

Ece-prits



სპრიტს
ტექნიკური

Principe du jeu

Spirits est un jeu récent (2012) inspiré du jeu Lemmings (1991). La mécanique de jeu est similaire : Le joueur doit aider des personnages à traverser un niveau en 2 dimensions, rempli d'obstacles et de pièges. Les personnages sont autonomes mais pas très intelligents, ils marchent dans une certaine direction, font demi-tour quand ils rencontrent un obstacle infranchissable, et c'est à peu près tout. Face à un trou, ils continuent d'avancer et tombent... comme, selon la légende, les lemmings (petits rongeurs arctiques) se jetant du haut des falaises.

Le joueur ne peut pas directement prendre le contrôle d'un personnage pour le diriger, mais peut accorder à tout personnage sélectionné une capacité (parmi un répertoire limité...) : creuser le sol ou les murs, construire un escalier, modifier le déplacement des autres... Les capacités disponibles ne permettent pas d'accompagner chaque individu à travers le décor, il faut donner les bonnes capacités aux bons moments et aux bons endroits pour mettre en place un ou plusieurs chemin(s) pour tous les autres personnages qui arrivent ensuite. Une seule erreur de *timing* ou de placement et la petite troupe se jette dans une zone mortelle ou dans un endroit d'où elle ne pourra plus sortir. Il faut alors recommencer le niveau. Pour chaque niveau il y a un nombre minimum d'individus à conduire sains et saufs à la sortie pour passer au niveau suivant, mais le joueur peut essayer d'en sauver plus pour améliorer son score.

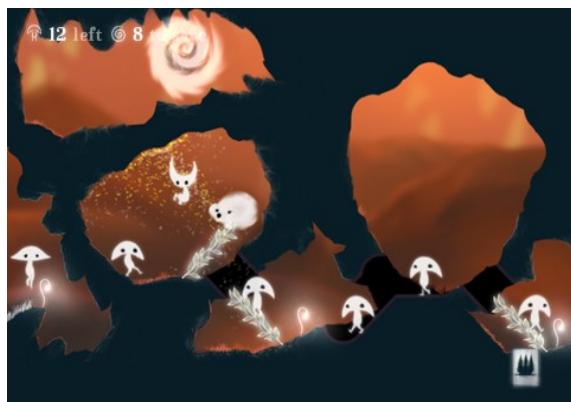
Ces principes généraux ont été inventés dans le jeu Lemmings et se retrouvent dans la ré-interprétation vidéo-ludique récente Spirits. C'est de cette dernière incarnation des principes de jeu que s'inspire le projet Ece-prits qui vous est proposé cette année. **Il ne vous est pas demandé de fournir une copie conforme du jeu Spirits** : nous souhaitons voir des propositions originales mais qui respectent une mécanique de jeu similaire. Le cahier des charges de base doit vous guider et vous permettre de montrer que vous savez atteindre un ensemble d'objectifs techniques précis. Deux semaines de TP (semaines du 07/03 et du 14/03) sont prévues pour vous encadrer dans la conception et le démarrage du projet. Un énoncé de TP orienté projet sera rédigé dans ce sens.

Cahier des charges de base : 15 points

Le jeu se déroule dans un univers en 2 dimensions vu de coté, avec des éléments solides (sols, plateformes, plans inclinés, murs...) en avant plan, et un fond qui ne sert que de décor. Des personnages ayant tous un même aspect se déplacent simultanément et de manière autonome en prenant en compte les éléments solides de l'avant plan : ils marchent sur les plateformes, grimpent sur les plans inclinés, font demi-tour quand ils atteignent un mur ou une pente trop forte, tombent sur une plateforme inférieure si ils avancent dans le vide. Les personnages ne s'arrêtent jamais, ils sont toujours en mouvement, si ils se retrouvent bloqués dans un trou il font du va-et-vient entre les parois. **Votre jeu doit pouvoir animer simultanément au moins 10 personnages.**

Pour chaque niveau l'objectif est d'amener une partie des personnages d'une entrée, l'endroit où les personnages apparaissent 1 par 1 à un rythme régulier, vers une sortie située à un autre endroit du terrain, en dotant certains d'entre eux (interactivement à la souris) de capacités spéciales. Le nombre total de personnages qui vont apparaître à l'entrée, l'emplacement de l'entrée et de la sortie, le nombre minimum de personnages à amener (intacts!) à la sortie, les capacités disponibles, et bien sûr l'aspect et la géométrie du terrain **varient d'un niveau à l'autre**.

Le jeu dans son ensemble se compose d'une succession de niveaux. Quand le nombre minimum prévu de personnages à atteint la sortie, le passage au niveau suivant est débloqué : le joueur peut déclencher ce passage ou continuer à jouer sur le même niveau pour faire passer plus de personnages. A tout moment le joueur qui pense qu'il est bloqué peut relancer la partie au même niveau. Il est demandé de réaliser **3 niveaux de jeu minimum**. Il n'y a pas de maximum mais votre jeu (l'ensemble des niveaux qui se succèdent) doit pouvoir être visité en 6 minutes à peu près.



*Un exemple de niveau de jeu.
Ici la sortie est en haut à gauche*



*Un autre niveau de jeu.
Ici la sortie est en bas à droite*

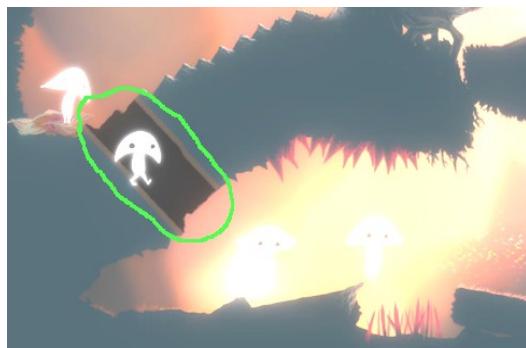
Le 1^{er} niveau servira de niveau d'initiation, seules les 3 capacités du CDC de base (voir ci-après) devront être activables, chacune au moins une fois, il y aura 10 personnages entrés et il faudra en faire sortir au moins 5 pour débloquer le passage au niveau suivant. Ce 1^{er} niveau devra être peu difficile afin de permettre à tout joueur (ayant compris le principe) de le réussir. Les niveaux suivants sont libres et serviront autant à illustrer la variété de situations que peut gérer votre « moteur de jeu » qu'à montrer le potentiel ludique de votre projet (si vous aviez le temps de faire 50 niveaux !).

Zones mortelles : il doit y avoir au moins un type de « zone mortelle » présent sur chaque niveau, visuellement identifiable par le joueur. Le contact d'un personnage avec une telle zone conduit à sa destruction. Dans le jeu Spirits ces zones mortelles sont représentées par des buissons de piquants. Dans l'image ci-dessous un exemple de zone mortelle, entourée en rouge :



3 capacités obligatoires (qui devront figurer au 1^{er} niveau de jeu) ...

Capacité de creusement : un personnage qui se voit affecter cette capacité commence à creuser le décor solide de l'avant plan, permettant aux autres personnages de passer par le tunnel ou le trou ainsi dégagé. A l'issu de la phase de creusement (qui s'arrête quand il n'y a plus rien à creuser) le personnage peut reprendre son comportement usuel ou disparaître (à vous de choisir selon ce qui vous semble le plus intéressant pour votre jeu).

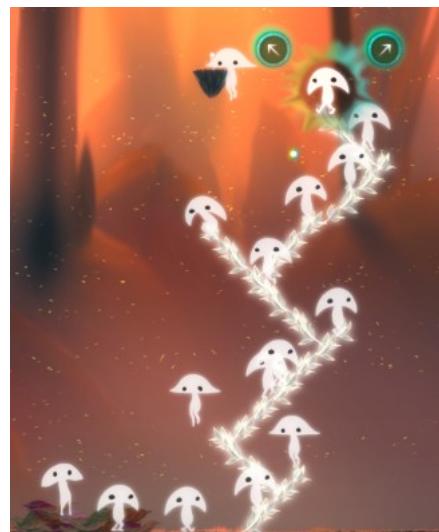


Tunnel diagonal creusé dans le décor



Le Lemmings historique, tunnel horizontal...

Capacité de construction : un personnage qui se voit affecter cette capacité commence à construire un plan incliné (escalier dans Lemmings, branche dans Spirits). A l'issu de la phase de construction (qui s'arrête quand la construction atteint une certaine taille) le personnage peut reprendre son comportement usuel ou disparaître (à vous de choisir selon ce qui vous semble le plus intéressant pour votre jeu). Il est possible de construire sur une construction (voir ci contre) mais pas de construire dans le vide avec un personnage en train de tomber ! La diagonale n'est pas obligatoire, vous pouvez préférer ponts horizontaux et/ou échelles verticales.



Capacité de souffle ou d'entraînement : principale innovation de Spirits par rapport à Lemmings, on peut transformer un personnage en « souffleur » qui crée un vent qui entraîne les autres personnages qui passent dans la zone du souffle. C'est sans doute la capacité la plus difficile à réaliser pour le cahier des charges de base, des indications complémentaires seront données en cours et en TP pour vous aider. Dans le cahier des charges de base il n'est pas exigé que les souffles générés par les souffleurs soient arrêtés par les obstacles, il n'est pas non plus exigé que la rencontre de plusieurs souffles donne un résultat réaliste comme dans l'exemple ci dessous où la rencontre de 2 souffles en diagonal donne un souffle vertical (difficile !). Dans ce genre de situation on pourra accepter un compromis technique comme par exemple le fait que le dernier à souffler « écrase » le souffle du 1^{er}. La zone où un souffle est actif sera visible pour le joueur, avec un effet visuel statique ou dynamique.



Pour chacune des 3 capacités du CDC de base, vous devez autoriser au minimum 2 directions distinctes, soit selon un choix direct de l'utilisateur grâce à l'interface, soit en fonction du sens de déplacement du personnage au moment de l'activation de la capacité. Les modalités d'interaction sont à votre libre appréciation : vous pouvez choisir un système de barre de boutons sous l'espace de jeu pour sélectionner les capacités (comme dans Lemmings) ou avoir un menu contextuel qui s'affiche au niveau du personnage sélectionné (comme dans Spirits). La seule contrainte d'interface est la possibilité de jouer uniquement à la souris. Les capacités sont données aux personnages un par un : un seul personnage à la fois est sélectionné.

Pour un niveau donné, la disponibilité des capacités peut être en tout ou rien : par exemple dans Spirits sur certains niveaux la capacité de creusement n'est pas proposée du tout, en revanche les capacités possibles sont en quantité illimitée. Vous pouvez aussi choisir une approche par « budget » comme dans Lemmings : il n'y a alors qu'une quantité limitée (éventuellement 0) d'utilisation pour chaque capacité, les quantités limitées proposées rendent chaque niveau intéressant.

Les autres possibilités accessibles par le joueur en cours de partie seront :

- Recommencer le niveau
- Passer au niveau suivant (dès qu'on a rempli l'objectif du niveau)
- Accélérer le rythme d'arrivée des personnages (sans accélérer le rythme du jeu)
- Accélérer le rythme du jeu (exceptionnellement on peut altérer le paramètre du **rest** ...)
- Retourner au menu (abandonner la partie en cours)

Autres consignes de réalisation du CDC de base :

- Les personnages sont animés (séquences d'images, au moins rudimentaires)
- Les personnages ont des aspects/animations différents selon leur capacité en cours
- Le programme démarre sur un menu, le menu permet d'accéder directement à un niveau de jeu déjà débloqué. Par commodité pour la soutenance il devra être possible d'éditer directement un fichier pour débloquer tous les niveaux, ou alternativement d'avoir un système de mot de passe ou *cheat-code* pour accéder aux niveaux pas encore débloqués...
- Le programme se quitte uniquement par l'intermédiaire du menu (bouton à cliquer)
- On peut jouer plusieurs parties sans quitter le programme (par retour au menu)
Il est exclu de gérer l'enchaînement menu → partie → menu → partie... par appels récursifs.

Extensions : 4 points

Le jury évaluera le CDC de base sur tous les points sus-mentionnés, tout manquement ou réalisation trop approximative sera pénalisé au barème de 15 points prévu. Le barème des extensions sur 4 points est séparé : vous ne pouvez pas compenser des manquements au CDC de base en collectionnant les extensions au delà de 4 points. Le jury se réserve le droit de ne pas considérer du tout les extensions si le CDC de base est trop déficient. Par exemple un jeu sans capacité de souffle, capacité de creusement qui marche mal, avec des personnages pas animés, et qui plante après une minute de jeu, les extensions seront ignorées car ce n'était clairement pas la priorité de l'équipe...

La liste suivante n'est pas exhaustive, il n'est pas nécessaire de réaliser toutes ces idées pour obtenir les 4 points d'extension, discutez avec votre chargé de TP des autres idées que vous pourriez avoir ! Les points (ou ½ points) accordés par le jury dépendront de la difficulté et de la qualité de réalisation.

Scrolling ou zoom+scrolling : le scrolling est le défilement d'un grand terrain de jeu dans un espace d'affichage (cadre écran) plus petit, seule une partie du grand terrain de jeu est visible. Attention, la mécanique du jeu continue de se dérouler normalement dans les zones hors du cadre visible ! Le scrolling peut se faire à échelle fixe (c'est le cas dans le jeu Lemmings) ou à échelle variable (c'est le cas dans Spirits) c'est à dire que le joueur peut voir tout le terrain ou zoomer dans une zone.

Carte : en relation avec un scrolling sans zoom il peut être intéressant de proposer une carte dynamique (ou « radar ») en vignette dans un coin de l'écran de jeu → voir le jeu Lemmings.

Effets visuels évolués, effets de particules : un soin particulier est donné à la représentation des zones de souffle par exemple. Dans Spirits les souffles sont représentés par des nuages de particules qui volent dans le courant d'air. Comptent également comme effets visuels évolués des explosions, des effets de flammes, d'aura, d'onde, des animations du décor, en fonction de l'univers de votre jeu...

Gestion avancée de la capacité de souffle : valorisation pour tout travail au delà du strict CDC de base pour la réalisation de l'interaction entre souffle et terrain (un mur ou une plateforme du terrain bloque/réoriente un souffle) ou entre souffles (souffles convergents) ou de la disparition du souffle (votre interface autorise la destruction d'un souffleur : son souffle disparaît)

Gestion avancée des trajectoires de vol : valorisation pour tout travail au delà du strict CDC de base pour l'obtention de trajectoires complexes pour les personnages qui chutent ou qui sont entraînés dans un souffle, en particulier effets d'inertie bien maîtrisés (accélérations). Voir par exemple l'impression de légèreté des personnages/feuilles qui volent dans Spirits...

Terrain dynamique : votre terrain de jeu bouge, par exemple une presse hydraulique monte et descend sur le trajet des personnages, ceux qui passent au mauvais moment sont écrabouillés...

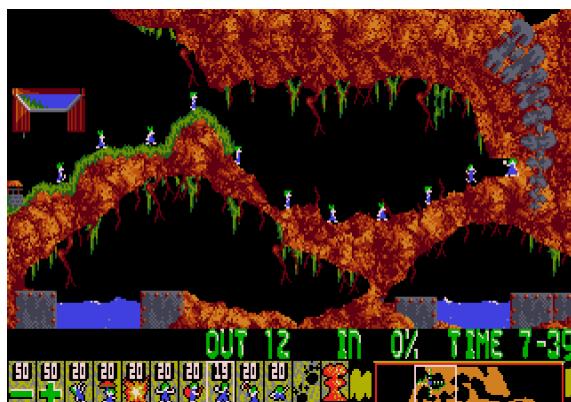
Objets à ramasser : comme dans Spirits vous pouvez ajouter un ensemble d'objets à prendre, soit directement (un personnage arrive dessus) soit indirectement : dans Spirits on peut envoyer une graine en tapant sur un personnage qui vole, si la graine tombe sur un objet celui-ci est collecté.

Capacités supplémentaires : vous pouvez ajouter d'autres capacités en plus des 3 du CDC de base. Exemples inspirés de Lemmings : explosion (avec destruction du décor), grimper, bloquer ...

Effets sonores interactifs : des bims, des bouums, des arghh ! Bien synchronisés avec l'action bien sûr. Il est aussi possible d'ajouter une musique de fond (voir format .ogg avec Allegro 4.4.2) mais n'espérez pas avoir beaucoup de points pour juste un seul load_sample et play_sample...

Originalité et cohérence : 1 point (+1 point bonus exceptionnel)

Le jury appréciera l'originalité de votre proposition. Essayez de déplacer la mécanique de jeu imposée dans un univers différent, adaptez le récit et le gameplay à cet univers pour en faire un tout cohérent et harmonieux. De ce point de vue votre liberté est complète. Le modèle du genre c'est la ré-interprétation du jeu Lemmings faite dans Spirits : l'Univers est très différent, une nouvelle capacité est inventée et devient le cœur du jeu, le tout s'accorde parfaitement à l'ambiance graphique et à la nouvelle histoire. On sent qu'une attention particulière a été donnée au game design de Spirits, par exemple pour trouver des alternatives plus douces et poétiques aux morts atroces des Lemmings...



Lemmings



Spirits

A titre d'illustration voici 3 propositions d'univers possibles (essayez d'en trouver d'autres !) :

Univers	Personnages	Creuser	Construire	Souffler	Spécial	Mortel
Technologique Base lunaire	Robots	Au laser	Ascenseurs (haut ou bas)	Répulsion magnétique	Inversion de gravité	Champs d'étincelles
Aquatique	Petits poulpes	Transforme en vrille...	Déroule tenta- cules diagonale	Courant d'eau	Bloqueur	Oursins, Anémones
Tolkien, mines de la Moria	Nains	A la pioche	Pont de pierre	Magie Elfique, sort de zone	Catapulte à nains...	Eau, Chute trop haut

Dans le point d'originalité est aussi compté (sans caractère obligatoire) un travail de design graphique personnel, valorisé par rapport à la simple utilisation de graphismes empruntés (voir ci-dessous). Très exceptionnellement 1 point supplémentaire de bonus est laissé à l'appréciation du jury (talent prodigieux à la palette, images de synthèse sensationnelles...) Attention ceci ne doit pas vous faire oublier qu'il s'agit avant tout d'un projet de programmation : chaque membre de l'équipe doit participer au code, pouvoir expliquer en détail les parties sous sa responsabilité et de façon synthétique le reste du code source, sans se retrouver perdu.

Consignes générales :

- Il est autorisé d'utiliser des graphismes tiers trouvés sur Internet, si possible libres de droits (*Creative commons...*) ou avec *Copyright*, dans ce dernier cas votre utilisation du logiciel doit rester strictement dans le cadre scolaire (tolérance citation et travail d'étude) et ne doit pas être publié, les sources doivent être rigoureusement citées dans un fichier **licence.txt** dans votre archive.
- Il est autorisé d'utiliser/adapter les codes sources publiés sur le site principal du cours, ainsi que des **extraits** de code source libre de droit (*public domain*, GPL, LGPL ...) en précisant ces emprunts. Les commentaires initiaux de ces codes sources empruntés doivent être mis à jour par vos soins pour refléter leur utilisation dans votre contexte (appropriation correcte des codes tiers utilisés). En soutenance vous devez être capable d'expliquer chaque ligne de code de votre projet déposé.
- Il n'est pas autorisé d'utiliser une librairie complète ou un « *framework* » en dehors des versions proposées de Allegro sur le site principal du cours (4.2.2 ou 4.4.2) et des bibliothèques standard du C (stdio, stdlib etc...). Il n'est pas autorisé de copier massivement un code tiers : l'objectif est de réaliser votre moteur de jeu, pas de se brancher sur un moteur de jeu pré-existant.
En dehors des exemples du cours, toute détection de copie massive de code tiers sera considéré comme plagiat et très sévèrement sanctionné, avec circonstances aggravantes si
 - l'emprunt n'est pas cité
 - l'emprunt vient du travail d'une autre équipe d'étudiants de l'ECE pour ce même projet
 - il y a tentative de dissimulation (maquillage de code...)
- Le code source doit être présentable : bien indenté, raisonnablement aéré et commenté. Pour rappel, un bon commentaire ne paraphrase pas le code (// xp prend la valeur 10) mais explique à quoi sert le code (// L'abscisse du perso est mise à gauche de l'écran). L'ensemble doit être structuré : sous-programmes regroupés dans des .c et .h par thématique, structures de données adaptées aux problèmes à traiter, constantes nommées, éviter autant que possible les duplications de codes similaires au profit de boucles, tableaux, sous-programmes, paramètres.

Contrainte de volume

Le dépôt à faire sur campus ne devra pas dépasser **50Mo zippé**. Allegro 4.2.2 ne prend pas en charge les formats images compressés, les images .bmp peuvent rapidement peser très lourd. Surveillez ce paramètre. Ce sont surtout les images de fonds et « cartes de collision » qui pèsent le plus. Avec la version 4.4.2 mise à disposition sur le [site principal](#) du cours vous pouvez utiliser des images en format .png (préférable pour les graphismes « lisses ») et pour conserver des couleurs exactes : cartes de collisions) et/ou des images en format .jpg (à privilégier pour des images bruitées comme des photos numériques). Il ne s'agit pas d'une obligation, vous pouvez rester sur du .bmp si le volume ne pose pas de problème.

Voir les options de compilation (Project → Build options... → Linker settings) de l'exemple de projet check_alleg_4_4_png_jpg de allegro_checking.zip pour compiler un projet capable d'utiliser les formats .jpg et .png.

Travail à faire

Le projet est à faire en trinôme ou binôme, 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. Tenez à jour, **dès le départ de la conception**, un « cahier de laboratoire » : 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.

La date de soutenance est fixée pour la semaine du 11 Avril, la date exacte et les modalités de rendu vous seront précisées rapidement. Le planning prévoit que le projet soit réalisé sur 4 semaines à partir de la semaine 14 Mars, toutefois vous aurez une 1ère séance de TP orienté projet dès la semaine du 07/03, durant laquelle vous pourrez poser toutes vos questions à votre chargé de TP et commencer à pratiquer les techniques de programmation spécifiques au projet (terrain altérable, zones de souffle).

Robin Fercoq et toute l'équipe encadrante,

28/02/2016

Version 1, sujette à ajustements mineurs en cas de besoin.