

Projet de jeu

Eric Debra

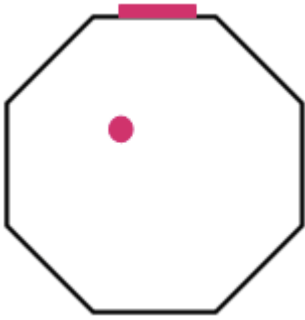
Jeu arcade où le but du mode principal est de survivre le plus longtemps possible en rattrapant une balle via le contrôle d'une raquette pour qu'elle ne sorte pas de la zone de jeu, un octogone.

Sommaire

I - Fonctionnement du jeu	3
1 - Bases générales	3
1.1 - Bases du mode « survie »	3
1.2 - Bases du mode « versus »	4
1.3 Multijoueur	4
2 - Niveaux de difficulté	5
3 - Contrôles	6
4 - Power-ups	6
5 - Scoring et classements	7
6 - Progression	7
7 - Système de sauvegarde	8
8 - Niveaux	8
9 - Obstacles temporaires dans les niveaux	8
10 - Feedbacks	8
II - Inspirations, choix artistiques, techniques, cibles et coûts	9
1 - Choix du moteur	9
2 - Développement par itération	9
3 - Choix artistiques et inspirations	9
4 - Coûts	9
5 - Cibles (joueurs et plates-formes)	9
6 - Prix de vente	10

I - Fonctionnement du jeu

1 - Bases générales



Le jeu se déroule dans un octogone. Le joueur contrôle une ou plusieurs raquettes d'une taille prédéfinie (cf. figure de gauche, rectangle en magenta). Une ou plusieurs balles seront dans la zone de jeu (cercle magenta ci-contre) avec comme but de ne jamais la ou les faire sortir de l'octogone. Le cas échéant, cela entraînerait la perte d'une vie, la fin de la partie (« Game Over »), ou simplement un point pour l'opposant (qui dans le cas d'un match à plus de 2 joueurs sera celui ayant envoyé le coup fatal).

La caméra de la zone de jeu fera des rotations sur elle-même dans les 2 sens tout au long d'une partie.

La balle aura une vitesse de rotation sur elle-même basée sur le mouvement de la raquette frappant la balle lors d'un rebond, affectant sa trajectoire.

Plusieurs niveaux apportant des modifications à la zone de jeu seront disponibles pour tous les modes de jeu. Par exemple, la présence de murs sur 2 des côtés parallèles de l'octogone.

Plusieurs niveaux de difficulté seront disponibles affectant différents paramètres tels que la rapidité à laquelle la vitesse de la balle augmente, et son hard cap (vitesse maximale). Une difficulté custom sera disponible pour affiner les paramètres manuellement.

En solo, le joueur pourra contrôler une raquette (mode normal), 2 raquettes (mode double), ou jouer contre l'IA (une raquette, mode versus IA). Contre l'IA seul le mode match sera disponible et non survie.

En multijoueur, chaque joueur contrôlera une raquette au maximum. Il y aura le mode coop et le mode versus.

1.1 - Bases du mode « survie »

Ce mode sera la base du jeu. Seuls le mode solo versus IA ou le multijoueur versus ne fonctionneront pas sous ce mode.

Le but du jeu sera de survivre le plus longtemps possible, et au minimum 100 secondes pour « valider le niveau ». Survivre signifie comme expliqué ci-dessus de ne pas faire sortir la balle de la zone de jeu, l'octogone, en se servant de sa ou ses raquettes pour cela.

La raquette pourra être bougée tout le long de la zone de jeu. En cas d'obstacles le long de celle-ci, le joueur aura sa raquette au-dessus ou en-dessous selon la position de la souris vs le centre de l'obstacle.

Chaque fois que la balle touchera la raquette / un obstacle, la vitesse de celle-ci augmentera. Selon des paliers à définir, la vitesse maximale que la balle peut atteindre sera augmentée. Par exemple, si la vitesse de la balle commence à 10, augmente de 0.1 par touche et est capée à 20 pour les 100 premières secondes de survie, cette vitesse capée sera à augmenter pour les 100 secondes suivantes par exemple afin de rendre la survie toujours plus compliquée. Un choix sera à faire si un « true hard cap » existera, ou si la vitesse pourra excéder ce qui est humainement possible de faire afin de provoquer une fin de partie à tous les coups.

Un mode « chaos » ou « enrage » pourrait être provoqué dans certaines conditions comme le fait que le joueur ferait rebondir la balle sur un axe identique à +/- X% près. Un avertissement sera affiché en jeu si un tel mode serait proche d'être déclenché. Son existence dépendra certainement de la présence ou non

d'aléatoire sur les rebonds en premier lieu. Ses effets pourraient être variés et dépendre du mode de difficulté actuel. Il pourrait impacter la déviation (ajout d'aléatoire) de la balle à chaque rebond, la rapidité à laquelle la vitesse de la balle augmente à chaque rebond, la vitesse maximum de la balle, la taille de la raquette, taille de la balle, etc.

Ce mode pourra se jouer en « classique » ou « powered-up ». En classique, il n'y aura pas de power-up, contrairement à l'autre.

Par défaut, le mode principal de jeu sera « solo classique survie normal ». « Normal » puisque le joueur contrôlera une seule raquette cf. I-1.

Liste de tous les modes « survie » :

- Solo classique survie : pas de power-up
 - Normal : 1 raquette
 - Double : 2 raquettes
- Solo powered-up survie : avec power-ups
 - Normal : 1 raquette
 - Double : 2 raquettes
- Multijoueur classique (survie) coop : pas de power-up
- Multijoueur powered-up (survie) coop : avec power-ups

1.2 - Bases du mode « versus »

Mode plus traditionnel où le but sera de faire sortir la balle de la zone de jeu dans le camp adverse. Chacun pour soi ou en team.

La victoire sera atteinte après avoir marqué un certain nombre de points, définis par le joueur.

Contrairement au mode survie où les joueurs seront (généralement ?) libres de se déplacer tout le long de la zone de jeu, ici le mouvement sera contraint à sa propre zone.

De la même façon que pour le mode survie, la vitesse de la balle augmentera via les rebonds.

Liste de tous les modes « versus » :

- Solo classique versus IA : pas de power-up
- Solo powered-up versus IA : avec power-ups
- Multijoueur classique versus : pas de power-up
- Multijoueur powered-up versus : avec power-ups

1.3 Multijoueur

Le jeu sera jouable en multijoueur de 2 à 8 joueurs. Dans le mode multijoueur de base, chaque joueur contrôlera une raquette. Cela fonctionnera de façon optimale de 2 à 4 joueurs, mais pourrait être disponible aussi à plus de joueurs (jusqu'à 1 joueur / côté de l'octogone).

Le choix sera donné pour les joueurs de se partager une raquette au lieu d'en contrôler une nouvelle. Un cumul des forces se fera dans ce cas. Par exemple si un joueur est tout à gauche de la zone qu'ils doivent protéger et l'autre tout à droite, la raquette sera au centre. S'ils sont plus ou moins à la même position, la raquette sera à cette position.

Chaque joueur choisira s'il veut sa propre raquette ou se joindre à un autre joueur, la couleur de celle-ci, sa team s'il y a lieu.

Selon le nombre de joueurs, et principalement de raquettes, la taille de chaque raquette devra être réduite en accord avec la zone à protéger.

Différents paramètres de jeux seront disponibles pour configurer une partie :

- En versus : nombre de points par manche à atteindre et nombre de manches.
- En versus : Team ou chacun pour soi. En team, chaque joueur pourra créer ou joindre une team.
- En versus : Mode dernier survivant (le round continue jusqu'à ce qu'il ne reste qu'un joueur) ou premier éliminé (le round s'arrête au premier joueur éliminé pour toujours avoir tous les joueurs dans la zone de jeu). En mode dernier survivant, les zones de jeu des joueurs éliminés deviennent des murs. Ces murs augmenteront peut-être la vitesse de la balle plus rapidement, et en dehors des limites habituelles afin de ne pas faire attendre de trop les joueurs éliminés.
- En coop : Mouvements libres (chaque joueur peut se déplacer sur la zone de jeu entière) ou restreints (chaque joueur a sa propre zone à protéger et ne peut se déplacer ailleurs).
- En coop : Mode dernier survivant (le round continue jusqu'à ce qu'il ne reste qu'un joueur) ou première défense échouée (le round s'arrête au premier joueur ratant une défense pour toujours avoir tous les joueurs dans la zone de jeu).
 - Si dernier survivant et mouvements restreints : Avec ou sans extension de la zone de jeu. Sans extension, comme en versus des murs dans les zones des joueurs ayant raté leur défense prendront place. Avec, les joueurs restants verront leur zone à défendre agrandie en se partageant la zone défendue par le joueur « éliminé ». Les raquettes et la balle seront immobilisées 2/3 secondes pour que l'extension soit bien visible et que les joueurs puissent se préparer.
- En coop : Avec ou sans collisions. Avec, les joueurs se bloqueront mutuellement rendant la survie plus difficile. Sans (par défaut), les joueurs pourront être au même endroit sans gêne.
- Versus & Coop : Niveau de difficulté.

2 - Niveaux de difficulté

Chaque niveau de difficulté pourra affecter des valeurs telles que :

- le nombre de vies
- la vitesse de base de la balle
- la vitesse maximale de la balle (à un palier donné, mais aussi un « true hard cap » éventuel)
- la rapidité à laquelle la vitesse de la balle augmente par rebond
- la chance d'obtenir un power-up (ne concerne que le mode « powered-up »)
- la pénalité de la mort sur le score (-X% du score actuel / mort, seulement dans les parties avec plus d'une vie)
- la vitesse de rotation de la caméra
- le pourcentage de déviation de la balle à chaque rebond par rapport à l'angle qu'aurait eu le rebond initial
- la chance de spawn d'obstacles supplémentaires temporaires dans la zone de jeu
- ...

Ces niveaux de difficulté seront à définir et à affiner par itération. On pourrait partir sur 5 niveaux, ainsi qu'un éventuel niveau bonus « custom » où le joueur pourrait à peu près tout paramétrer et sauvegarder ces paramètres sous forme de « preset ».

Les 5 niveaux seraient :

- Zen : Nombre de vies important, fait pour jouer en profitant simplement du jeu avec des vitesses relativement basses.
- Facile : Quelques vies (3 à 5), fait pour jouer presque normalement mais avoir un répit de par le nombre de vies.

- Normal : Une seule vie, vitesses de jeu standards. Difficulté par défaut.
- Difficile : Une seule vie, vitesses augmentées, etc.
- Expert

3 - Contrôles

Soumis à changement selon les itérations du jeu, comme tout. Il est possible que le contrôle total de la position de la raquette selon la position du curseur de la souris (sorte de téléportation de la raquette donc) soit modifié en faveur d'un système à vitesse de déplacement gauche/droite linéaire (sensibilité définie par le joueur ?). Cela nécessitera une décision rapide tôt dans le développement du jeu de par l'impact de ce choix.

Par défaut, les contrôles se feront à la souris afin d'avoir un contrôle total de la position de la raquette. Le stick d'une manette fonctionne également en indiquant précisément la direction souhaitée.

Le positionnement de la raquette se fera en fonction de la position de la souris par rapport au centre de la zone de jeu. Deux autres systèmes de contrôles pourront être testés afin de choisir le meilleur :

- Déplacer la souris dans un sens ou dans l'autre, le déplacement s'appliquera à la raquette (exemple : souris déplacée horizontalement vers la gauche, la raquette va se déplacer vers la gauche d'autant sinon plus selon la sensibilité définie par le joueur)
- Laisser seulement la possibilité gauche / droite (clic gauche / droit, touche directionnelle gauche / droite, etc.). La différence avec la solution ci-dessus étant qu'ici la vitesse de base sera non plus définie par le joueur qui déplacera plus ou moins vite sa souris ou son stick. Elle sera définie par le jeu lui-même. A voir si on permet avec ce choix de régler la sensibilité du déplacement. Selon les vitesses de jeu ce choix peut être difficilement jouable à moins d'adapter la vitesse de déplacement à celle du jeu, ce qui pourrait être complexe et problématique à jouer à cause d'une constante variation de la vitesse du déplacement.

Une « zone morte » pourrait être configurable pour empêcher les téléportations non souhaitées de la raquette si le curseur de la souris est trop au centre de la zone de jeu. Il s'agirait d'une zone partant du centre dans laquelle la raquette ne serait pas affectée par la position du curseur de la souris.

Selon l'endroit où la balle rebondira sur la raquette, l'angle du rebond en sera affecté.

La vitesse de déplacement de la raquette au moment du rebond (avec une fenêtre relativement généreuse, selon la difficulté peut-être aussi) définira également la vitesse de rotation sur elle-même de la balle, donnant un effet de courbure à sa trajectoire. Principalement utile en versus, puisqu'en survie, à moins de vouloir se donner de la difficulté en plus, ça ne sera pas utile. Sauf peut-être si un impact sur le score est établi avec la rotation de la balle.

4 - Power-ups

Dans le mode « powered-up », des power-ups feront leur apparition. Chaque seconde, un power-up aura une chance d'apparaître, chance définie par la difficulté actuellement jouée. Au maximum, un seul power-up pourra être présent en jeu. A tester par itération, une autre possibilité pourrait être de permettre la présence de 2 power-ups mais jamais 2 positifs ou 2 négatifs, seulement 1 de chaque.

Les power-ups ne seront récupérables qu'avec la balle. Leur position pourrait être fixe ou suivre un certain pattern. La durée de leur apparition sera à définir, peut-être même liée à la difficulté. De même que la durée de leur effet quand cela a une importance.

Chaque power-up devra avoir un « poids » différent affectant sa probabilité d'être tiré au sort parmi les power-ups existants. Potentiellement différents selon la difficulté. Certains pourraient même être

retirés du « pool » de power-ups disponibles selon la difficulté. Par exemple, les vies supplémentaires pourraient ne concerner que les difficultés zen et facile.

Liste non exhaustive de power-ups possibles :

- « Ball save » : Les bords de la zone de jeu sont protégés. Si la balle touche les bords elle rebondira automatiquement et la protection disparaîtra.
- Augmentation/Diminution de la taille de la raquette (20s)
- Augmentation/Diminution de la vitesse de la balle
- Vie supplémentaire
- Augmentation/Diminution (s'il y a lieu d'être) de la déviation de la balle à chaque rebond (10s)
- Augmentation/Diminution de la vitesse de rotation de la caméra
- Augmentation/Diminution des effets de rotation sur la balle
- Augmentation/Diminution de la taille de la balle
- Multijoueur versus seulement - Laser (10s)

5 - Scoring et classements

Tous les modes de survie en solo auront un système de scoring en plus du temps durant lequel le joueur a survécu. Ces temps et scores seront enregistrés et visibles dans des classements.

La formule de calcul du score restera à définir durant le développement, en tenant compte d'éviter au maximum les comportements de jeu « non fun », de favoriser la prise de risque, etc. Par exemple : Chaque rebond augmente le score et la vitesse de la balle affecte ce gain de score. Le joueur pourrait tenter de faire des passes courtes pour augmenter plus vite la vitesse de la balle afin de caper la vitesse très tôt. La vitesse de rotation de la balle sur elle-même (cf. I-3 dernier §) au moment du rebond pourrait aussi affecter cela. Déclencher le mode « chaos » / « enrage » (défini plus haut (cf. I-1.1)) le plus tôt possible pourrait également être une stratégie.

Idéalement, en faisant du jeu un jeu déterministe et en enregistrant les données nécessaires, on pourrait produire des fantômes de chaque partie, ou à défaut, vérifier que le score et temps envoyé dans le classement est valide afin d'empêcher autant que possible la triche.

Seuls les meilleurs scores et les meilleurs temps du joueur seraient envoyés en ligne. En local, les XX meilleurs scores seraient conservés.

6 - Progression

En dehors de la progression même des réflexes du joueur, une progression pourrait être mise en place sous différentes formes avec des éléments à débloquer en fonction du niveau atteint.

Chaque seconde survécue contribuerait par exemple à faire augmenter une jauge d'expérience, permettant d'augmenter son niveau en jeu. Certains niveaux ne pourraient être franchis sans avoir rempli une condition supplémentaire à l'XP requise, comme par exemple de survivre un certain nombre de secondes dans un niveau précis.

Ce « niveau de joueur » permettrait de débloquer de nouveaux niveaux à défaut de ne pas les avoir débloqués en ayant survécu 100s. Par exemple en imaginant qu'il y aurait en tout 6 niveaux (zones de jeu différentes) dans le jeu avec 3 accessibles de base. Le joueur pourrait débloquer le niveau 4 en survivant 100s dans le 1^{er} niveau, ou en atteignant le niveau de joueur 10 qui nécessiterait un montant spécifique d'XP (donc de secondes survécues dans différentes parties) et seulement 50s survécues dans une même partie par exemple.

L'éventuel niveau de difficulté bonus « custom » pourrait être débloqué avec ce « niveau de joueur » en ayant atteint le niveau maximal (100), ou progressivement les paramètres accessibles pour configurer sa partie se débloquent avec.

Encore une fois, à voir à l'usage. L'intérêt pourrait être trop faible si les récompenses ne suivent pas. Le jeu étant minimaliste, pas vraiment de place aux skins par exemple.

7 - Système de sauvegarde

Plusieurs profils (aka slots de sauvegarde) disponibles. Un pseudo local peut être renseigné pour chaque profil. La sauvegarde est automatique en fin de chaque partie. Le classement local est partagé entre les différents profils existants sur l'appareil où le jeu est installé.

8 - Niveaux

Cf. I-1, plusieurs niveaux apportant des modifications à la zone de jeu seront disponibles pour tous les modes de jeu. Par exemple, la présence de murs sur 2 des côtés parallèles de l'octogone.

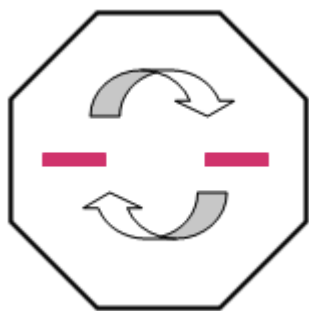
Il n'y en aura pas nécessairement beaucoup, mais suffisamment pour proposer différentes expériences de jeu.

Certains niveaux imposeront certaines contraintes selon le mode de jeu. Par exemple si 2 des 8 côtés de l'octogone sont des murs, il ne pourra pas y avoir 8 joueurs avec chacun une raquette mais que 6 raquettes maximum.

Quelques idées non exhaustives de niveaux :

- Aucun obstacle (niveau par défaut)
- 2/4 côtés parallèles de l'octogone murés
- Carré/Rectangle/Croix au centre (en obstacles fixes, et éventuellement en version mobiles)
- 2/4 rectangles symétriques plus ou moins longs (on peut imaginer 3 longueurs courts/moyens/longs) partant des bords de l'octogone venant séparer en 2/4 la zone de jeu

9 - Obstacles temporaires dans les niveaux



A voir si leur existence même sera ou non justifiée via tests. Il s'agirait d'obstacles/patterns qui viendraient dans la zone de jeu à certains moments dans une partie pour une durée plus ou moins longue.

Par exemple (cf. figure ci-contre), 2 petits rectangles symétriques en rotation (gauche et/ou droite) ancrés au centre du « rayon » de l'octogone.

Ces obstacles devront être correctement signalés quand ils arrivent avec aucune collision le temps de leur apparition complète. Certains patterns ne seront pas compatibles selon le niveau, ou demanderont une gestion supplémentaire pour leur bon fonctionnement. Par exemple les 2 petits rectangles en rotation sur un niveau avec 2 rectangles moyens/longs vont entrer en collision.

10 - Feedbacks

Point important aussi bien pour l'aspect visuel que sonore.

Je tenais juste à préciser ici qu'un des feedbacks importants (en plus de la position de la balle et de la raquette) sera celui de la position du curseur de la souris, sans que celui-ci soit visible durant une partie.

La position du curseur de la souris par rapport au centre de la zone de jeu définit la position de la raquette (ou les dans le mode double). Plus le curseur est proche du centre, plus la précision de la position de la raquette est faible et le risque de « téléporter » la raquette ailleurs est grand (d'où la

zone morte configurable cf. I-3). On pourrait imaginer un filtre de la couleur de la raquette d'une opacité faible avec blend mode additif partant du centre vers la raquette, et proche de l'endroit où le curseur est, une opacité ou luminosité plus ou moins prononcée.

Outre les feedbacks majeurs et vitaux, un jeu se doit d'être « [juiced up](#) » : ball trail, etc.

II - Inspirations, choix artistiques, techniques, cibles et coûts

1 - Choix du moteur

Le jeu sera développé avec [Godot](#). Moteur de jeu libre et open-source, avec un focus sur les jeux 2D à la base, même si la 3D est ensuite arrivée. D'autres choix étaient possibles tels qu'Unity ou GameMaker mais pour avoir effectué différentes recherches, c'est ce qui correspondra le mieux pour le projet.

2 - Développement par itération

Comme indiqué quelques fois dans les paragraphes précédents, le jeu sera développé [par itération](#), dans le sens où il est difficilement possible de trouver les bonnes valeurs à chaque attribut, et comment interagira l'ensemble. La priorité sera bien sûr donnée aux contrôles et à la zone de jeu.

3 - Choix artistiques et inspirations

Le style choisi est le minimalisme. Style s'adaptant parfaitement aux jeux typés arcade avec un gameplay « easy to learn, hard to master ».

Le jeu s'inspire de [Super Hexagon](#) et de Pong. L'aspect survie pour le 1^{er}, le minimalisme pour les 2 bien qu'on se rapprochera évidemment du minimalisme d'un Super Hexagon plus que d'un Pong où il est difficile de faire plus minimal.

4 - Coûts

Le moteur étant gratuit et le style étant le minimalisme, peu de dépenses seront à prévoir du côté des graphismes à part peut-être sur les quelques images pour habiller le site officiel et les différentes plates-formes sur lesquelles le jeu se retrouvera.

Pour la musique, je n'ai pas encore de certitude mais si on prend par exemple [les prix de Zircon \(ou ceux indiqués ici\)](#) on est aux alentours des 3 200€ pour 10 minutes de musique exclusive. Il est également possible de prendre des licences de musique non exclusive, ou de voir avec des compositeurs pas nécessairement très connus.

Pour le sound design, il y a assez peu de sons à produire, on devrait s'en sortir, sinon il y aura toujours moyen de trouver des packs spécifiques pas nécessairement trop chers.

Pour la communication, il s'agira principalement outre la création des comptes sur un ou plusieurs réseaux sociaux et de les faire vivre, de mettre en avant le jeu de différentes façons (via des extraits vidéo par exemple) à différents endroits. Principalement sur Internet, mais peut-être également dans différentes conventions. [Le coût de ces conventions](#) étant très variable.

De plus en plus de jeux se servent de l'accès anticipé (ou « early access »), principalement sur Steam. Cette possibilité peut s'avérer très utile et n'est pas forcément à exclure même si pour un jeu comme celui-ci, l'impact serait certainement assez faible.

5 - Cibles (joueurs et plates-formes)

Le jeu même s'il sera simple à comprendre ne sera pas facile (en tout cas en difficulté normale). Dans la même veine de Super Hexagon dont il sera en partie inspiré.

La plate-forme principalement ciblée sera le PC (Windows, Mac, Linux), puis une sortie mobile (Android et iOS). Il est probablement prudent de ne pas envisager une sortie simultanée des deux. Les consoles ne sont pas envisagées par défaut, Godot ne permettant pas l'export. Ça reste une possibilité à long terme si l'intérêt et l'opportunité se présentait.

6 - Prix de vente

Le prix envisagé pour PC est de 9.99€. Il restera à définir pour les mobiles. Il est potentiellement délicat de mettre un prix plus faible sur mobile. A voir s'il ne serait pas possible d'inclure la version mobile lorsque le jeu est acheté sur PC. Ceci étant, de nombreux jeux ont un prix nettement inférieur sur mobile par rapport au PC, bien que la date de sortie ait probablement une importance ici. Exemple avec Mini Metro à 1.19€ sur Android et à 8.19€ sur PC.