**Level Design : Exercice**

• Choisissez et analysez 3 jeux existants de l'industrie qui selon vous ont su marier leurs 3C avec un level design qui en exploitait les possibilités de façon optimale : Rythme et combinatoire entre les capacités de l'avatar et les éléments de level design lié aux gameplay elements.  
  
• Établissez le concept document de votre projet, et répondez aux questions suivantes :  
- Quels éléments de votre réflexion vous ont poussé à choisir ce traitement spécifique ?  
- Expliquez votre traitement des trois notions-clés de progression (Learning, Difficulty, Rhythm) vis-à-vis de votre projet.  
- Expliquez selon vous les meilleures façons de relier les capacités de vos avatars à des mécaniques combinatoires liées aux gameplay elements.  
  
• Prendre au moins un exemple de gameplay element et établir les étapes du cycle d'apprentissage lié à ce dernier ainsi que les combinatoires exploitables liées à vos avatars.

**Sommaire :**

**[à faire]**

**Premier exemple : Mario Kart**

**Point de vue :**

En vue TPS, on contrôle un kart avec différents attributs, mais des contrôles similaires. Dans Mario Kart la vue à la troisième personne est parfaitement adaptée : elle permet à la fois de voir au loin mais aussi tout autour du Kart. Elle est positionnée de manière à offrir au joueur une marge de manoeuvre conséquente, et ne pas le surcharger d'informations inutiles : celles dont le joueur a besoin se trouvent dans l'environnement immédiatement proche et sont donc facilement lisibles.

**Gestion du rythme :**

Le rythme de la partie est fortement dépendant de la **topologie du circuit** : si le joueur négocie mal un virage, il peut n’apercevoir les pièges qu’au dernier moment, rater une opportunité d'obtenir un bonus, ou tomber dans le vide.

L'action se déroule à **haute vitesse**, étant donné qu'il s'agit d'un jeu de karting : de plus, toute interaction entre deux joueurs implique forcément que l'un **bouscule** l'autre et marque un frein au rythme de sa partie. Percuter consécutivement plusieurs joueurs avec un boost est donc un défouloir important et rend le **rythme d'autant plus effréné qu'il est précaire.**

**Capacités de l'avatar et environnement : combinatoire**

Le Level Design va influencer le contrôle avec des pièges sur la pistes, rendant les karts plus glissants par exemple. Ce risque est réduit par la simple compétence de saut : les joueurs ont effectivement toujours la possibilité d’esquiver ces pièges car ils ont la capacité de faire de petits sauts permettant de passer au dessus de ces éléments. Pour un bon joueur il sera donc tout à fait possible de passer tous les obstacles sans en toucher un seul s’il maîtrise correctement la conduite et le saut.

De plus, la **sensation de vitesse** est accentuée et optimisée par de nombreux éléments de l'environnement, notamment les boosters et les tremplins qui permettent non seulement d'augmenter grandement sa vitesse mais aussi de renouveler **vertigineusement** l'expérience dans les airs.



**Deuxième exemple : OGame**

**Point de vue :**

Interface entière. Il ne s'agit pas pour autant d'un jeu textuel, mais il fonctionne par menus, à tel point que l'on peut considérer la GUI comme terrain de jeu.

****

**Gestion du rythme :**

Le principe de ce jeu est que le fait que le challenge soit ou non soutenu dépend entièrement du reste de la **communauté**. En effet, OGame est un jeu construit autour des notions de **production, échange et vol de ressources**, et le joueur ne peut se sentir activement challengé que s'il y a danger, donc présence d'autrui, ou situation requérant la surveillance d'un grand nombre de paramètres jusqu'à la surcharge (multitasking).

**Capacités de l'avatar et environnement : combinatoire**

Tout d'abord, il est difficile de définir l'avatar dans un jeu tel que celui-ci. Le joueur est scénaristiquement placé dans la peau d'un empereur galactique mais cela n'influe en rien sur le gameplay. Le propre de ce personnage est de contrôler de multiples entités et gérer différents **flux de ressources** simultanément. Or, ces flux de ressources sont également **l'objet des affrontements entre joueurs** et ce sont eux qui modélisent un environnement de jeu : certains joueurs sont plus proches de "zones" à forte production de l'une ou l'autre des ressources, alors même qu'il n'y a aucun environnement spatialisé explorable.

**Troisième exemple : Portal**

**Point de vue** :

Caméra FPS, un personnage principal presque inconnu, Chell (que l'on voit juste une fois dans un miroir), et **une aptitude centrale** : pouvoir utiliser le Portal Gun pour créer des raccourcis spatiaux. **Les contrôles sont simples.**

**Gestion du rythme :**

Dans Portal, la vue FPS apporte une dimension assez originale puisqu’il s’agit avant tout d’un **platformer** : le challenge de visée est existant mais n'est pas au centre, et le choix d'une telle caméra pour des épreuves de platforming peut à première vue sembler aussi déroutant. Contrairement à beaucoup de jeu du genre platforming, on ne voit donc pas son personnage. Le rythme de jeu est donc relativement **calme** car le joueur ne peut prendre part à une expérience à haute vitesse dans cet état : cependant, cela se prête très bien à un **gameplay orienté réflexion**.

Les salles se présentant sous forme d’énigmes mettent toutes en scène le même challenge pour le joueur : aller d'un point A à un point B. Le Level Design est construit de manière à ce qu’il n’y ait qu’une seule solution possible aux énigmes, et celles-ci sont de plus en plus compliquées au fur et à mesure de l’aventure. L’apprentissage de toutes les mécaniques se fait donc au fil de l’avancée, nous donnant alors toutes les clés pour réussir à passer. Et souvent le LD est fait de telle manière que l’on voit exactement comment faire car nous l’avons appris auparavant mais le problème est **« comment le réalise-t-on ? »**



**Capacités de l'avatar et environnement : combinatoire**

Les contrôles du FPS sont restés dans leur forme classique, ce qui permet de les retrouver assez facilement. Il a donc fallu **adapter le level design** à tout cela.

Le Portal Gun permet de créer des portails, il est entièrement exploité et tout le LD tourne autour de son utilisation : il est même prévu que ce LD soit en fin de compte modelé par les joueurs qui, avec leur gun, peuvent **changer à volonté la circulatoire des niveaux**.

Le Level Design joue donc avec l’utilisation du Portal Gun et bien souvent avec la restriction de déplacement de l’avatar. Nombreux sont les cas où le joueur ne peut presque pas se déplacer car il est sur une plate-forme et il doit trouver comment passer l’obstacle avec la simple utilisation des portails. Cette mécanique revient à chaque pièce mais se décompose de plusieurs manières : utilisation de caisses, de peintures ou autre. **Le potentiel combinatoire des objets de LD et des aptitudes de Chell, bien qu'elles soient peu nombreuses, est énorme.** Ce jeu répond à la question suivante : comment activer le mécanisme à distance ?

**Concept document de notre projet :**

**Structure du MMO asynchrone :**

Le principe de ce jeu, **en ligne sur PC**, est de proposer à une large communauté de joueurs, ne se connaissant pas forcément (donc de type massivement multijoueur), de se **relayer** sur une même partie d'un jeu en format endless. Chacun se voit donc attribuer dix minutes pour jouer, avant de **léguer sa partie** **à un autre joueur et** **d'hériter d'une autre au hasard** : durant les dix minutes imparties, le but est d'avoir un **score** aussi haut que possible pour finir en haut des leaderboards.

**Type de gameplay associé à cette structure :**

Le jeu fonctionne sur une logique d'accumulation de **ressources**, que l'on assimilera à de l'énergie, que le joueur pourra ensuite dépenser pour passer d'un archétype "**gestion**" à une dynamique "**défouloir**" lui permettant d'acquérir du score.

Contrôlant un avatar en vue TPS dans un environnement 3D de très grande taille (map trop grande pour être totalement explorée en une seule partie), le joueur doit traverser des épreuves de **puzzle-platforming** pour atteindre et **gérer** des structures générant de l'énergie. Cette énergie accumulée peut ensuite être dépensée pour obtenir des boosts en mobilité (par exemple : vitesse ou saut) et acquérir la capacité de **dégrader et détruire l'environnement**, ce qui rapporte du score.  
 L'écosystème dégradé et/ou détruit se **régénère** lentement, et ce processus peut également être accéléré par l'action du joueur.

Le déroulement de ce gameplay peut donc se résumer en une boucle :

**Mécaniques de jeu principales :**

**1) S'amuser avec l'environnement :**  
• L'environnement est quasi-exclusivement constitué d'éléments de platforming simples : trous, marches, piliers...  
• Un type d'objet est doté d'un comportement particulier : le **bloc suiveur**. Ses dimensions sont variables, parfois le bloc est assez bas pour que l'avatar puisse **monter dessus**. À partir du moment où le joueur l'approche, le bloc suiveur se met à imiter tous les mouvements du joueur au sol.  
• Le joueur peut donc se servir de cet objet pour accéder à des zones de l'espace qui étaient jusque-là inaccessibles.

**2) Centrale énergétique et collectibles :**  
• Autre objet présent dans l'environnement : la **centrale énergétique**. Cet objet est de grande taille, bien visible, et ne peut pas bouger. Lorsqu'un bloc suiveur est **acheminé** près d'une centrale énergétique, celle-ci se met à **émettre** une certaine quantité de **collectibles** (énergie) à intervalle de temps fixe.  
• Les blocs suiveurs déposés près de la centrale énergétique arrêtent de nous suivre.  
• Plus une centrale a été approvisionnée en blocs suiveurs, plus les collectibles sont produits fréquemment.

**3) Gestion de l'énergie et boosts :**  
• Les collectibles émis par la centrale remplissent la **jauge d'énergie** du joueur.  
• Les collectibles **disparaissent** après un laps de temps donné. Il appartient donc au joueur de se trouver près de la centrale au bon moment.  
• À tout moment, s'il a assez d'énergie, le joueur peut en **dépenser** une partie pour s'octroyer un **boost de mobilité au choix** : un boost de vitesse, un double saut, un aimantage des collectibles... **pendant une durée limitée**.  
• Le joueur peut tout à fait dépenser plusieurs fois consécutivement de l'énergie pour **cumuler ou améliorer** ses boosts. Le cas échéant, cela lui coûtera encore plus cher.  
• La jauge d'énergie, même au maximum, ne permet pas de cumuler tous les boosts à pleine puissance simultanément : le joueur devra faire des **choix**.

**4) Dégradation de l'environnement et score :**  
• Lorsqu'il est boosté, le joueur obtient la possibilité de **frapper** les objets alentour tels que les blocs suiveurs et les centrales. Ils émettent alors des collectibles différents, donnant cette fois-ci non pas de l'énergie mais du **score**.  
• Les objets frappés à répétition peuvent être **détruits**. Plus ils sont détruits vite et plus ils donnent un bonus de score conséquent (abondance de collectibles).  
• Une fois détruits, les objets de l'environnement se **restaurent** peu à peu. C'est un processus naturel qui prend plusieurs minutes.  
• Si tous les blocs suiveurs et toutes les centrales de la map sont **simultanément détruits**, tout l'écosystème **se régénère entièrement, instantanément**. Le joueur a donc intérêt à partir à la recherche des derniers objets présents pour les casser au plus vite et gagner du temps.

**Expérience associée et comportements de joueur :**

Ce système de jeu a pour but premier de mettre en place, de façon paradoxale, une **interdépendance** forte entre les joueurs d'une même communauté mais qui ne joueront jamais "ensemble" et n'interagiront pas directement. Ainsi, il est prévu que deux grands archétypes de joueurs se dégagent :  
• Les ***seeders***, plus attirés par l'aspect gestion et puzzle, qui chercheront à optimiser la production d'énergie et en laisseront donc beaucoup aux joueurs suivants.  
• Les ***leechers***, plus attirés par les possibilités de défouloir qu'offrent les boosts, et qui vont avoir tendance à dépenser toutes les réserves d'énergies pour optimiser leur score.

Une tension va s'installer entre ces joueurs : y aura-t-il **rivalité ou entraide** ? Pour permettre une communication minimum, débouchant aussi sur certaines formes d'émergence et de créativité, un **journal de la partie** sera disponible et consultable à tout moment par le joueur. À la fin de sa partie, avant de la léguer à un autre joueur (inconnu) au hasard, il pourra y inscrire **jusqu'à 100 caractères** de son choix. Remarques, tentatives de mise en place d'une stratégie long-terme, roleplay improvisé, critique à l'égard du joueur précédent... Toute la communauté pourra lire ce journal, à défaut d'être certain de pouvoir jouer cette partie un jour.

**Apprentissage :**

Etant donné qu'il s'agit, même indirectement, d'un mode multijoueur, on attend des joueurs qu'ils connaissent déjà les règles et mécaniques du jeu. C'est pourquoi, pour être certain que chacun maîtrise ce système, **un mode offline (solo) est disponible à titre de tutoriel**, pour permettre l'apprentissage des fondamentaux du jeu.

Dans ce mode solo, le joueur pourra, au cours d'une **partie plus courte** que la normale, sur une **map plus petite**, suivre le déroulement d'une **boucle de jeu** complète, depuis la découverte de l'environnement et des premiers blocs suiveurs jusqu'à la destruction de l'environnement et à la récolte des collectibles de score.

Le mode solo est considéré comme terminé, et les connaissances acquises, lorsque le joueur aura terminé une partie en ayant atteint un certain score et/ou détruit une certaine proportion de l'écosystème de jeu : s'il peut marquer des points et détruire des objets, c'est qu'il aura assimilé les mécaniques de jeu indispensables au jeu en multijoueur.

**Traitement du thème "Avatar & Environnement" :**

Ce jeu a été conçu autour d'un mot-clé imposé correspondant à une idée particulière : "**Avatars**". Plus précisément, après des recherches sur le thème, l'équipe a choisi d'approfondir la **notion de trace et d'héritage**, capitale lorsque l'on parle d'avatars : il s'agit d'interroger les modalités selon lesquelles dont un avatar perdure dans le temps, sous quelle forme et de quelle manière la présence passée d'un avatar impacte un environnement présent.

C'est ici qu'intervient la mécanique d'héritage d'une partie de jeu : le **journal** de la partie, permettant de visualiser la liste ordonnée des joueurs qui y ont pris part et de leurs commentaires, est d'une importance cruciale ici puisqu'il représente le parcours passé de la partie et les traces du passage de chacun sur le jeu.

De plus, le jeu propose, de par sa structure, deux configurations intéressantes pour les questions de l'avatar et de l'identité :  
• Les joueurs **changent** perpétuellement d'avatar, à chaque fois qu'ils héritent d'une partie différente, c'est à dire toutes les dix minutes.  
• Les avatars sont **partagés** par les joueurs à travers le temps. Il est difficile de pouvoir estimer que l'on en "possède" un sur le long terme, excepté peut-être si le hasard fait qu'un même joueur retombe plusieurs fois sur la même partie.

Le gameplay à proprement parler est donc construit en fonction de cette structure macro : la problématique centrale est "comment rendre les joueurs fortement interdépendants alors qu'ils n'interagiront pas directement ?". D'où trois notions-clés de ce gameplay :  
• Un **système de ressources** matérialisant l'héritage transmis de l'un à l'autre, et permettant l'émergence de comportements différents, éventuellement opposés et/ou complémentaires.  
• Un **environnement à explorer** (sur le modèle du platformer), sous la forme d'une map trop grande pour être entièrement parcourue en une seule partie. De la sorte, les joueurs vont devoir se relayer pour la traverser en entier, et un même joueur devra passer plusieurs parties avant de connaître son environnement et développer une bonne ***map-awareness***. C'est pour cette raison qu'il n'est pas question d'un environnement procédural, généré aléatoirement.  
• Ainsi, les joueurs sont réunis autour d'une même situation de jeu bien claire, qui est celle de la **gestion d'un écosystème complet**. Et, de la même manière que dans la vraie vie, ils peuvent choisir d'oeuvrer pour la sauvegarde de cet écosystème ou de le dégrader pour leur propre intérêt.

Le thème "avatar" nourrit également le gameplay micro, très orienté environnement : ce système de jeu pose la question "de quelle structure ai-je réellement le **contrôle** ?". S'agit-il uniquement du personnage principal évoluant dans l'environnement, ou aussi des blocs suiveurs autour de lui ? De plus, selon cette considération, ces blocs seraient, eux aussi, des **avatars éphémères** : on n'en a le contrôle que jusqu'à ce qu'ils soient amenés vers une centrale, ou détruits.

**Progression du joueur : Learning, Difficulty, Rhythm**

La **difficulté** est un des éléments les plus cruciaux en level design. Dans notre projet, ce sont les phases de puzzle-platformer qui viendront challenger directement le joueur.

Les puzzles seront donc relativement simples au début : Ici le but est que le joueur comprenne les mécaniques de base du "parenting" ( = fait de guider et acheminer des blocs par nos propres mouvements). Et à chaque introduction de nouvelles mécaniques, une boucle d'apprentissage sera mise en place afin de **valider le processus d'apprentissage**.

Ce processus d'apprentissage correspond au mode solo, tout comme dans *StarCraft* : c'est un **préparatif au mode multi** que les joueurs considèreront comme "le vrai jeu" ou "la cour des grands". Au cours de cette session d'apprentissage, le joueur vivra une itération complète de la métaboucle du jeu, dont chaque étape est **validée par un reward** (des collectibles) pour faire comprendre au joueur qu'il agit correctement, et l'y encourager, selon le modèle de la Skinner Box :  
• Lorsqu'il **dépose** un bloc près d'une centrale, il reçoit de l'**énergie**.  
• Lorsqu'il **revient** vers une centrale, il reçoit encore de l'**énergie**.  
• Lorsqu'il **consomme** son énergie et frappe des objets, il reçoit du **score**.  
• Lorsqu'il **détruit** l'environnement, il reçoit encore du **score**.

Puis la difficulté augmentera, mais proposera des **pics de challenge** à un instant T, et bien défini, afin de varier le rythme.

**L'environnement** constitue aussi un challenge. La méthode Super Meat Boy est appliquée. Pas d'ennemi, ou d'IA, simplement des **obstacles létaux** : pics, gouffres, etc.

Concernant le **rythme**, les puzzle-platformers ont tendance à posséder un tempo lent qui requiert une réflexion sérieuse et de la contemplation, le joueur doit avoir le temps de regarder autour de soi, et de déterminer sa propre voie. Pour augmenter le **pacing**, il est pensé de **mouvoir les obstacles**, ou encore de rajouter une **composante temps** dans les énigmes, de façon directe ou indirecte.

La phase **open-world** quant à elle est une phase plus détendue : ici le joueur **contemple** l'environnement, et prend le temps de **chercher** les centrales dans les différents endroits de la map. Le rythme pourra être également contrôlé par l'ajout de mécaniques futures, tout comme dans le mode solo le joueur apprend sans cesse de nouvelles choses en avançant dans la game loop : cela permet au joueur d'être **constamment engagé dans le jeu**. (ceci pourra être créer via une matrice de variétés)

Lorsque le joueur a son énergie au maximum, il peut alors **tout détruire** très efficacement. Ici, le rythme va considérablement augmenter. Il y aura donc un **effet cathartique**, où il cherchera parcourir la map et la nettoyer autant que possible, en peu de temps. Ici, on alterne donc les **phases de** **repos** et de détente, avec les **phases plus nerveuses** en détruisant l'environnement. Ce qui est intéressant dans ce cas, c'est que c'est le joueur qui construit sa propre expérience. Le level design sert à donner une **sensation de pouvoir** au joueur. Ici, on rejoint la notion *d'empowerment* évoquée par Dan Taylor dans son article *Ten Principles of Good Level Design*.

**Combinatoire : synergie entre capacités du joueur et propriétés de l'environnement**

Les contrôles de base sont très **simples** : le joueur peut uniquement se déplacer, sauter et (parfois) attaquer, ce qui constitue deux inputs en plus du contrôle du mouvement.

Mais les capacités du joueur sont censées **évoluer dynamiquement** avec la situation de jeu, grâce au système de boosts : ainsi, le joueur pourra acquérir temporairement, lorsqu'il le peut et en a besoin :  
• Un double-saut, voire triple-saut  
• La capacité d'attaquer  
• La capacité de détruire qui, à haut niveau d'énergie, ne nécessite pas forcément de lancer une attaque mais juste de foncer dans un objet

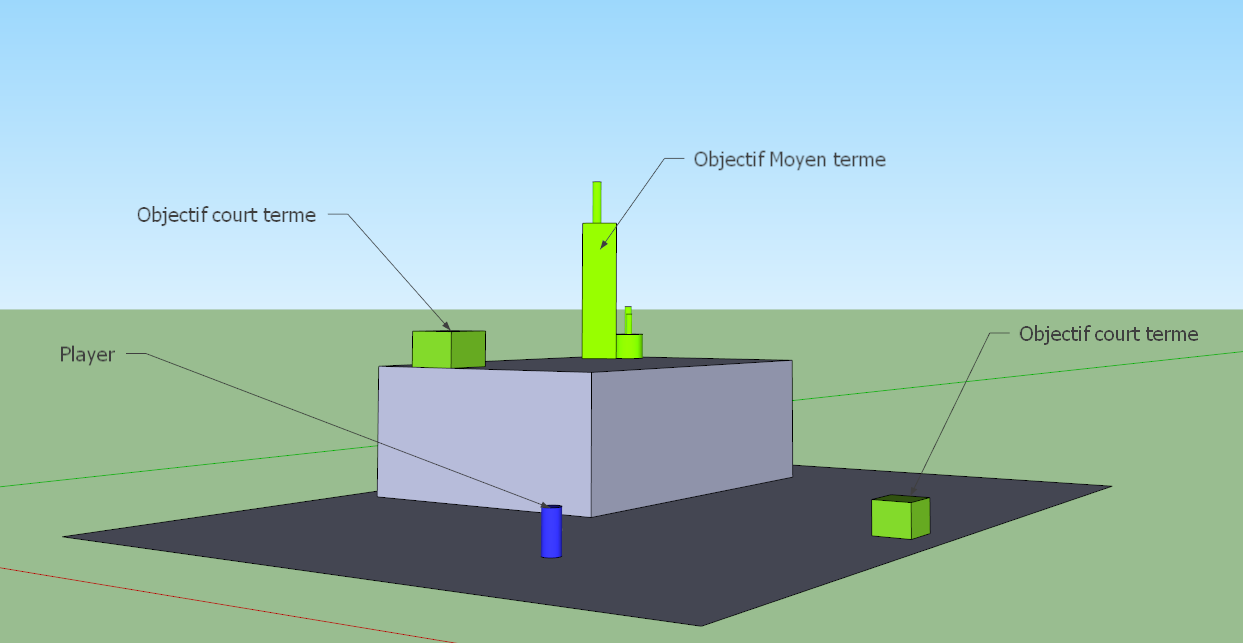
**L'accélération** étant également due à une consommation d'énergie, on peut la considérer comme une capacité du joueur.

L'environnement s'adapte donc à ces changements dynamiques de capacités du joueur : l'idée est de **verrouiller indirectement** certaines régions et certains puzzles tant que le joueur n'a pas les capacités pour. Par exemple, certaines plateformes et certains blocs suiveurs seront situés en hauteur et le joueur aura besoin d'un double saut pour y monter. Ou bien, il devra amener un bloc en bas pour monter dessus comme une marche.

**Exemple concret : utilisations multiples des objets liés au gameplay**

La configuration de l'environnement peut **suggérer** au joueur quelles **utilisations détournées** du bloc suiveur il peut faire, guidé par la centrale. Si le joueur a appris leur fonctionnement de base en mode solo, le mode multi peut être l'occasion de découvrir par lui-même d'autres moyens indirects de s'en servir.

**Situation n°1 : utiliser un bloc suiveur comme marchepied.**

****

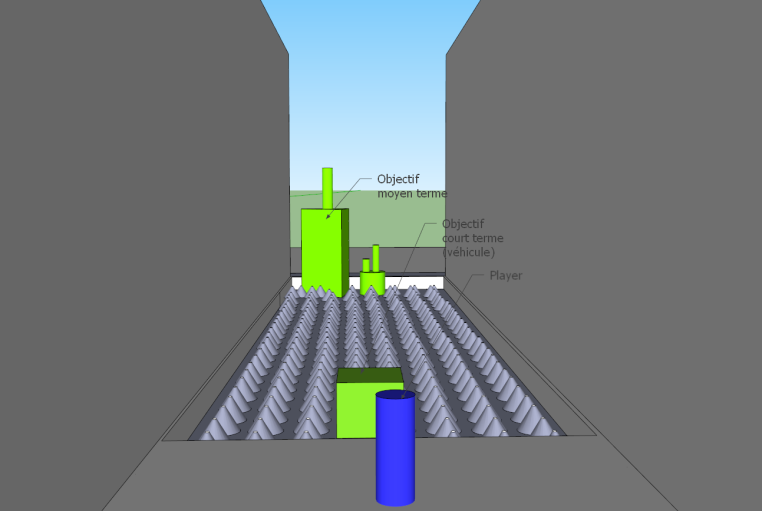
• **Teasing** : en voyant une portion de l'environnement qui est trop haute pour être atteinte en un seul saut, mais qui contient un objectif (une centrale et un bloc), le joueur comprend qu'il peut y aller. Il a alors envie de l'atteindre.

• **Learning** : la présence d'un bloc suiveur au sol suggère fortement que le joueur peut s'en servir pour faire la courte-échelle, comme il connaît le comportement de cet objet.

• **Practicing** : le joueur rencontrera cette même situation plusieurs fois sur la map.

• **Mastering** : après un certain temps, le joueur rencontrera une telle situation mais à difficulté supérieure et avec un enjeu lié à sa vie : par exemple, il sera nécessaire d'utiliser plusieurs blocs suiveurs, et la zone en hauteur pourra être entourée de vide ou de pics.

**Situation n°2 : utiliser des blocs suiveurs comme "véhicule" pour marcher sur des pics.**



• **Teasing :** le joueur voit un gouffre contenant des pics. Mais au bout de ce gouffre se trouve une centrale. Les centrales étant immobiles, il comprend qu'il existe un moyen de l'alimenter en blocs. Il va chercher à l'alimenter, car il sait que c'est une source de reward potentielle.

• **Learning / Practicing** : la première fois qu'il va rencontrer cette pattern (donc en mode solo), des blocs seront déjà disposés sur les pics. De cette manière, le joueur apprendra à monter sur des blocs pour les faire bouger en même temps que lui, mais sans danger car la zone de pics sera recouverte. Au fur et à mesure qu'ils rencontre de telles patterns, il sera de plus en plus difficile de trouver un bloc et la superficie sur laquelle son avatar pourra se tenir se réduira.

• **Mastering :** en mode multi, le joueur peut être forcé d'exploiter cette centrale car il n'aura pas le temps d'aller en chercher une autre. Il devra donc constamment aller chercher des blocs suiveurs pour les acheminer vers elle en montant dessus. Les dimensions des blocs suiveurs n'étant pas toujours les mêmes, trouver l'ordre idéal dans lequel les poser pour ne pas tomber dans les pics et se blesser peut constituer un challenge.