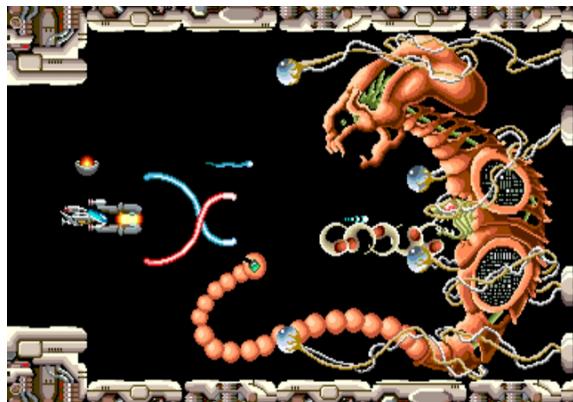


RETYPE

© 1987 BY IREM CORP.


Principe du jeu

Un grand classique des jeux d'arcade : un vaisseau spatial progresse horizontalement dans un univers hostile, assailli par des vagues de chasseurs ennemis, pris pour cible par des tourelles de tir, il doit aussi éviter toute collision avec les éléments solides du décor. Le vaisseau du joueur est libre de se déplacer dans le cadre de l'écran mais le décor avance à vitesse constante en « scrollant » de droite à gauche. Le jeu est constitué d'un enchaînement de niveaux à traverser, chaque niveau se concluant par un « boss » à abattre pour passer au niveau suivant.



Dobkeratops : un boss charismatique



Chaque niveau a un style différent

Divers bonus, à ramasser en passant dessus, permettent de changer d'armement, d'obtenir un bouclier, ou de bénéficier d'autres avantages (vies supplémentaires...). Certains bonus n'apparaissent qu'une fois un ennemi détruit. Les ennemis sont plus ou moins coriaces (points de vie) et certains nécessitent plusieurs tirs ou l'utilisation d'une arme plus puissante (dégâts cumulés \geq points de vie). Il n'est pas indispensable de détruire tous les ennemis pour avancer, mais le score dépend des pertes infligées.

Le joueur dispose de 3 vies, en cas de destruction du vaisseau, le jeu reprend au début du niveau, ou au dernier point de reprise pour les niveaux longs. Dans le jeu original, les bonus (bouclier, missiles ...) sont perdus. Le jeu original était notoirement difficile !

La spécificité de R Type par rapport aux nombreux « shoot'em up » basés sur des principes similaires est dans le bonus « Force pod » qui est un module complémentaire qui peut venir se greffer à l'avant du vaisseau, ou à l'arrière, ou être détaché et devenir partiellement autonome. Le Force pod est indestructible, il peut servir de bouclier pour se protéger, il peut servir de bâlier pour attaquer les ennemis fragiles, il peut envoyer des tirs supplémentaires selon son niveau (upgrades) obtenu par des bonus.

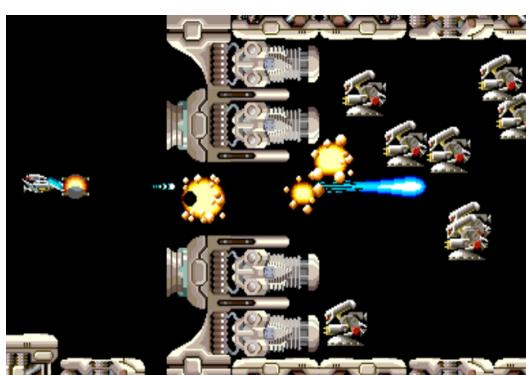


Force pod et lasers à l'arrière



Force pod détaché :tirs latéraux

L'autre spécificité est la jauge de tir : en laissant le bouton de tir appuyé longtemps on charge progressivement la puissance de tir (jusqu'à un maximum). Au relâchement du bouton on obtient un tir unique d'autant plus puissant qu'on a appuyé longtemps. Ce mécanisme de jeu oblige le joueur à trouver la tactique optimale entre beaucoup de tirs légers (répéter des appuis courts) ou de rares salves dévastatrices (laisser appuyer avant de relâcher) pouvant balayer plusieurs assaillants.



Salve puissance max



BEAM = Jauge de charge (version XBox)

Nous vous invitons à faire vos recherches et à passer un peu de temps (pas trop, et pas en TP !) à consulter les captures écran et vidéos (« R-Type longplay » ...) pour apprécier les différents mécanismes de jeu proposés par R Type, en version originale ou dans ses versions plus récentes (si les gros pixels vous font mal aux yeux).

ECE-TYPE

2019, L'empire Bydo

sera de retour, seul un puissant vaisseau ECE-Type (prononcer S-Type) équipé d'un générateur d'énergie 2.0 inventé par de brillants élèves de l'ECE pourra contenir la menace.

2015, votre mission : coder un simulateur de combat permettant de préparer les meilleurs pilotes.

Cahier des charges

Ce projet a pour objectif de vous faire pratiquer la **méthode DTI (Données/Traitements/Interface)**, le **langage C**, la **programmation graphique**, et l'utilisation d'une **librairie : Allegro** (branche 4)

Il vous permet aussi de vous approprier un jeu classique, à votre manière... Il laisse la place à votre imagination et offre des perspectives de jeu qui dépassent largement les quelques illustrations qui ont été données. **L'objectif n'est pas de faire un clone aussi parfait que possible de R Type** mais de vous inspirer de ses mécanismes pour développer votre vision d'un jeu d'arcade à scrolling horizontal. A vous de nous épater en faisant preuve d'ingéniosité et en inventant des situations que nous n'avions pas prévu et qui constituent une véritable valeur ajoutée à votre projet, autant en esthétique qu'en jouabilité.

L'univers et le « story telling » est donc totalement libre à l'intérieur de la mécanique de jeu spécifique au genre : au lieu d'un environnement spatial alien et biomécanoïde, vous pouvez imaginer un univers sous marin où un poisson clown joue le rôle du vaisseau du joueur et des requins lanceurs d'oursins sont les ennemis, le boss étant un gros poulpe sur son rocher. Ou dans un monde fantastique un dragon cracheur de feu à la reconquête de son royaume, ou une sorcière médiévale sur son balais, ou un super globule blanc combattant des virus...

Pour des raisons pratiques de temps et de compétence, vous êtes autorisés à emprunter tout ou partie des graphismes sur Internet ou d'autres sources. Dans ce cas vous devrez citer les sources de ces emprunts sur votre rapport. Il n'y aura donc pas de pénalité à assembler des éléments graphiques tiers. Cependant l'évaluation tiendra compte de la cohérence visuelle (bonne intégration) et valorisera tout effort de personnalisation esthétique, en particulier des graphismes originaux dessinés ou rendus en 3D ou photographiés par vos soins. L'objectif principal restant la programmation, attention de ne pas vous perdre dans les images au point d'oublier le code !

Niveau

Un niveau dure entre 30 secondes (pas moins) et 2 minutes (temps évalué sans « mort » du personnage). Les niveaux les plus longs et les plus difficiles pourront proposer des « points de reprise » où le personnage revient en cas de mort.

Contrainte de volume

Le dépôt à faire sur campus ne devra pas dépasser **50Mo zippé**. Allegro 4 ne prenant pas en charge les formats images compressés, les images .bmp peuvent rapidement peser très lourd. Surveillez ce paramètre en zippant régulièrement vos graphisme. Ce sont surtout les images de fonds qui pèsent

le plus. Avec des dessins simples et réguliers, une collection d'image .bmp de 150Mo peut souvent être compressé à 50Mo. Ce sont les images photo qui se compressent le moins bien...

Travail à faire

Le projet est à faire en trinôme, à l'intérieur d'un même groupe de TP. Les équipes de projet ont été faites par les enseignants.

Vous réaliserez ce projet en respectant la méthode de conception Données/Traitements/Interface apprise : conception puis réalisation.

Vous écrirez le code correspondant à la conception DTI en reproduisant sous forme de commentaires l'ACD du contrôleur. Vous structurerez votre code en sous-programmes.

Vous remettrez, à l'occasion de la soutenance, un **bilan écrit** par équipe qui contiendra des schémas de vos structures de données (structs, tableaux, listes chaînées éventuelles...), des schéma géométriques (repères, vecteurs) seront appréciés, votre ACD, le graphe d'appels de vos sous-programmes en précisant les entrées et les sorties, quelques captures écrans, ainsi que le bilan collectif du fonctionnement de votre équipe, et le bilan individuel de votre implication, de vos difficultés et de vos progrès via ce projet.

1. Jeu de base (15 points)

Votre jeu devra au minimum contenir 3 niveaux. **Un seul boss est exigé, à la fin du dernier niveau.** Quand on arrive sur un boss, le scrolling (défilement du décor) s'arrête et il faut l'abattre pour passer.

Votre jeu devra disposer d'un point d'accueil au démarrage, menu, permettant de lancer une nouvelle partie. A la fin d'une partie, game-over ou jeu terminé, on retourne au menu. Le programme ne se termine jamais brutalement ! On peut relancer une nouvelle partie. On peut également abandonner à tout moment la partie en cours et revenir au menu en appuyant sur la touche échap. On pourra aussi figer le jeu en cours de partie en appuyant sur la touche p (pause) et reprendre en ré-appuyant sur p.

Il est indispensable de prévoir des raccourcis menu pour accéder directement au début de chacun des niveaux, ainsi qu'à la scène finale du boss, pour pouvoir nous montrer tous les éléments de votre jeu pendant la soutenance, même en cas de fébrilité ou de maladresse du joueur au clavier.

Le premier tableau, imposé, doit correspondre à environ **1 minute de progression** (temps évalué sans mort du vaisseau) dans un décor varié avec des ennemis pas trop coriaces et des obstacles pas trop difficiles à négocier. Il peut contenir des bonus (armement, bouclier...) si votre jeu en comporte. Il sert de didacticiel au joueur : il doit être ludique et de difficulté bien dosée. **Il vous servira également de démo des capacités principales de votre moteur de jeu (animations multiples, scrolling, collisions...).** Les aspects plus avancés et expérimentaux de votre jeu sont à réservé aux niveaux suivants.

Font partie du cahier des charges de base la gestion correcte des aspects suivants :

- Scrolling, navigation du vaisseau du joueur avec touches directionnelles, détection des collisions avec des "murs".
- Tirs du joueur multiples, plusieurs projectiles en même temps à l'écran
- Des ennemis volants (mobiles), des ennemis fixes (fixés sur le décor)
- Des tirs ennemis, trajectoires rectilignes uniformes, depuis les ennemis volant et depuis les ennemis fixes.
- La détection des collisions de tirs : tirs_joueur/ennemis, tirs_ennemis/joueur, et les conséquences : accumulation de points de dégâts, destruction éventuelle.
- La détection des collisions vaisseau du joueur/ennemis volants : le vaisseau du joueur explose.
- Des explosions quand il faut et où il faut.
- Des allocations et des chargements maîtrisés : dans l'explorateur de processus votre application ne doit pas peser plus lourd après 3 parties qu'après une seule, pas de fuites mémoire.

2. Extensions (5 points)

Les extensions se jugeront sur l'originalité, le nombre et le type de bonus développés : Bouclier/Force pod ou équivalent, jauge de tir (puissance de tir réglable), armes multiples... vous pouvez vous inspirer des extensions au vaisseau de R-Type (laser rebondissant, lance flamme, missiles, tir en diagonales...) ou emprunter des idées d'autres jeu (tir automatique cadence rapide, mégabombes...) ou inventer les vôtres.

Autres extensions au cahier des charges de base valorisées :

- Des ennemis particulièrement sophistiqués : trajectoires complexes, collimation des tourelles de tirs (viser le joueur), missiles autoguidés, comportements adaptatifs, schémas corporels articulés.
- Des boss supplémentaires ou des boss intermédiaires mobiles qui suivent le vaisseau du joueur à travers une partie d'un niveau.
- Des décors destructible : il faut casser des murs pour se frayer un chemin.
- Des effets spéciaux spectaculaires : explosions animées, traces d'impacts ou prise en compte des dégâts sur l'apparence, boules de plasma de tailles variables...

- 2 vaisseaux joueurs simultanés en coopération : 2 joueurs sur un même clavier, chacun ses touches.
- Des options de configuration : laisser le joueur paramétrer ses touches de jeu préférées, avec sauvegarde, par exemple.
- Des outils de conception auxiliaire tels que éditeur de niveau, éditeur de trajectoire, outils de prétraitement graphique ...
- L'utilisation (pertinente) d'aspects plus avancés de la librairie Allegro : transparences progressives (canal alpha), polices de caractères plus jolies que celle par défaut, utilisations de timers avec découplage du temps du moteur de jeu du temps d'affichage (temps réel précis), ou toute autre possibilité de la librairie nécessitant de creuser les docs en anglais.
- Sonoriser : Allegro peut déclencher des effets sonores, à synchroniser avec les événements du jeu (tirs, explosions...). Attention : Allegro (version 4 imposée) utilise un format .wav non compressé. Une douzaine d'effets sonore de quelques secondes chacun représente un volume acceptable. Un morceau musical (musique de fond) de 5 minute pèse trop lourd dans ce format. Une musique de fond donne une "plus value esthétique indéniable" mais pose ce problème technique, et ne présente que très peu d'intérêt en terme de programmation (3 lignes de code). La librairie fmod permet de charger et de jouer des formats .mp3 plus compacts, possibilité à discuter avec votre chargé de TP.
- Toute autre idée qui vous semble intéressante peut être discutée avec votre chargé de TP !

Robin Fercq

17/03/2015

Version préliminaire sujette à ajustements mineurs ultérieurs en cas de besoin.

