**Thématique : l’opposition symbolique entre les seeders et les leechers.**

**Support :**

• PC, un seul joueur, mais il devra aller sur le site du jeu avant et après sa partie.  
• Chaque partie ne peut durer qu’un temps limité. (10mn ?)  
• Ensuite, la save est updatée et uploadée (.xml) sur une base commune où un autre joueur la continuera. C’est donc un mode endless où tout le monde se relaye.

**Principe :**

• On contrôle un véhicule/mecha/whatever, qui se déplace sur un terrain contenant pas mal d’objets et de ressources.  
• Le but final : parcourir le plus de distance possible et détruire le plus d’objets possibles sur ce terrain (mode gros défouloir) pendant sa partie, de façon à scorer et terminer sur la leaderboard.  
• **MAIS**, accélérer et détruire sont deux actions (jouissives pour le joueur) qui lui coûtent des ressources. D’autant que les objets à détruire seront aussi ceux qui donnent des ressources. Et ces ressources, il les accumulera calmement sur le terrain, grâce aux items dessus, qui proposeront d’autres activités (réflexion/gestion/etc). => C’est donc un pan de gameplay totalement différent.

**Comportement de joueur en conséquence :**

Le joueur n’aura pas le temps matériel de faire les deux vu le temps limité de sa partie.  
**Donc, un dilemme s’offre à lui :**  
• Soit il voit qu’il peut accumuler des ressources et faire un bon score. (LEECH)  
• Soit il va se montrer sympa envers le prochain qui aura sa partie, et lui ménagera un truc cool avec pas mal de ressources. (SEED)

Le seeder pourra même être crédité sur la leaderboard si le joueur suivant gère bien sa partie.

Pour faire simple : c’est une décomposition en deux temps de la tension gameplay « j’accumule des ressources pour mon but final qui est de détruire leur source ».

Et vu la durée de la partie, quelle que soit son action, le joueur n’aura pas le temps de s’ennuyer. Ça donne un jeu au gameplay variable (voire aléatoire, selon sur quelle partie on tombe), et avec une rejouabilité très importante.

**Extras :**

• Plusieurs maps possibles avec un LD variable ?  
• Journal de la partie : consultable à tout moment par le joueur, à la fin de sa partie celui-ci a le droit d’y écrire jusqu’à 100 caractères.  
• Visionner l’historique du véhicule/mecha/whatever/truc contrôlé par les joueurs : nombre de kilomètres parcourus depuis le début, nombre d’objets détruits, date de création, etc.  
• Customisation de l’apparence de l’avatar ? (ça formerait même un troisième pan de gameplay, accessoire, pour ceux qui ne sont pas intéressés par le score)  
• Un mode multi local en coop ?  
• Un mode multi local où on peut se tirer dessus et réduire le terrain plein de ressources en un PUTAIN DE CHAMP DE RUINES (auquel cas, ça tournerait presque au RTS, et ce serait trop cool)  
• Sur le journal de la partie, le joueur n’aura plus qu’à mettre : « j’ai invité un pote à moi, on s’est battus, on a tout cassé, désolé ! »

**Inspirations :**

• Github  
• µTorrent  
• Chatroulette  
• Jeter une bouteille à la mer  
• Farmville  
• Mario Kart  
• Katamari Damacy

**Ce que j’aurai à coder sur Unity :**

• Une bonne caméra TPS  
• Amélioration du characontroller 3D d’Unity (pour accélérer, dasher/charger avec l’avatar, et déclencher les anims)  
• Destructibilité/régénération des objets ( = deux animations)  
• Le gameplay gestion des ressources et la GUI qui va avec  
• Export de l’état des ressources du terrain et de l’avatar dans un fichier XML read-only comme on a fait en spé l’an dernier

=> Jusque-là, ce sont des trucs très simples. Ensuite il y a tout ce qui est web-related :  
• S’assurer qu’un même joueur ne joue pas plusieurs fois d’affilée sur la même partie  
• Polish : compte, stats d’un joueur, etc.

**Un jour plus tard : résultats de la jam de l’autre moitié du groupe**

• Le concept cité ci-dessus est peut-être très cool mais son gameplay micro est insuffisant.

• À l’inverse, l’autre groupe présente une mécanique qui soulève plein de points intéressants du côté micro.

• Leur base étant plutôt orientée puzzle-game, le merging direct n’est pas possible, mais il est possible de prendre une large partie de l’expérience et des contrôles proposés :

**1 -** Le contrôle multiple « soude » l’avatar et les objets de LD. On s’y sent lié.  
**2 -** Si cette synchronisation est forcée, le joueur ne va pas se sentir libre de ses mouvements.  
**3 -** Et ça n’est pas une mauvaise chose, parce que cela va le forcer à réfléchir ses moves.  
(exemple : utiliser les murs pour se désynchroniser des autres trucs qui bougent comme nous) => On couple une dimension gestion à une dimension puzzle, et toujours en temps réel.  
**4 -** Ainsi, lorsque le joueur n’est plus linké à ces autres objets, il se sent **libéré**. Et le côté défouloir devient réellement libérateur/cathartique et fait encore mieux son travail. Le joueur va aller courir, il va pouvoir frapper les objets auxquels il était contraint.

=> Les objets détruits par le joueur lâchent des collectibles en grande quantité, alors qu’une utilisation de ces objets dans un cadre puzzle/gestion rapportait des quantités plus petites de collectibles.

=> Le système de scoring, central, provient de ces collectibles.

=> Les joueurs sont mis en **coopétition** à cause de ce mode multijoueur indirect : leur objectif est clairement d’atteindre le début de la leaderboard, mais ils doivent s’entraider pour que l’un d’entre eux atteigne des scores intéressants.

**La mécanique au centre de la gestion : accumulation de ressources et burst**

• Le LD doit permettre au joueur plusieurs types de moves possibles avec son environnement, de sorte qu’il puisse produire deux types d’outputs : (on va mettre un code couleur histoire que ça puisse être compréhensible sans schéma)  
- Collectibles jaunes => Score  
- Collectibles bleus => Energie (disons, une jauge)

• Le joueur peut ensuite consommer de l’énergie. Il peut choisir le type d’aide que cela va lui apporter : possibilité de détruire son environnement, de se déplacer plus vite, de sauter, etc.  
=> Il gagne en **puissance**, au sens premier du terme.

• Plus le joueur consomme d’énergie en même temps, plus cette montée est importante. (non-linéaire, par exemple exponentielle, pour lui donner intérêt à patienter)

**=> N’oublions pas que la durée de la partie est limitée. En fait, optimiser sa partie est donc une question de choix stratégique** (quand et comment claquer ses boosts ?), **tactique** (comment optimiser la production ?) **et aussi une question de skill physique pur** (dextérité éventuellement imposée par le LD. Imaginons, le joueur doit faire tomber des objets dans des trous, mais risque d’y tomber lui aussi…)

• Attendre jusqu’à stacker tous les boosts amène donc, en théorie, à un état d’omnipotence, jouissif pour le joueur et très rentable.  
=> Dilemmes intéressants : si cet état ne peut jamais être atteint et/ou fait perdre en maniabilité, le joueur aura toujours besoin de skill et pourra rendre sa performance difficile à égaler.

• Problématique : quelles peuvent être les motivations, objectifs et rewards d’un joueur qui va seeder la partie suivante ?  
=> Être crédité dans la leaderboard  
=> Pour qu’il ne s’ennuie pas, il doit scorer un minimum. On peut par exemple ne valider sa partie que s’il atteint un certain seuil de score.  
=> Le score peut aussi découler de l’énergie accumulée, de sorte que certains joueurs préfèrent accumuler que dépenser.

**Le dernier point, très important, la dimension sociale :** les joueurs vont seeder par conscience que s’ils ne le font pas, et si tout le monde pense comme ça, ils n’hériteront jamais d’une partie où assez d’énergie a été laissée pour atteindre un high score. Ensuite, le **journal de la partie** peut être utilisé de multiples façons : du role-play ? La tentative de monter une stratégie long-terme entre inconnus ? L’anonymat, l’incertitude de la réception et la limite de caractères peut pousser à plein de comportements très diversifiés et bénéfiques pour le gameplay.