de longue date, car lorsque l'on diminue le nombre d'étapes qui ne nécessitent pas ou peu de prises de décision de la part de l'auteur, cela a pour effet de privilégier chez celui-ci des réflexions d'avantages liées à la conception de son jeu au détriment de questions plus techniques ou d'optimisation pouvant être fastidieuse et chronophages si elles ne représentent pas un champ d'exploration en soi pour l'utilisateur.

Représentation et personnalisation

Bitsy a été conçu à l'origine pour créer des jeux basiques où l'on incarne un personnage qui déambule dans un univers représenté par une vue aérienne et fixe, où les seules interactions possibles avec les éléments qui constituent ces univers se produisent lors d'une collision avec ceux-ci. Ces interactions sont limitées au nombre de trois (lire/écouter un acteur, prendre un objet ou passer d'une scène à une autre) et diffèrent selon la nature de l'élément touché (décor, acteur, objet ou portail).

Communauté et jeux produits

L'une des raisons du succès de *Bitsy Game Maker* sur internet est due en grande partie à la communauté de créateurs qui postent régulièrement des jeux réalisés avec *Bitsy* sur la plateforme de partage/réseau social *Itch.io*. Le fait que les jeux produits avec cet outil puissent être contenus dans un fichier au format *HTML*, qui est le langage de programmation "par défaut" de tous les navigateurs web. Ce qui rend leurs diffusions extrêmement simples et ne nécessite donc aucun matériel particulier pour être joué.

Les jeux créés avec *Bitsy* sont facilement identifiables et différenciables d'autres jeux usant de la même esthétique pixel art. Ainsi, ils commencent tous par un encart textuel et partagent des contraintes visuelles telles que la taille de la fenêtre de jeu, son format carré, les animations limitées à deux images par secondes ou la palette restreinte de couleur. Mais également des contraintes mécaniques comme le fait de pouvoir déplacer son avatar uniquement d'une brique à une autre, et ce à l'intérieur d'espaces constamment limités à 16x16 briques qui ne permettent aucun mouvement de caméra, ou encore, l'absence de moteur gérant les phénomènes physiques tels que la gravité. Ces contraintes ont par exemple pour effet

de rendre impossible la création d'un certain type de jeu comme les *platformers*, genre qui se définit généralement par le point de vue de la caméra, qui se déplace avec le joueur et l'importance que prend l'action "sauter" dans ces jeux. Ainsi, *Bitsy* priviliege la réalisation d'objets faisant plutôt référence au jeu d'aventure tel que *Zelda*, ou plus particulièrement, *Adventure*, apparu en 1979 sur Atari 2600.

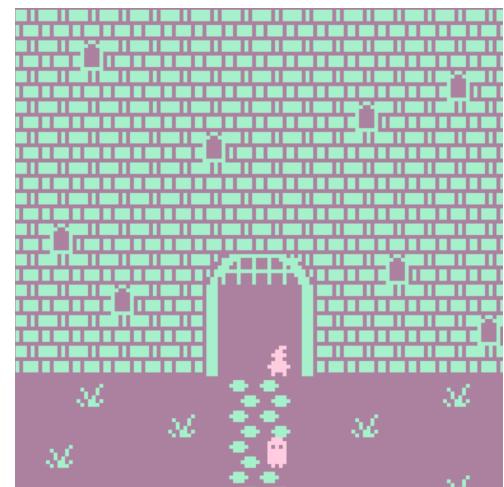
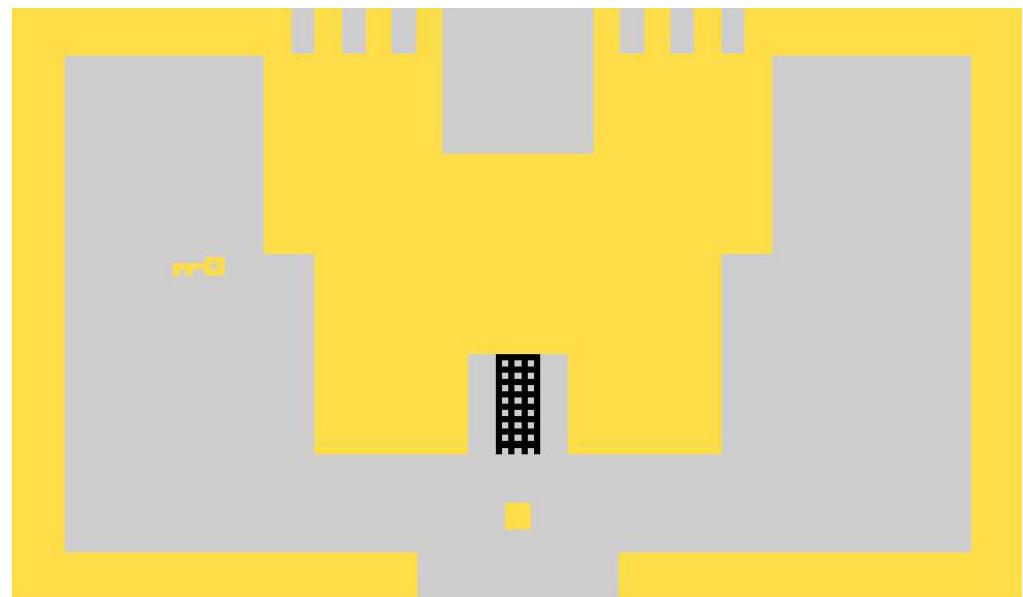


Fig. 51 Image provenant du jeu *Adventure*, créé en 1979 par Warren Robinett pour Atari.

Fig. 52 Image de l'un des encarts textuels qui sont présents à chaque démarrage d'un jeu Bitsy.

Fig. 53 Capture d'écran représentative du type de jeu qu'il est possible d'obtenir en utilisant Bitsy.

Malgré ces limitations, la communauté fait preuve d'une grande créativité et arrive à proposer des expériences souvent poétiques et inattendues, certains jeux se révèlent être très impressionnantes lorsque l'on a connaissance des contraintes de l'outil. Parfois, il s'agit simplement pour l'auteur d'explorer les limites techniques de *Bitsy* en proposant par exemple de nouvelles mécaniques ou moyens de représentation comme une vue isométrique ou des mouvements de caméra.

Cette observation met en lumière une activité pratiquée par certains créateurs qui vise à explorer, voire repousser, les limites d'un outil en exploitant ses contraintes. Cela contribue d'ailleurs à voir émerger de nouvelles "fonctionnalités" non prévues à l'origine par l'auteur de l'outil. Ainsi, je me demande si dans l'idéal, un développeur devrait tenir compte ou non des "découvertes" faites à propos de son outil afin de le faire évoluer pour faciliter l'utilisation de ces "fonctionnalités" officieuses. Ou de faire le choix, comme Adam Ledoux avec *Bitsy*, de ne pas proposer de nouvelles possibilités afin de ne pas prendre le risque d'altérer l'identité de son outil ou de le rendre potentiellement plus difficile à aborder.

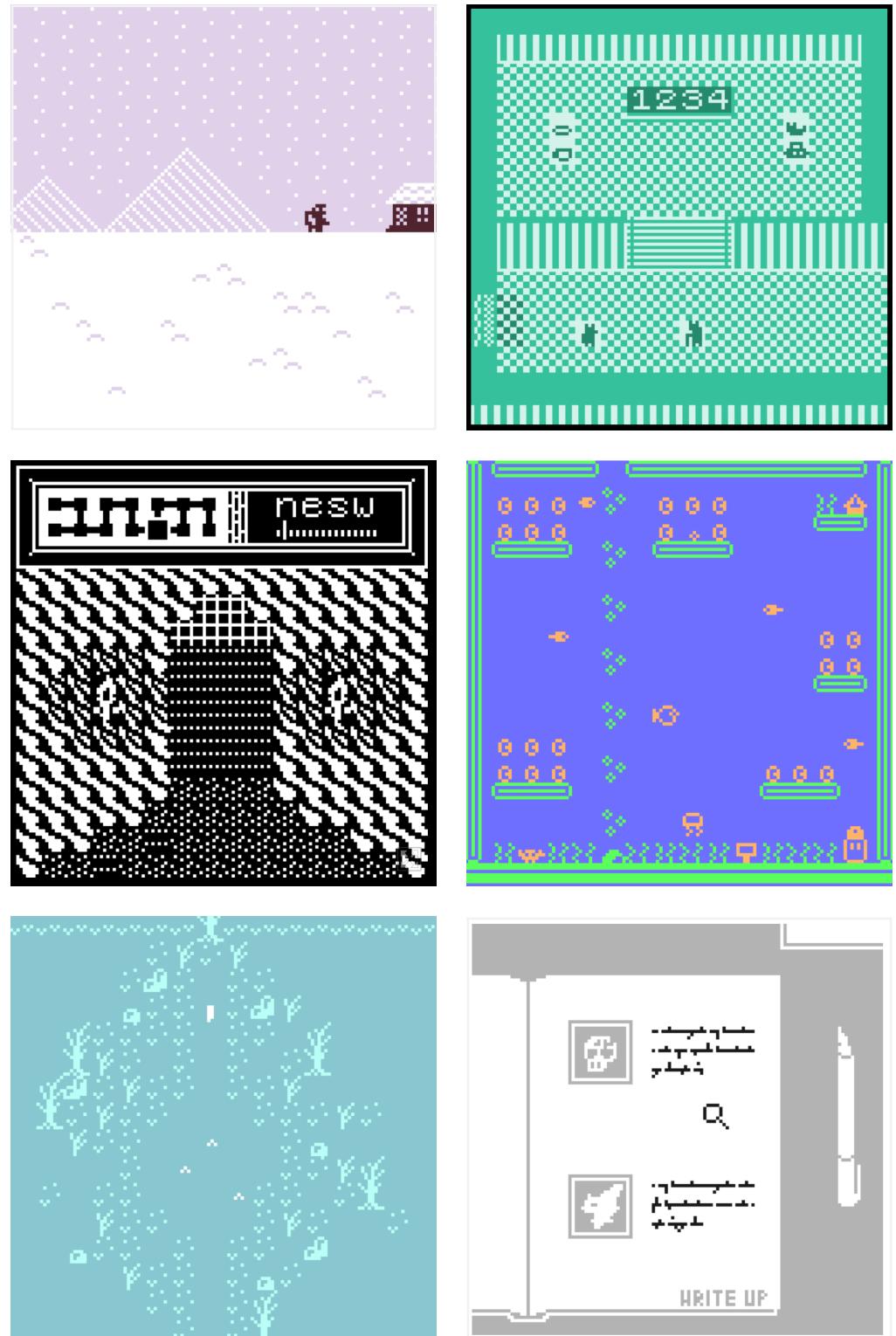
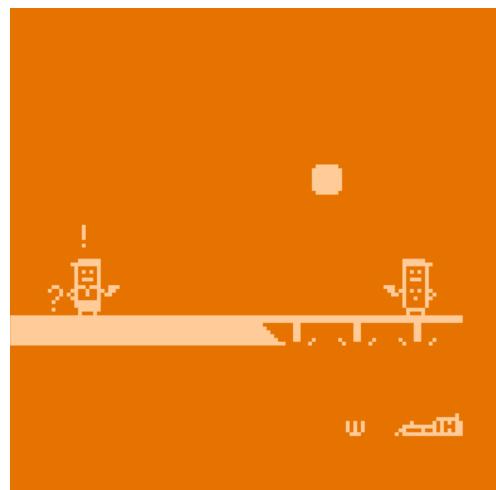
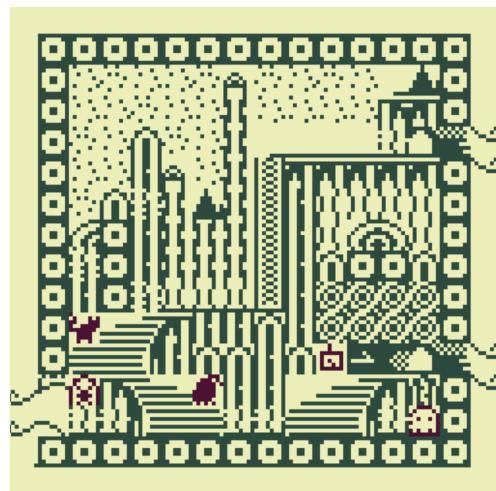
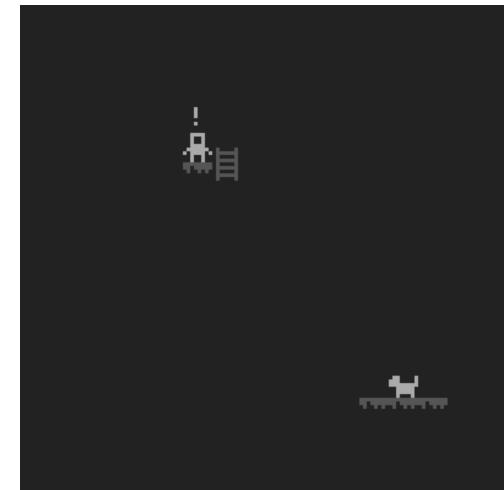
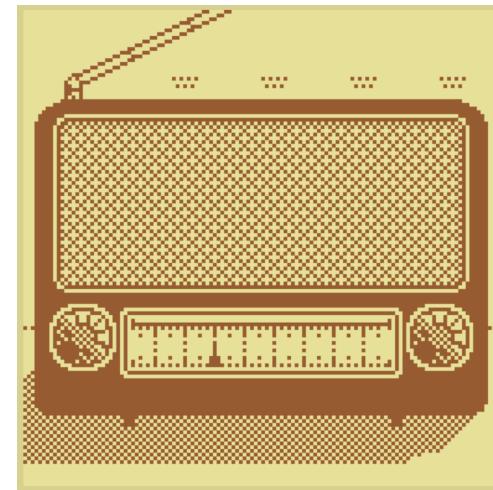
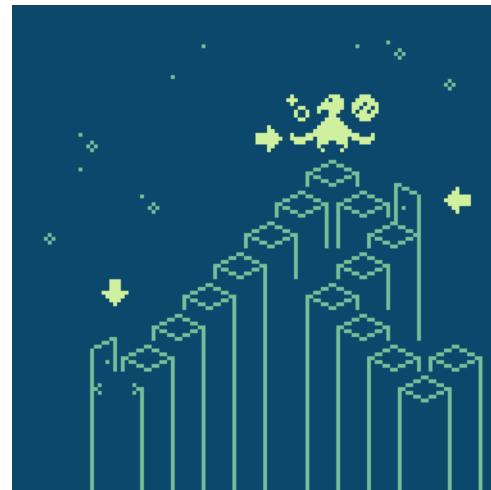


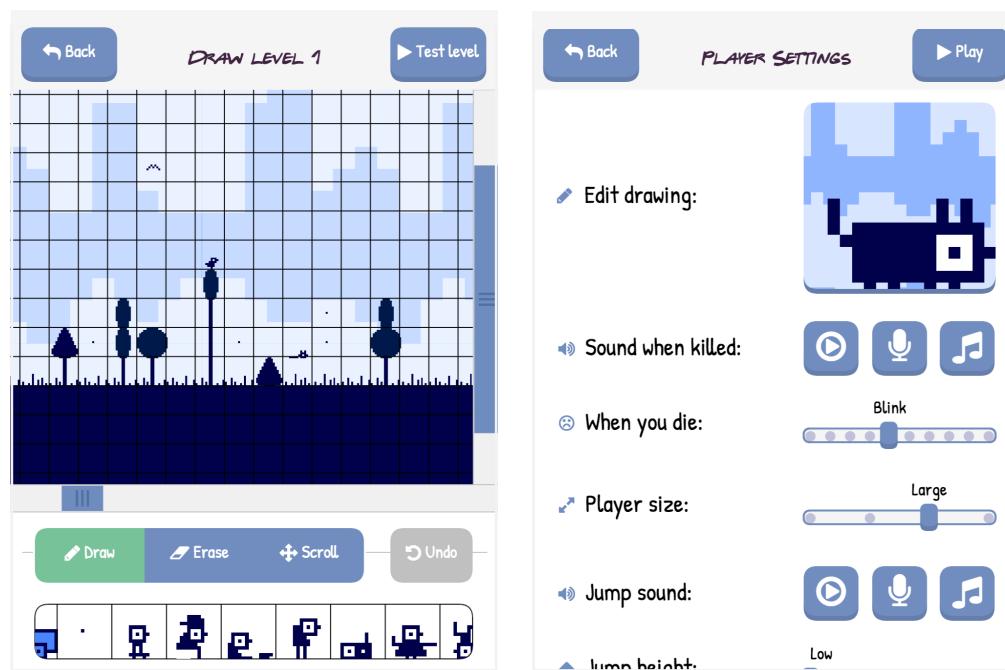
Fig. 54 - 71 Sélection en image de jeux créer sur Bitsy Game Maker.

Easy to learn, Hard to Master



Analyse

Sketch Nation



Présentation

Sketch Nation est une application qui permet de créer son propre mini jeu à partir de templates inspirés de jeux à succès comme, *Flappy Bird*, *Doodle Jump*, *Canabalt*, *Candy Crush* ou encore d'un genre connu tel que les jeux de plateforme. Les conditions d'analyse de cet outil sont un peu particulières, car celle-ci repose autant sur de nombreuses heures d'utilisation personnelle que d'observations lors de différents ateliers avec des participants néophytes.

Accessibilité

Cette application est disponible gratuitement sur tablettes et accessible depuis un navigateur web ce qui en fait un outil facile à utiliser lors d'ateliers. Les développeurs de *Sketch Nation* ont fourni un travail remarquable pour permettre

Fig. 72 Espace permettant de composer les niveaux à l'aide de blocs dans *Sketch Nation*.

Fig. 73 Interface visible lors de la création d'un personnage.

aux néophytes d'utiliser leur outil. En effet, avant de commencer à créer un jeu, l'utilisateur doit choisir un type de jeu avec lequel commencer, puis il lui sera demandé d'évaluer son niveau d'expérience en sélectionnant ce qui s'apparente à un "mode de difficulté" (simple, intermédiaire ou expert). Cette décision a pour conséquence de diminuer le nombre de possibilités en réduisant les choix de l'utilisateur à l'essentiel dans le mode simple. Alors qu'en mode expert l'utilisateur a logiquement à sa disposition toutes les possibilités offertes par l'outil, qui sont d'ailleurs très limitées si on les compare à des véritables moteurs de jeux comme *Unity* ou *Construct*.

Interface et interaction

La prise en main immédiate et intuitive de *Sketch Nation* est rendue possible en grande partie grâce à l'attention particulière que les développeurs ont portée à cet aspect. En effet, certains choix de conception de cet outil permettent de désamorcer l'aspect effrayant que peuvent revêtir d'autres outils plus complexes. Ainsi il est facile de constater que les éléments de l'interface sont plus volumineux que ceux observés dans d'autres outils, la police de caractère utilisée semble amicale et chaque paramètre est représenté par une icône sur laquelle il suffit d'appuyer pour en obtenir la signification.

Les interactions proposées dans *Sketch Nation* peuvent être séparées en plusieurs séquences d'actions. L'une d'elles est similaire à tous les outils qui sont construits sur un système de briques comme *Bitsy*, c'est à dire, sélectionner un outil/couleur/entité et remplir avec celles-ci une grille composée de cases. Cette manière de réaliser des jeux présente certains avantages, l'un d'eux est qu'elle permet de définir une unité de mesure absolue (la brique/case) qui facilite grandement la conception de niveaux et l'équilibrage des règles par exemple.

Une autre façon d'interagir passe par la modification des paramètres des différents éléments via des sliders. Ces sliders offrent un choix granulaire et les informations sont traduites de manière à être comprises facilement en substituant les valeurs abstraites contre des adjectifs contextualisés, ce qui correspond à une échelle de proportion plus représentative.

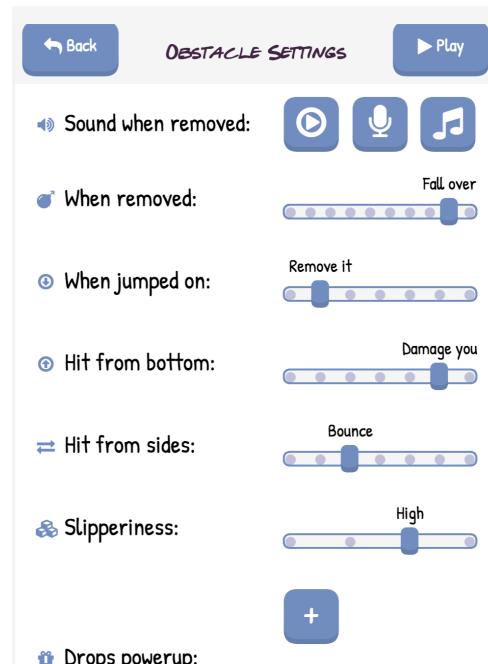
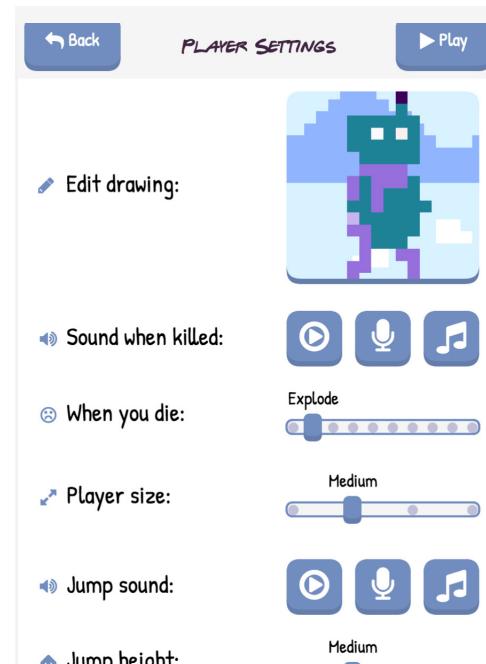
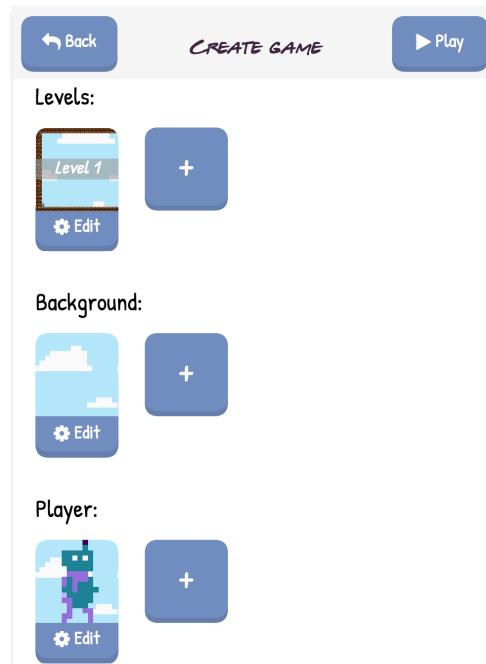
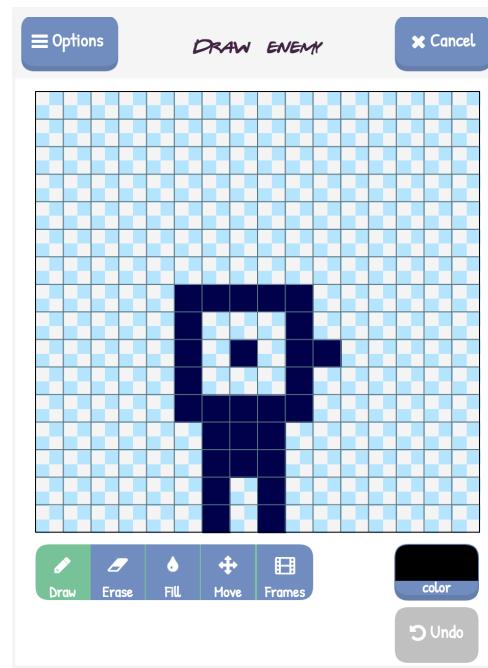
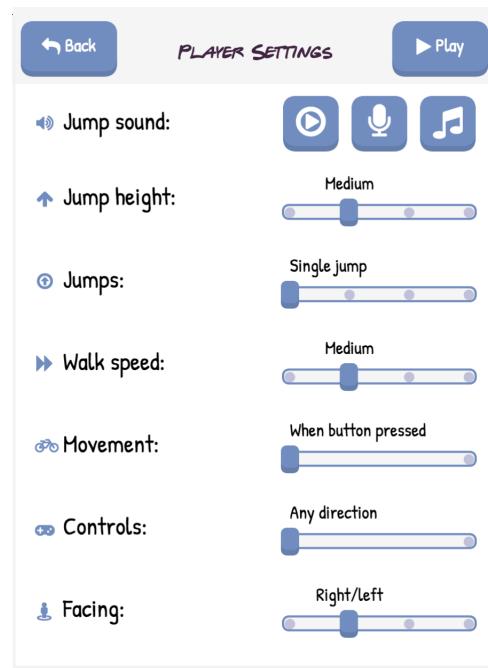
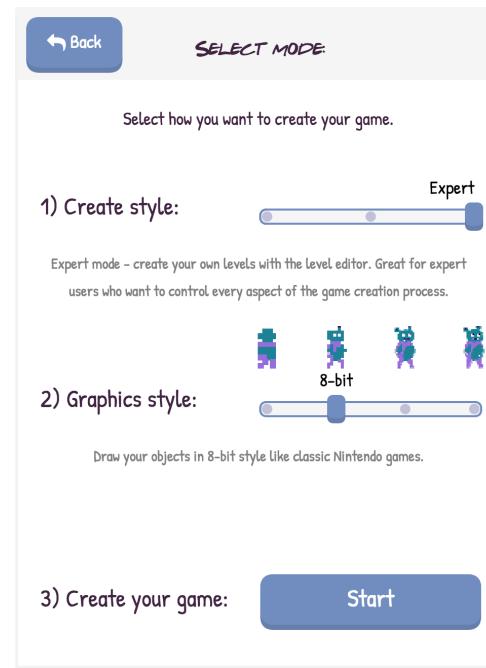
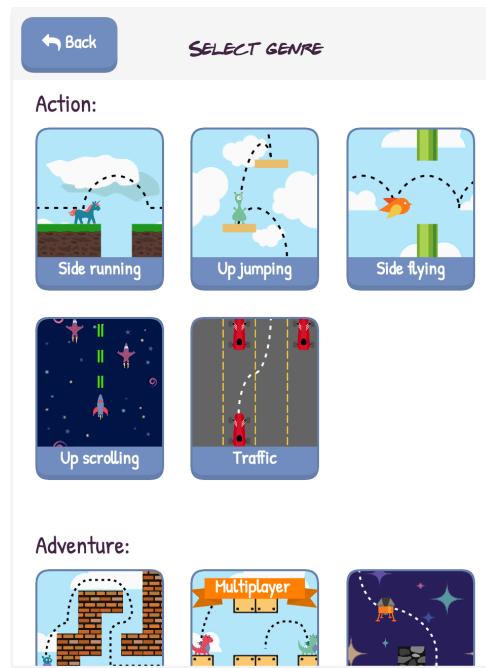


Fig. 74 Image illustrant le premier choix que l'utilisateur doit faire, choisir un modèle de jeu.
 Fig. 75 Fenêtre permettant de choisir le niveau de difficulté de l'outil et l'aspect graphique du jeu.
 Fig. 76 Éléments par défauts présents dans le template dédié aux jeux de plateformes.
 Fig. 77 Interface dédiée à la création de personnages.

Fig. 78 Image illustrant la modification des paramètres d'un personnage via des sliders.
 Fig. 79 Espace permettant de dessiner son personnage.
 Fig. 80 Fenêtre regroupant tous les éléments de jeu créés.
 Fig. 81 Les variables visibles sur cette image permettent de créer une grande variété de blocs.

Représentation et personnalisation

Sketch Nation ressemble davantage à un éditeur de niveau comparable à *Super Mario Maker* qu'à un logiciel plus complet comme *Construct* ou *Game Maker*. La similarité entre ces deux outils se vérifie notamment lorsqu'il est question de composer l'espace du jeu.

Cependant, contrairement à *Super Mario Maker*, où l'utilisateur est contraint d'utiliser les entités propres à l'univers de *Mario*, ayant un visuel et un comportement inaltérable, *Sketch Nation* nécessite de créer soi-même chaque élément (avatar, obstacles, acteurs, etc..), en passant aussi bien par son aspect esthétique que par le paramétrage de ces attributs fonctionnel dans le jeu.

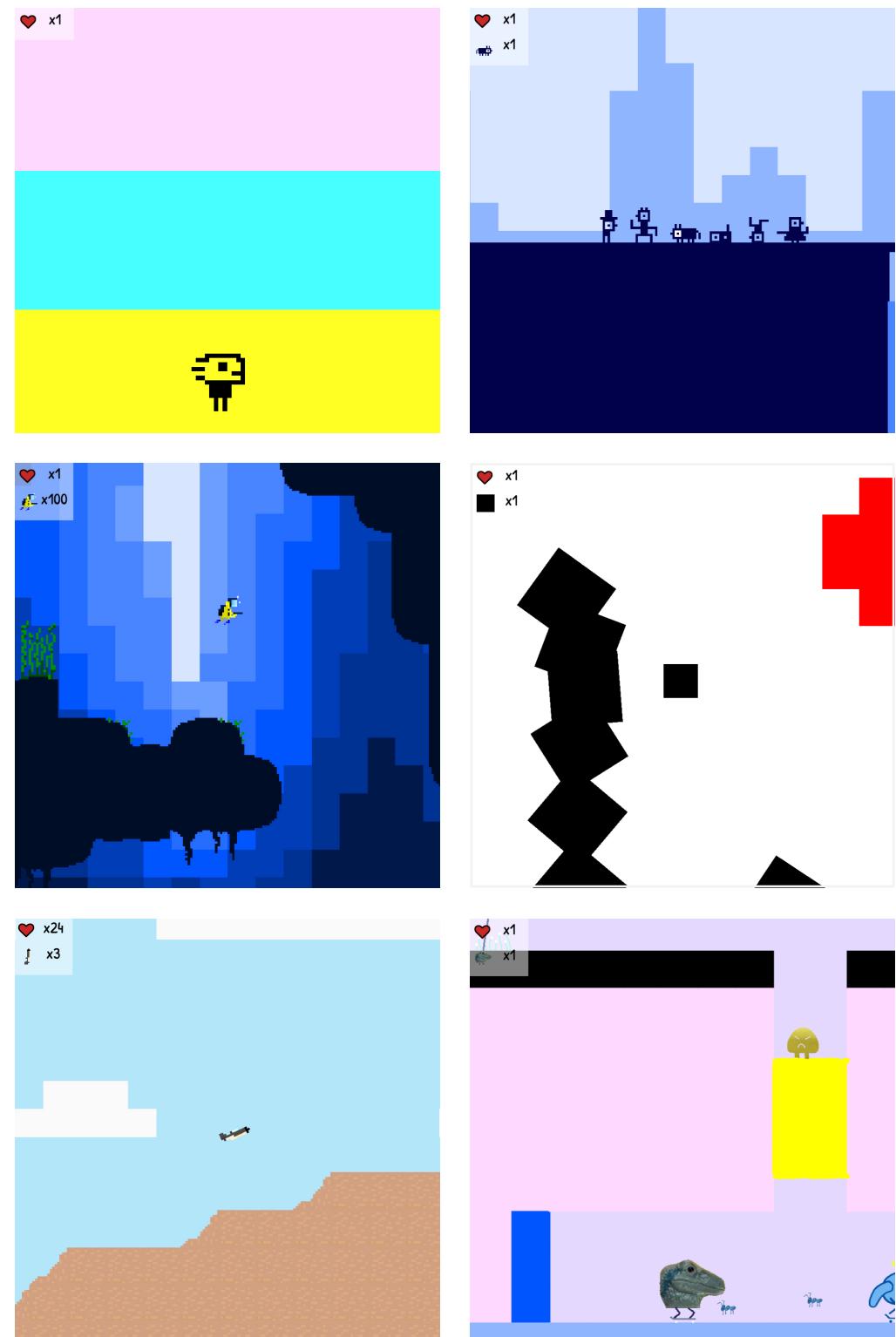
D'ailleurs, les manières de créer du contenu visuel dans *Sketch Nation* sont multiples. Il est par exemple possible d'importer des images, de prendre une photo directement depuis sa tablette ou sa webcam et l'intégrer en quelques secondes dans le jeu, ou bien d'utiliser l'outil de dessin inclus dans le logiciel. Concernant les sons, les fonctionnalités sont bien plus limitées, mais il est toutefois possible d'enregistrer un son directement depuis l'application. Ces caractéristiques contribuent à rendre possible l'obtention de résultats très immédiats avec cet outil et donc de l'utiliser lors d'ateliers ayant une durée très courte.

Communauté et jeux produits

Lorsque l'utilisateur est satisfait du jeu qu'il a créé, il peut le publier sur le portail de la communauté qui est intégrée dans l'application, être noté et recevoir des commentaires sur son travail afin de s'améliorer. En revanche, il lui est impossible de l'exporter ou le partager nativement sur des plateformes comme *Itch.io* ou *Game Jolt*.

Bien que les jeux produits par la communauté présente dans *Sketch Nation* dépendent fortement du canvas servant de modèle de base qui est choisi par l'utilisateur. Les auteurs font preuve d'une grande créativité et cela se vérifie dans la quasi-totalité des jeux que j'ai eu l'occasion d'essayer. Ils présentent en effet une expérience ou un point de vue à chaque fois différents. Originalité que j'ai pu également observer durant les ateliers que j'ai organisés.

Fig. 82 - 87 Exemple en images de jeux produits avec *Sketch Nation*.



Scribblenauts



Présentation

Scribblenauts est un jeu vendu sur consoles, PC et tablettes, dont la mécanique principale est axée sur la création de contenu par l'utilisateur. Je m'y intéresse, car les développeurs ont néanmoins mis en place des moyens d'action particuliers et des outils d'édition très complète qui méritent à mon sens d'être abordés dans ce mémoire.

Il s'agit donc d'un jeu de réflexion basé sur la créativité et la découverte de solutions émergentes destiné à un public relativement jeune. Le jeu nous invite à aider ses habitants au moyen d'un carnet magique qui fait apparaître les objets et entités qui y sont inscrits textuellement. Ainsi, il est possible par exemple d'invoquer un être vivant et de lui donner des attributs physiques et comportementaux ou bien de redéfinir un élément existant à condition que ces attributs figurent dans la base de données.

Fig. 88 Interface visible lorsque l'on souhaite interagir avec un élément dans Scribblenauts.

Accessibilité

L'approche qui est faite pour enseigner au joueur ses mécaniques est très classique, et ce via un tutoriel sous forme de quête qui met en scène un personnage non jouable qui explique pas à pas au joueur quels seront ses pouvoirs dans le jeu. Ensuite, il sera offert au joueur une extrême liberté pour progresser et sera libre de mener son aventure au rythme qu'il le souhaite sans aucune aide. En effet dans Scribblenauts, il existe d'innombrables façons de résoudre les énigmes. Par exemple, si le joueur fait la rencontre d'un personnage visiblement triste à côté d'un arbre avec un chat à son sommet, l'une des solutions logiques sera de créer une échelle ou de faire apparaître un pompier, mais il est tout à fait possible de réduire la taille de l'arbre ou encore de rajouter des ailes au chat, voir de trouver une solution encore moins conventionnelle.



Fig. 89 Image illustrant l'exemple du chat dans l'arbre.



Fig. 90 Fenêtre visible lorsque créer un élément ou modifier les attributs d'une entité existante.
Fig. 91 Interface permettant de modifier les propriétés d'un élément.

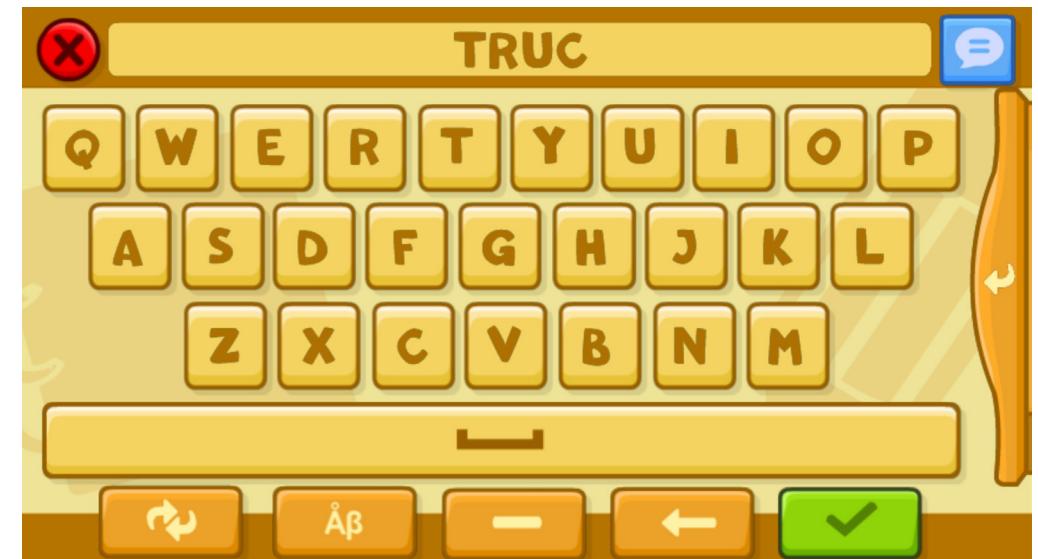


Fig. 92 Image illustrant le moyen de faire apparaître ou altérer du contenu dans Scribblenauts.
Fig. 93 Fenêtre visible lorsque l'entité ou l'adjectif n'est pas présent dans le dictionnaire du jeu.

Interface et interaction

Les types d'interactions proposées au joueur sont limités et relativement originaux. Ainsi, le moyen d'action principal du joueur s'effectue à travers l'écriture. Le jeu intègre toutefois un éditeur visuel lors de la création de nouvelles entités. Bien documenté, cet éditeur permet de s'approprier le fonctionnement lexical du jeu en permettant à ses utilisateurs de programmer simplement des événements sur la base du vocabulaire inclus dans le dictionnaire du système. Par exemple, il est possible de paramétriser le comportement d'un acteur afin qu'il réagisse à des événements basiques tels que "si je suis mouillé, alors je deviens méchant envers les enfants". Cette manière de définir des conditions sur le modèle "si ceci alors cela" permet en les accumulant, de créer un véritable petit écosystème sans aucune connaissance en programmation.

Représentation et personnalisation

La mécanique du jeu exploite une personnalisation de son contenu particulièrement profonde qui réside sur la description textuelle des éléments présents dans le jeu. Ainsi, il est offert au joueur la possibilité de créer de nouvelles entités et de les enregistrer dans la base de données. La cohérence graphique est garantie par l'utilisation exclusive d'éléments déjà existants qu'il faudra ensuite utiliser pour composer du contenu inédit.

Jeux produits et communauté

Scribblenauts ne permet pas de créer de jeu. En revanche, il est possible de créer des entités dont il est possible de définir le comportement. Ces éléments peuvent être mis à la disposition d'autres joueurs via le *Steam Workshop*, l'une des plateformes les plus populaires de partage de contenu généré par les utilisateurs dans le domaine du jeu vidéo.



Fig. 94 Interface permettant de créer des règles qui vont définir le comportement d'une entité.

Fig. 95 Liste d'une partie des comportements qu'il est possible de sélectionner.



Fig. 96 Fenêtre regroupant les éléments de jeu créer.
Fig. 97 Espace dédié à la création de l'aspect graphique d'une entité.



Fig. 98 Dans Scribblenauts il est également possible de créer des événements contextuels.
Fig. 99 Une partie des déclencheurs disponible.

Dreams



Présentation

Disponible en 2019 exclusivement sur *Playstation 4*, *Dreams* est un jeu/outil développé par le studio *Media Molecule* proposant deux types d'expériences. D'un côté, l'outil donne accès à un logiciel de création très complet avec lequel il est possible de réaliser son propre jeu ou uniquement des éléments séparés. De l'autre, les joueurs ont à disposition une infinité de jeux produits grâce aux utilisateurs qui investissent l'outil. D'ailleurs, l'aventure principale qui a été fabriquée par les développeurs de *Dreams*, a été entièrement construite avec exactement la même interface et les mêmes outils qui sont mis à disposition des utilisateurs. Cette particularité est à ma connaissance inédite dans l'industrie du jeu vidéo.

Étant donné que *Dreams* ne sera pas disponible sur le marché avant 2019, l'analyse s'est faite d'après de nombreux articles et vidéos qui circulent sur internet et plus particulièrement sur les réseaux sociaux du studio *Media Molecule*¹⁸.

Fig. 100 Image d'un jeu créé avec *Dreams*

Accessibilité

Un des enjeux de *Dreams* sera d'être suffisamment accessible à une partie des utilisateurs afin de leur permettre de générer du contenu régulièrement pour alimenter le catalogue de jeux disponible à l'autre partie des joueurs qui ne souhaiteraient pas se lancer dans la création. Lors d'une entrevue pour le journal *Polygon*¹⁹, les développeurs affirment avoir appris de leurs précédents projets, les séries des *Little Big Planet*, qui sont eux aussi des jeux/outils offrant des caractéristiques similaires à *Dreams*. Ils ont notamment prévu de créer eux-mêmes des vidéos visant à faciliter l'apprentissage de leurs outils.

La mesure la plus efficace mise en œuvre par *Media Molecule* semble être le système de partage et de remix du contenu créé. En effet, à l'instar du modèle popularisé par un outil comme *Scratch*, chaque élément créé et enregistré en ligne par un utilisateur est mis à disposition gratuitement à l'ensemble de la communauté. Un néophyte aura alors accès à un répertoire contenant d'innombrables éléments qu'il pourra se contenter d'assembler entre eux ou les examiner afin de comprendre comment ils ont été conçus par ces créateurs plus expérimentés.

En plus de son prix, le fait que *Dreams* ne soit pas disponible sur un ordinateur ou une tablette, rends son utilisation très compliquée lors d'un atelier, car il nécessiterait un investissement matériel trop spécifique et pas suffisamment polyvalent.

Interface et interaction

Dreams permet une approche de la création par tâtonnement. En effet, chaque élément peut être modifié au moyen d'une "carte" qui contient plusieurs paramètres relatifs à l'élément. Ces paramètres sont identifiables grâce à une icône et peuvent être facilement modulés en direct, le plus souvent via des sliders ou tableaux à 2 axes. En outre chaque paramètre bénéficie d'une entrée et d'une sortie par lesquelles il est possible de transmettre des valeurs depuis/vers n'importe quel autre paramètre d'une autre "carte", pour autant qu'elle puisse "accepter" le même type de valeur en entrée (integer, float, boolen, etc.). Cette manière de créer visuellement des interactions entre les différentes entités d'un jeu incite selon moi à l'expérimentation. Ainsi, dans une vidéo postée par *Media Molecule*²⁰, il est possible de voir l'un des développeurs s'amuser à associer la vitesse de déplacement de son avatar à la tête de lecture de la musique de fond, et ce en quelques secondes à peine.

¹⁸ *MediaMolecule. Dreams PS4 - PSX 2017 [Vidéo]*

¹⁹ *Sarkar, Samit. Media Molecule's Dreams makes me wonder if there's anything it won't let you create [Article]*

²⁰ *MediaMolecule. Music in Dreams [Vidéo]*

Au fil des vidéos, on peut constater que *Media Molecule* a fait évoluer son interface. Dans un premier temps, la palette d'outils est intégrée à l'environnement 3D et à la diégèse du jeu (le curseur du joueur faisant partie intégrante de la fiction et les éléments d'interfaces se déplacent relativement à sa position). Mais les développeurs ont finalement privilégié la lisibilité à l'immersion en optant pour une représentation plus frontale de l'interface qui sera donc ancrée à l'écran. Cependant, il est intéressant de mettre en lumière une particularité propre à *Dreams*. Les éléments logiques qui permettent de modifier les paramètres de chaque objet sont disposés dans l'espace 3D par l'utilisateur. C'est-à-dire que ces éléments ne sont pas tous accessibles depuis une même fenêtre en sur couche, ce qui aurait permis d'avoir une vision d'ensemble et hiérarchisée de chaque élément comme dans un outil traditionnel tel que *Unity*. Il s'agit là d'un choix intéressant et il faudra attendre que le jeu sorte pour voir si le fait de disposer les informations de cette manière facilite ou son utilisation.

Bien qu'étant manipulable uniquement avec une manette, *Dreams* a été conçu pour exploiter le *PlayStation Move*, (le périphérique d'entrées dédié à la réalité virtuelle de la *PlayStation 4*) et semble totalement compatible avec une logique de création en VR où les actions de l'utilisateur sont reproduites tel quel dans l'espace du jeu. Ce qui permet par exemple de s'appuyer sur des gestes faisant référence à de la sculpture. Ainsi, il semble que la manipulation d'objets dans un environnement en 3 dimensions soit plus intuitive et précise. Ceci grâce à l'utilisation d'un contrôleur dans chaque main munie de capteurs capable de suivre chacun de leurs mouvements et donc de les restituer fidèlement à l'écran. D'ailleurs, les développeurs ont prévu à terme de rendre *Dreams* compatible avec le casque de réalité virtuelle.

Il est intéressant de noter que le joueur ne contrôle jamais réellement un avatar, mais plutôt la représentation spatiale de ses propres mains dans cet environnement virtuel. Elles prennent la forme de gouttes d'eau qui peuvent prendre possession de certains objets ou personnages. Cette manipulation prend alors un aspect performatif, voire comique, car il ne s'agit plus seulement de déplacer son avatar, mais de l'habiter en contrôlant plus ou moins précisément certains de ses membres ainsi que son équilibre.



Fig. 101 Capture d'écran illustrant la possibilité d'associer des valeurs "complexes" entre elles.

Fig. 102 Image où l'on peut voir l'artiste, Men Lu, utiliser les contrôleurs du Playstation Move.

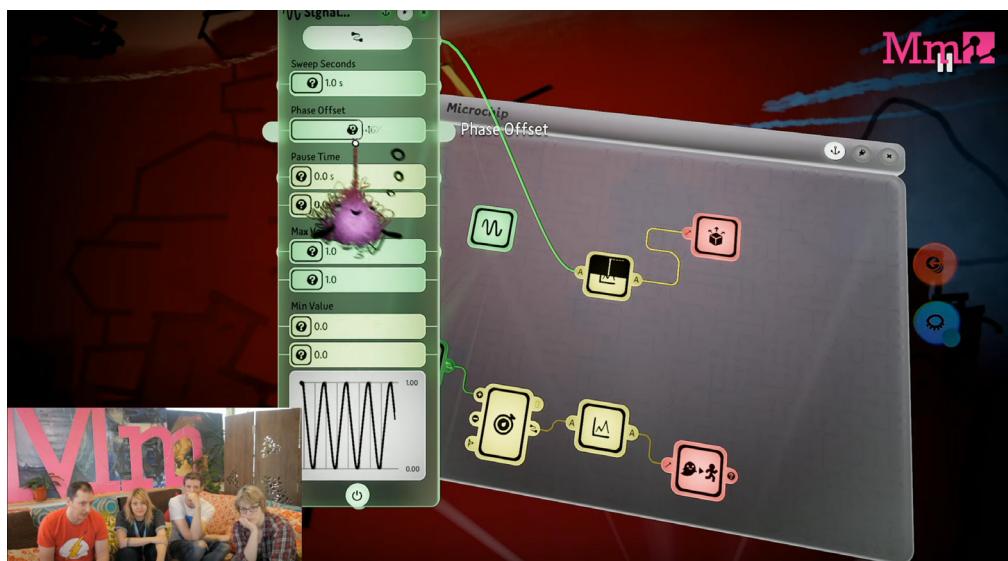


Fig. 105 Capture d'écran d'une vidéo présentant l'outil de création musicale intégré dans Dreams.

Fig. 106 Image détaillant la manière de créer des éléments logiques.

Fig. 107 Espace dédiée à la création de personnages.

Fig. 108 Image illustrant la disposition des éléments d'interfaces dans l'espace de création.

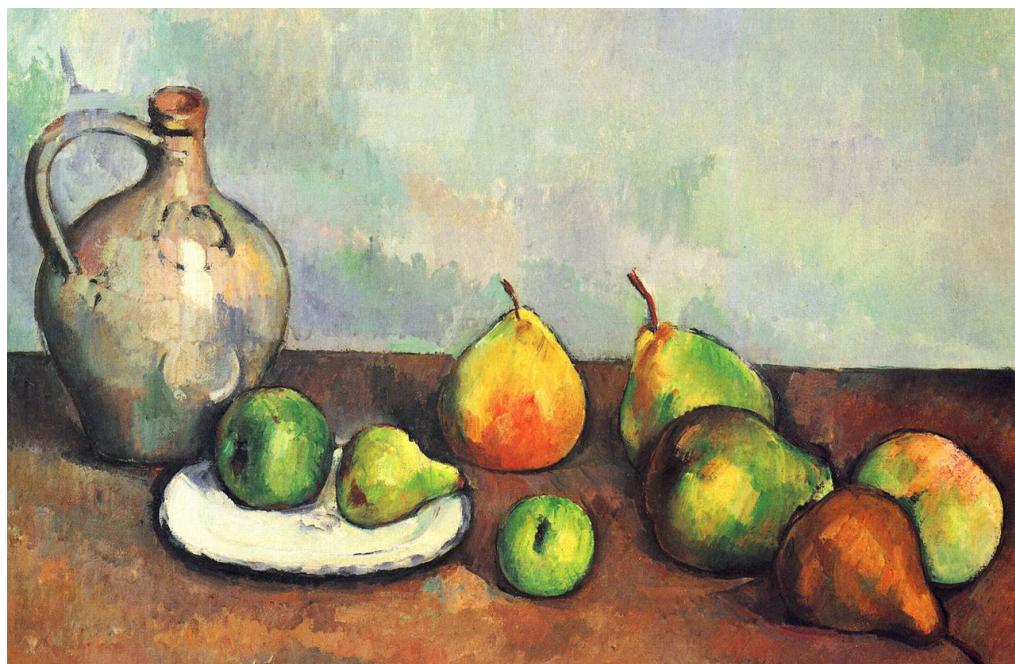
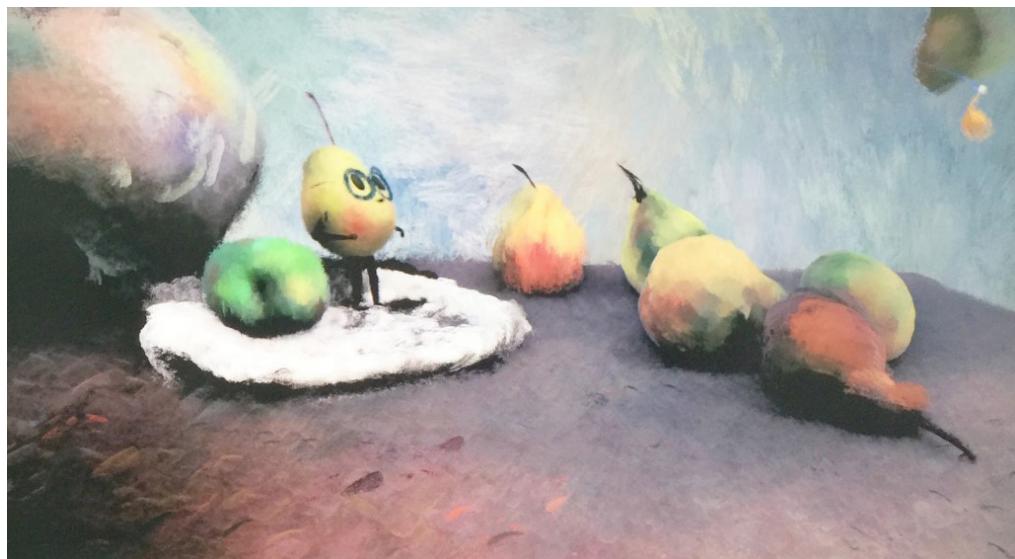


Fig. 107 Image issue d'un jeu créé avec Dreams, inspiré d'une nature morte de Paul Cézanne.

Fig. 108 Image de la nature morte de Paul Cézanne en question.

Représentation et personnalisation

Les jeux produits avec *Dreams* partagent entre eux une patte esthétique facilement reconnaissable. Cela s'observe notamment dans leurs arrières-plans ou plus précisément leur skybox (sorte de dôme qui englobe un univers en trois dimensions et qui représente l'horizon). En effet, elles ont toutes en commun un aspect cotonneux qui semble être le résultat d'une accumulation de taches de peintures en mouvements. Cela est dû à la technologie particulière qui est utilisée dans ce moteur de jeu, dont l'un des objectifs, d'après Kareem Ettouney, Directeur Artistique de *Dreams*, est de reproduire le style graphique de différents courants picturaux comme celui de l'impressionnisme par exemple²¹.

Communauté et jeux produits

Dreams pourra s'appuyer sur une communauté de créateur déjà présente, héritée d'une précédente licence de *Media Molecule*, *Little Big Planet*, qui proposait déjà un éditeur complet et une plateforme de partage inclus dans le jeu. Les développeurs, conscients qu'une partie des joueurs ne sentiront pas nécessairement le besoin de créer du contenu, proposent de parcourir histoire créer par leurs soins, mais également un mode qui permet de voyager aléatoirement de "rêve" en "rêve" à la manière d'un cadavre exquis, le jeu faisant en sorte qu'il y ait à chaque fois un élément qui les connectent entre eux.

Cet outil ne sera d'ailleurs pas limité à la création de jeux vidéo. Les développeurs l'ont conçu de sorte à offrir la possibilité de créer différents types de contenus comme des films d'animation ou des morceaux de musique. En effet, *Dreams* contient un logiciel complet dédié à la création musicale qui semble exploiter de manière pertinente les capteurs inclus dans le contrôleur de la *PlayStation 4*. Il est par exemple possible d'utiliser le micro interne pour enregistrer des sons, alors que le gyroscope et l'accéléromètre serviront à se positionner avec précision dans l'interface de création.

²¹ MediaMolecule. DreamsPS4 - Still life Art Creation with Kareem and Men Lu [Vidéo]

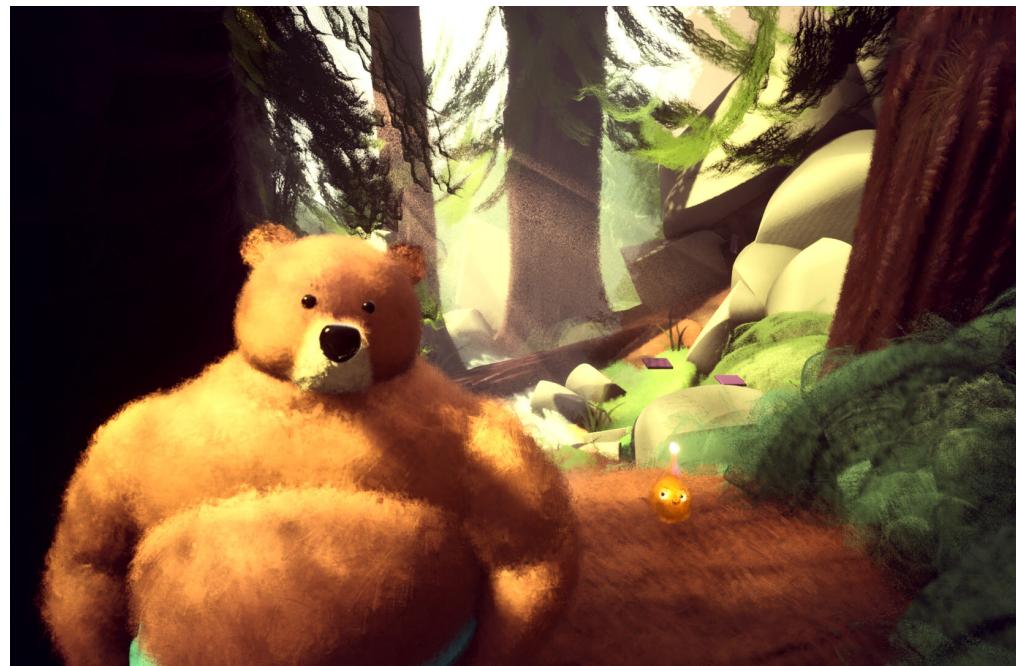


Fig. 109 - 112 Exemple en images de jeux produits avec Dreams.

Discussion

Contraintes et contextes

L'analyse de ces différents outils m'a permis de constater plusieurs choses. Il est intéressant de noter la tension existante au cœur de l'usage des outils de création de jeu vidéo à destination des néophytes. En effet, ceux-ci sont faciles à utiliser rapidement, mais plus difficiles à employer pour créer quelque chose de l'ordre de la maîtrise.

Les outils sélectionnés dans le corpus restrent ne sont, selon moi, pas assez complet pour permettre une courbe d'apprentissage débouchant sur de la maîtrise à long terme, à l'exception de *Dreams*. Cependant celui-ci n'étant pas encore disponible, je n'ai pas pu encore le tester par moi-même ou en cours d'atelier afin de déterminer s'il est aussi simple à prendre en main qu'il semble l'être. *Sketch Nation* semble rester l'outil à privilégier lors d'ateliers, car il permet à la fois une prise en main rapide et une liberté de production plus profonde qu'un outil similaire tel que *Bitsy Game Maker*.

Durant l'écriture de ce mémoire, je me suis également rendu compte de l'importance que peut prendre une communauté de créateurs investis lorsqu'ils ont à disposition une plateforme de partage pour diffuser leurs réalisations et communiquer entre eux. Ces utilisateurs deviennent alors la principale source de création de contenu, gratuite et inépuisable qui plus est. Certains passionnés s'occupent d'ailleurs spontanément de combler le travail des développeurs en proposant des tutoriels et autres contenus pédagogiques pour apprendre leurs outils sous forme d'articles, d'encyclopédie ou de vidéos via des plateformes de partages.

De même, les contraintes propres aux outils font partie intégrante de l'expérience proposée et qu'elles offrent aux créateurs un terrain d'exploration stimulant. De la même manière que pour les auteurs Oulipien (*Ouvroir de Littérature Potentiel*), l'activité qui consiste à interroger et explorer les limites d'un médium (ou d'un outil) au travers de ses contraintes, peut être une source d'intérêt en soi qui encourage le détournement et la découverte d'utilisation qui n'avaient pas été nécessairement prévues par les développeurs à l'origine, comportement qui s'observe facilement sur des outils très limités, mais suffisamment profonds tel que *Bitsy Game Maker* ou *Sketch Nation* notamment.

Un outil idéal

L'un des points qui me semble intéressant à aborder résulte de l'observation des participants durant les ateliers que j'ai menés et concerne l'investissement de l'auteur vis-à-vis de sa création, et particulièrement, les actions fournies par les utilisateurs et leurs cohérences avec les effets produits par l'outil.

Par exemple, on peut imaginer un outil hypothétique avec lequel on serait en mesure de générer un projet complet digne d'un jeu à gros budget aux mécaniques prédéfinies, comportant un grand nombre d'éléments préfabriqués et d'intelligences artificielles complexes. Et ce, en ne formulant uniquement que quelques mots clés à voix haute ou en répondant à une série de questions "Ok Google crée moi un jeu de tir pour l'anniversaire de..."

Ainsi, le résultat obtenu via un processus de fabrication excluant toute notion d'effort, d'apprentissage ou de maîtrise n'aurait à priori pas une grande valeur aux yeux de son auteur. À moins peut-être que le sentiment d'appropriation puisse provenir d'un certain nombre de choix cruciaux ou d'intentions exprimées par l'utilisateur qui serait cette fois représentatif d'un engagement vis-à-vis de sa création.

On peut alors se demander si l'une des caractéristiques d'un outil idéal réside dans le choix de ses contraintes et la manière d'accompagner l'utilisateur néophyte dans sa progression, en rendant accessible de plus en plus de possibilités et en affichant d'avantages d'éléments dans l'interface, au fur et à mesure qu'il maîtrise les différents modules de cet outil hypothétique.

Ces interrogations déclenchent inévitablement un certain nombre de questions qui resteront sans réponse dans ce mémoire, mais que je souhaite néanmoins énoncer ici, car elles représentent selon moi des pistes d'exploration et de réflexions pour la suite. On peut par exemple se demander: quelles sont ces étapes cruciales énoncées plus tôt ? Ces étapes sont elles uniquement liées à des questions de design ou de communication ? Quelles sont les actions qui peuvent être automatisées ou laissées à la charge de la machine ? À quoi ressembleront les outils de demain et les nouveaux paradigmes de créations ? Seront-ils davantage axés sur la manipulation d'un vocabulaire de conception ?

Pour conclure, je souhaite désormais concevoir mon propre outil de création de jeux vidéo à destination d'un public néophyte.

Références

- ¹ Crockett, L. J. (2016, October). "Easy to Learn, Difficult to Master": Accessible Front-Ends to Challenging Science Assignments. In European Conference on Games Based Learning (p. 144). Academic Conferences International. [Livre]
- ² Csikszentmihalyi, M., & Csikszentmihalyi, I. (1975). *Beyond boredom and anxiety* (Vol. 721). San Francisco: Jossey-Bass. [Livre]
- ³ Brossard, M. (1999). *Apprentissage et développement: tensions dans la zone proximale*. Y. Clot (Éd.). Avec Vygotski. Paris: Éd. La Dispute. [Livre]
- ⁴ Grace, L. (2014). *Critical games: Critical design in independent games*. In Proceedings of DIGRA. [PDF]
- ⁵ Genvo, S. (2012). *Comprendre et développer le potentiel expressif*. Hermès, La Revue, (1), 127-133. [Livre]
- ⁶ Grine, Esteban, (2016). "Lieve Oma, les grands-mères sont plus cools que les trolls". Chroniques videoludiques. [Article en ligne]
- ⁷ Frasca, Gonzalo (2001). "The Sims: Grandmothers are cooler than trolls". Game Studies, 1(1). [Article en ligne]
- ⁸ Hurel, Pierre-Yve (2017). "Faire créer des jeux sur la migration à des jeunes concernés". Carnet de jeu. [Article en ligne]
- ⁹ Julian Alvarez, Damien Djaouti, Jean-Pierre Jessel, Gilles Methel, Pierre Molinier, "Morphologie des jeux vidéo", Hermes science, Lavoisier, 2007" [PDF]
- ¹¹ Brett, Victor (2013). *Stop Drawing Dead Fish*. <<https://vimeo.com/64895205>> [Vidéo]
- ¹² Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.. [Livre]
- ¹³ Ludoscience (2011) Game Creation Tool Classification. <<http://creatools.gameclassification.com/FR/creatools/index.html>> [Site web]
- ¹⁴ Djaouti, Damien (2011) "Serious Game Design Considérations théoriques et techniques sur la création de jeux vidéo à vocation utilitaire". Université Toulouse III Paul Sabatier. [Thèse]
- ¹⁵ Bogost, Ian (2009). "Persuasive Games: Familiarity, Habituation, and Catchiness". Gamasutra. [Article en ligne]
- ¹⁶ Video game Densetsu (2018). <https://vgdensetsu.tumblr.com> [Blog]
- ¹⁷ Dixon, Adam (2018). "How small game makers found their community with Bitsy". Rockpapershotgun. [Article en ligne]
- ¹⁸ MediaMolecule (2017). *Dreams PS4 - PSX 2017*. <<https://www.youtube.com/watch?v=OwNpPoK2PgY>> [Vidéo]
- ¹⁹ Sarkar, Samit (2018). "Media Molecule's Dreams makes me wonder if there's anything it won't let you create". Polygon. [Article en ligne]
- ²⁰ MediaMolecule (2018). *Music in #DreamsPS4*. <<https://www.youtube.com/watch?v=E3BPsKb8p6I&t=49m30s>> [Vidéo]
- ²¹ MediaMolecule (2016). *DreamsPS4 - Stil life Art Creation with Kareem and Men Lu*. <<https://www.youtube.com/watch?v=YJaptH6TEw>> [Vidéo]

Crédits iconographiques

Pages 8 - 9	Pages 20 - 21	Fig. 38 EPFL. Tyhmio VPL. 2011	Fig. 58 Jeu créer avec Bitsy. [capture d'écran]
Fig. 1 <i>Gulizia, Ivan. Schéma illustrant la loi de Bushnell.</i> 2018	Fig. 18 <i>In Albon, Elias. Prototype du jeu.</i> 2018 [capture d'écran]	Fig. 39 Pixel Press. Bloxels. 2016	Fig. 59 Jeu créer avec Bitsy. [capture d'écran]
Fig. 2 <i>Gulizia, Ivan. Représentation schématique de la théorie du flow.</i> 2018	Fig. 19 <i>In Albon, Elias. Prototype du jeu.</i> 2018 [capture d'écran]	Fig. 40 Microsoft. Kodu. 2009	Pages 48 - 49
Pages 12 - 13	Fig. 20 <i>In Albon, Elias. Prototype du jeu.</i> 2018 [capture d'écran]	Fig. 41 Miyamoto, Shigeru. Tezuka, Takashi et al. <i>Super Mario Maker.</i> Nintendo. 2015	Fig. 60 Jeu créer avec Bitsy. [capture d'écran]
Fig. 3 <i>Unity Technologies. Unity 2018.</i> 2018. [capture d'écran]	Fig. 21 <i>In Albon, Elias. Prototype du jeu.</i> 2018 [capture d'écran]	Fig. 42 Pixel Press. <i>Adventure Time: Game Wizard.</i> 2013	Fig. 61 Jeu créer avec Bitsy. [capture d'écran]
Pages 14 - 15	Fig. 22 <i>In Albon, Elias. Prototype du jeu.</i> 2018 [capture d'écran]	Fig. 43 <i>Gulizia, Ivan. Tableau d'analyse 1re partie.</i> 2018	Fig. 62 Jeu créer avec Bitsy. [capture d'écran]
Fig. 4 <i>Veltman, Florian. Lieve Oma.</i> 2016. [capture d'écran]	Fig. 23 <i>In Albon, Elias. Prototype du jeu.</i> 2018 [capture d'écran]	Fig. 44 <i>Gulizia, Ivan. Tableau d'analyse 2e partie.</i> 2018	Fig. 63 Jeu créer avec Bitsy. [capture d'écran]
Fig. 5 <i>Hurel, Pierre-Yves. Photo issue de son blog personnel.</i> 2017	Fig. 24 <i>In Albon, Elias. Dessins de recherches.</i> 2018	Fig. 45 <i>Sploder. Interface de Sploder.</i> 2007	Fig. 64 Jeu créer avec Bitsy. [capture d'écran]
Pages 16 - 17	Fig. 25 <i>In Albon, Elias. Dessins de recherches.</i> 2018	Fig. 46 <i>Flowlab. Interface de Flowlab.</i> 2012	Fig. 65 Jeu créer avec Bitsy. [capture d'écran]
Fig. 6 <i>Game Moutier. Moon Does Stress.</i> 2016. [capture d'écran]	Fig. 26 <i>In Albon, Elias. Dessins de recherches.</i> 2018	Fig. 47 Miyamoto, Shigeru. Tezuka, Takashi et al. <i>Level design sheet for Super Mario Bros.</i> Nintendo. 1984	Fig. 66 Jeu créer avec Bitsy. [capture d'écran]
Fig. 7 <i>Game Moutier. Moon Does Stress.</i> 2016. [capture d'écran]	Fig. 27 <i>In Albon, Elias. Dessins de recherches.</i> 2018	Fig. 48 Sega. <i>Photo d'une personne utilisant le Sega Digitizer System III.</i> 1986	Fig. 67 Jeu créer avec Bitsy. [capture d'écran]
Fig. 8 <i>Game Moutier. Moon Does Stress.</i> 2016. [capture d'écran]	Fig. 28 <i>In Albon, Elias. Dessins de recherches.</i> 2018		Fig. 68 Jeu créer avec Bitsy. [capture d'écran]
Fig. 9 <i>Game Moutier. Moon Does Stress.</i> 2016. [capture d'écran]	Pages 22 - 23		Fig. 69 Jeu créer avec Bitsy. [capture d'écran]
Fig. 10 <i>Game Moutier. Moon Does Stress.</i> 2016. [capture d'écran]	Fig. 29 <i>Gulizia, Ivan. Schéma illustrant le concept de briques de gameplay.</i> 2018		Fig. 70 Jeu créer avec Bitsy. [capture d'écran]
Fig. 11 <i>Game Moutier. Moon Does Stress.</i> 2016. [capture d'écran]	Fig. 30 <i>Gulizia, Ivan. Représentation schématique de la mécanique de tir.</i> 2018		Fig. 71 Jeu créer avec Bitsy. [capture d'écran]
Fig. 12 <i>Game Moutier. Moon Does Stress.</i> 2016. [capture d'écran]	Pages 24 - 25		Pages 50 - 51
Fig. 13 <i>Game Moutier. Moon Does Stress.</i> 2016. [capture d'écran]	Fig. 31 Lego. Mindstorm EV3. 1998. [capture d'écran]		Fig. 72 Sketch Nation, Inc. <i>Interface de Sketch Nation.</i> 2012. [capture d'écran]
Pages 18 - 19	Fig. 32 Labs Division of SNQ Digital. Proto.io. 2011. [capture d'écran]		Fig. 73 Sketch Nation, Inc. <i>Interface de Sketch Nation.</i> 2012. [capture d'écran]
Fig. 14 <i>Game Moutier. Fiche: Crédit de personnage humainoïde.</i> 2016	Fig. 33 Autodesk. Sketchbook Motion. 2015. [capture d'écran]		Pages 52 - 53
Fig. 15 <i>Medawar, Larissa. Photo de l'atelier "Game Moutier".</i> 2016	Pages 26 - 27		Fig. 74 Sketch Nation, Inc. <i>Interface de Sketch Nation.</i> 2012. [capture d'écran]
Fig. 16 <i>Medawar, Larissa. Photo de l'atelier "Game Moutier".</i> 2016	Fig. 34 <i>Gulizia, Ivan. Schéma illustrant le comportement d'un Creeper.</i> 2018		Fig. 75 Sketch Nation, Inc. <i>Interface de Sketch Nation.</i> 2012. [capture d'écran]
Fig. 17 <i>Medawar, Larissa. Photo de l'atelier "Game Moutier".</i> 2016	Fig. 35 Resnick, Mitchel. Scratch. MIT Media Lab. 2011		Fig. 76 Sketch Nation, Inc. <i>Interface de Sketch Nation.</i> 2012. [capture d'écran]
	Fig. 36 Hutong Games LLC. Playmaker. 2011		Fig. 77 Sketch Nation, Inc. <i>Interface de Sketch Nation.</i> 2012. [capture d'écran]
	Pages 28 - 29		Fig. 78 Sketch Nation, Inc. <i>Interface de Sketch Nation.</i> 2012. [capture d'écran]
	Fig. 37 Digital Pocket Limited Liability Company. Viscuit. 2015		Fig. 79 Sketch Nation, Inc. <i>Interface de Sketch Nation.</i> 2012. [capture d'écran]
			Fig. 80 Sketch Nation, Inc. <i>Interface de Sketch Nation.</i> 2012. [capture d'écran]
			Fig. 81 Sketch Nation, Inc. <i>Interface de Sketch Nation.</i> 2012. [capture d'écran]

Pages 54 - 55

- Fig. 82 Jeu créer avec Sketch Nation. 2018 [capture d'écran]
Fig. 83 Jeu créer avec Sketch Nation. 2018 [capture d'écran]
Fig. 84 Jeu créer avec Sketch Nation. 2018 [capture d'écran]
Fig. 85 Jeu créer avec Sketch Nation. 2018 [capture d'écran]
Fig. 86 Jeu créer avec Sketch Nation. 2018 [capture d'écran]
Fig. 87 Jeu créer avec Sketch Nation. 2018 [capture d'écran]

Pages 66 - 67

- Fig. 101 Media Molecule. Interface de Dreams. 2018
Fig. 102 Media Molecule. Capture d'écran d'un live stream. 2018. [capture d'écran]

Pages 68 - 69

- Fig. 103 Media Molecule. Interface de Dreams. 2018
Fig. 104 Media Molecule. Interface de Dreams. 2018
Fig. 105 Media Molecule. Interface de Dreams. 2018
Fig. 106 Media Molecule. Interface de Dreams. 2018

Pages 70 - 71

- Fig. 107 Media Molecule. Jeu créer avec Dreams. 2018
Fig. 108 Cézanne, Paul. Nature Morte. Vers 1890

Pages 72 - 73

- Fig. 109 Media Molecule. Jeu créer avec Dreams. 2018
Fig. 110 Media Molecule. Jeu créer avec Dreams. 2018
Fig. 111 Media Molecule. Jeu créer avec Dreams. 2018
Fig. 112 Media Molecule. Jeu créer avec Dreams. 2018

Pages 56 - 57

- Fig. 88 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012
Fig. 89 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012

Pages 58 - 59

- Fig. 90 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012
Fig. 91 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012 [capture d'écran]
Fig. 92 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012 [capture d'écran]
Fig. 93 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012 [capture d'écran]

Pages 60 - 61

- Fig. 94 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012 [capture d'écran]
Fig. 95 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012 [capture d'écran]

Pages 62 - 63

- Fig. 96 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012 [capture d'écran]
Fig. 97 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012
Fig. 98 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012 [capture d'écran]
Fig. 99 Slaczka, Jeremiah. Scribblenauts. 5th Cell. 2012 [capture d'écran]

Pages 64 - 65

- Fig. 100 Media Molecule. Jeu créer avec Dreams. 2018

Remerciements

Merci Nicolas Nova de m'avoir sereinement guidé tout au long de l'élaboration de ce mémoire.

Merci à l'équipe pédagogique de la HEAD pour votre suivi et votre compréhension.

Merci Larissa Medawar pour ton inestimable patience.

Merci Agustina Mancini et Mylène Silva pour votre aide précieuse.

Merci Emilie Parendeau pour ton soutien.

Merci maman pour tous les sacrifices que tu as faits pour moi.