



SQUIRREL DRAMA



UNIVERSITÉ
LAVAL

TP3 – Préproduction

Par équipe 03:

Nicolas Beaulieu
Frédéric Bélanger
Henri Bernard-St-Laurent
Alexis Normand-Bélanger
Mathis Rhéaume
François Veilleux-Lessard

collaborateur externe
536 897 267
111 285 905
536 921 175
536 897 922
111 113 219

Présenté à :
François Chéné

Dans le cadre du cours :
IFT-3113 – Projet de jeu vidéo

Remis le :
24 février 2023

Table des matières

Squirrel Drama	4
Expérience de jeu	4
Esthétiques	4
Intégration du thème	4
Inspirations	4
Boucle de jeu	5
Mécaniques prévues	6
Objectifs	6
Déplacements	6
Enchaînement des niveaux	6
Obstacles	6
Gestion des vies	6
Écureuil animé	6
Ressources	7
Modèles 3D non texturés	7
Modèles 3D texturés	7
Modèles 3D texturés et animés	7
Effets de particule	7
Images 2D	7
Sons et musiques	7
UI	8
Arrière-plan 3D	8
Plateforme technologique	9
Architecture logicielle	10
Présentation de l'équipe	11
Nicolas Beaulieu	11
Frédéric Bélanger	11
Henri Bernard-St-Laurent	11
Alexis Normand-Bélanger	11
Mathis Rhéaume	12
François Veilleux-Lessard	12
Annexe 1 — analyse technologique	13
Modèle 3D avec textures	13

Animation 3D	14
Ascenseur et fruit	15
Tests sonores	16
Annexe 2 — rapport d’avancement 1	17
Annexe 3 — galerie de production	25
Arbre de l’écureuil	25
Petit fruit rouge	26
Abeille	27
Patte d’ours	28
Menu principal (logo, UI et arrière-plan)	29
Annexe 4 — prototype supplémentaire	31
Génération procédurale	31
Annexe 5 — lien vers le code source	32

Squirrel Drama

Jeu d'action 2.5D de type « pinball » vertical, où l'on doit faire appel à la physique pour remonter des fruits sphériques le long d'un tronc d'arbre en évitant des obstacles.

Expérience de jeu

Squidgy, l'écureuil maladroit, a perdu sa réserve de nourriture. En voulant la récupérer, il est tombé et s'est fracturé une patte. Incapable de bouger, il est pris en haut de son arbre. Il faut trouver un moyen de remonter ses petits fruits pour l'aider dans sa convalescence et faire en sorte que sa pauvre patte se répare rapidement.

Cottonfield, le lapin magicien, peut utiliser ses pouvoirs télépathiques pour faire léviter une branche morte sur laquelle repose un fruit. Le joueur doit utiliser cet ascenseur improvisé afin de remonter la nourriture le long de l'arbre. Mais cette opération n'est pas aussi simple qu'elle en a l'air. Plusieurs obstacles font obstruction aux passages de la précieuse cargaison ! Abeilles malignes, vers gluants et écorce noueuse ne sont que quelques exemples des périls potentiels.

Pour éviter tous ces obstacles, il faudra jouer avec les hauteurs des deux extrémités de l'ascenseur et ainsi faire rouler le fruit transporté. Saurez-vous faire preuve de suffisamment d'agilité pour sauver ce pauvre écureuil en péril et mettre fin à son drame ?

Esthétiques

- **Défi** : agilité et rapidité sont nécessaires afin d'éviter les obstacles sans avoir le contrôle direct de l'objet déplacé.
- **Sensation** : personnage animé original et ludique (écureuil blessé et lapin magicien) réagissant aux actions du joueur

Intégration du thème

Le thème « récupération et réparation » est intégré dans l'histoire et l'objectif du jeu. L'écureuil doit être nourri afin que sa convalescence se déroule bien et que son os fracturé se répare. Pour y arriver, il faut récupérer les fruits qui sont tombés au bas de l'arbre. On récupère aussi une branche morte afin de la convertir en ascenseur à nourriture.

Inspirations

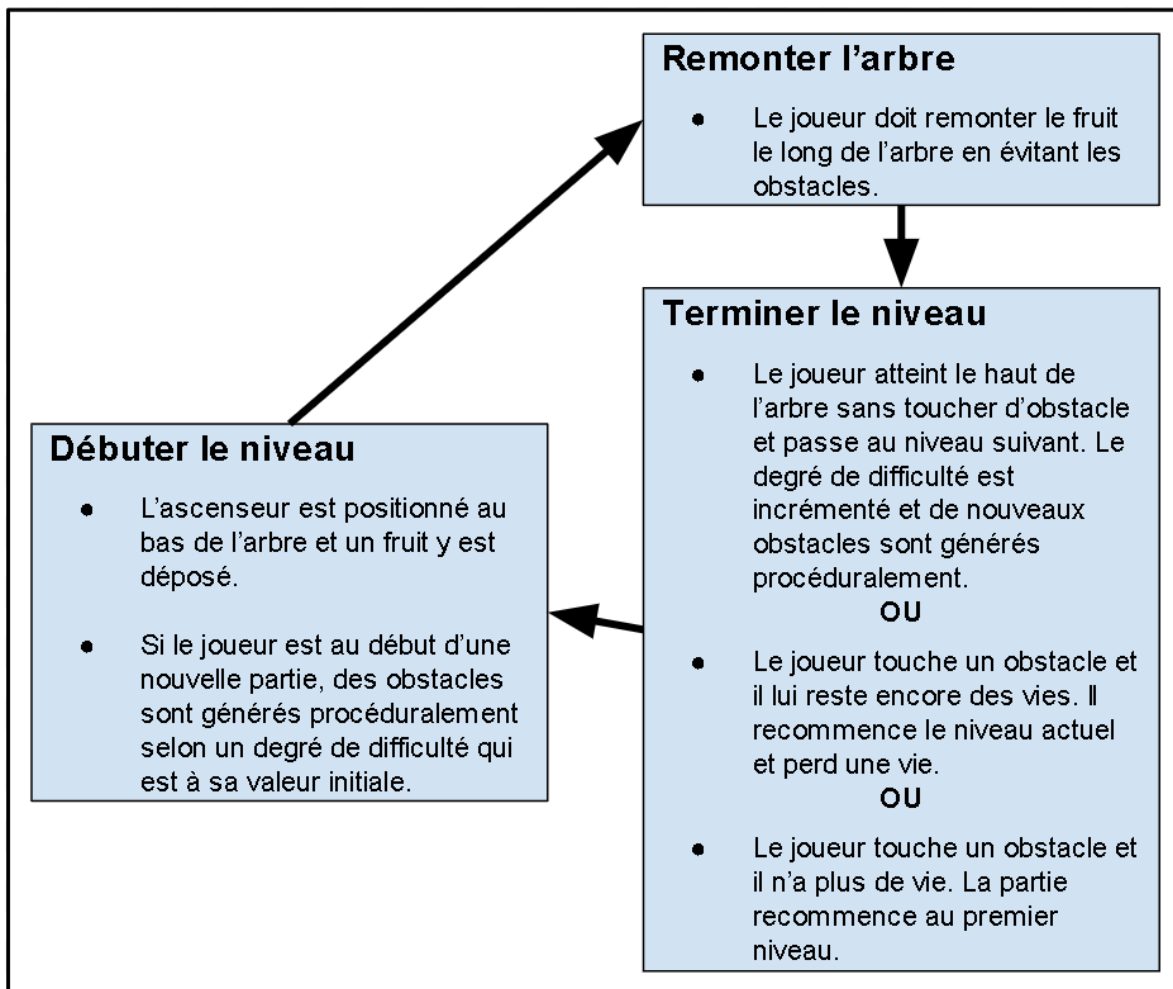
La mécanique de base du jeu est inspirée de Ice Cold Beer, un cabinet d'arcade mécanique de type « pinball » vertical créé par Taito. Le côté dramatique et ludique de l'écureuil et du lapin sont influencés par le personnage principal de la série de jeux vidéo Conker, créé par Rare, ainsi que par Scrat, un personnage des films d'animation Ice Age de Blue Sky Studios.

Le style artistique minimaliste et la composition des obstacles ont été inspirés par le jeu de rôle Bug Fables : The Everlasting Sapling, développé par Moonsprout Games.

Boucle de jeu

La boucle de jeu de Squirrel Drama comporte trois phases. Au début d'un niveau, l'ascenseur est mis à sa position initiale au bas de l'arbre et on y dépose un fruit. S'il s'agit du commencement d'une nouvelle partie, le degré de difficulté est fixé à sa valeur initiale et des obstacles sont générés procéduralement. Ensuite, le joueur utilise l'ascenseur pour remonter le fruit le long de l'arbre en évitant que celui-ci ne touche à un obstacle. Puis le niveau prend fin. Il peut se terminer de trois façons :

- Si le joueur atteint le haut de l'arbre sans toucher d'obstacle, il passe à un niveau suivant. Le degré de difficulté est incrémenté et les obstacles sont à nouveau générés procéduralement.
- Si le joueur touche à un obstacle et qu'il lui reste des vies, il recommence le même niveau, mais avec une vie en moins.
- Si le joueur touche à un obstacle et qu'il ne lui reste plus de vie, il commence une nouvelle partie au premier niveau.



Mécaniques prévues

Objectifs

L'objectif du joueur est de remonter le fruit au haut de l'arbre sans toucher aux obstacles.

Déplacements

Le joueur peut mouvoir verticalement et indépendamment les deux extrémités de l'ascenseur à l'aide des deux joysticks d'un « gamepad » ou encore avec quatre touches du clavier.

Un fruit sphérique est déposé sur l'ascenseur. La position verticale du fruit est modifiée selon celle de l'ascenseur. Selon l'angle d'inclinaison de ce dernier, le fruit peut également rouler vers les extrémités et ainsi se déplacer horizontalement. Le joueur contrôle donc indirectement la position du fruit par le biais de l'ascenseur.

Enchaînement des niveaux

En début de partie, le joueur a un certain nombre de vies et le niveau de difficulté est à sa valeur initiale.

À chaque début de niveau, l'ascenseur est positionné au bas de l'arbre et un fruit y est déposé. Le joueur utilise ensuite l'ascenseur pour tenter de remonter le fruit en évitant les obstacles. Si le fruit touche un obstacle, il est détruit. Le joueur perd alors une vie. S'il n'a plus de vie, il recommence la partie. Sinon, il recommence le niveau.

Si le joueur parvient au haut de l'arbre, le degré de difficulté est augmenté et un nouveau niveau est généré.

Obstacles

La version démo du jeu est constituée de quatre obstacles qui font obstruction au passage du fruit. Il y a des trous sur le tronc dans lesquels le fruit peut tomber, des abeilles volant horizontalement, des vers qui sortent des trous et un ours qui donne des coups de patte.

Gestion des vies

Le joueur débute une partie avec un certain nombre de vies. Il peut en accumuler davantage en ramassant des bonus répartis le long des niveaux. Les vies restantes du joueur sont transférées d'un niveau à l'autre.

Écureuil animé

Afin de divertir le joueur, l'écureuil blessé qui doit être nourri apparaît sporadiquement à l'écran pour réagir à ses actions. Par exemple, lorsqu'un fruit est détruit, l'écureuil réagit dramatiquement et fait semblant de mourir.

Ressources

Modèles 3D non texturés

- 1 Cylindre noir (trous dans le tronc de l'arbre)
- 4 prismes rectangulaires invisibles pour la gestion des collisions
- 2 sphères invisibles pour servir de points d'ancrage aux extrémités de l'ascenseur

Modèles 3D texturés

- 1 petit fruit rouge
- 1 petit fruit vert
- 1 branche d'arbre servant d'ascenseur

Modèles 3D texturés et animés

- 1 abeille jaune avec 1 animation :
 - Battement d'ailes
- 1 vers avec 2 animations :
 - Yeux qui clignent (pour signaler une sortie imminente)
 - Étirement et mouvement circulaire (le vers qui sort du trou)
- 1 patte d'ours avec 1 animation :
 - Réaction à l'impact avec le tronc de l'arbre
- 1 écureuil ayant une patte dans le plâtre avec 4 animations :
 - Mort dramatique
 - Extase devant le fruit
 - Impatience
 - Peur
- 1 lapin magicien avec 1 animation :
 - Télépathie

Effets de particule

- 1 effet magique animé pour les extrémités de l'ascenseur
- 1 effet pour l'impact de la patte d'ours avec le tronc de l'arbre

Images 2D

- 1 image 2D de format panoramique représentant une vue d'ensemble du jeu pour l'écran « crédits »
- 1 image 2D formatée pour la page titre des TP

Sons et musiques

- 1 musique pour le menu principale
- 1 musique pour les niveaux

- 1 son pour une perte de vie
- 1 son pour début de niveau
- 1 son pour un niveau gagné
- 1 son pour une partie perdue
- 1 son pour les mouvements de l'ascenseur
- 1 son pour les roulements du fruit
- 1 son pour l'écureuil
- 1 son pour le lapin
- 1 son pour l'abeille
- 1 son pour le vers
- 1 son pour la patte d'ours
- 1 son pour « mouse over on UI »
- 1 son pour « click on UI »

UI

- 1 ensemble d'éléments d'interface utilisateur du Unity Asset Store

Arrière-plan 3D

- 1 scène 3D du Unity Asset Store pour l'arrière-plan des niveaux et du menu principal

Plateforme technologique

Le tableau suivant résume les différents besoins technologiques pour le projet ainsi que les outils choisis pour les combler.

Besoin	Outil
Documentation	<ul style="list-style-type: none">● Google Docs
Diaporama	<ul style="list-style-type: none">● Google Slides
Prototypage*	<ul style="list-style-type: none">● Unity 2021.3.9 F1● Visual Studio 2022 (code en C#)● Autodesk Maya● Substance 3D Painter● Logic Pro● Contenu téléchargeable libre de droits
Modélisation, textures et animation 3D	<ul style="list-style-type: none">● Autodesk Maya● Substance 3D Painter● Contenu téléchargeable libre de droits
Art 2D (sauf textures)	<ul style="list-style-type: none">● Adobe Photoshop
Éditeur de son et musique	<ul style="list-style-type: none">● Logic Pro● Contenu téléchargeable libre de droits
Moteur de jeu	<ul style="list-style-type: none">● Unity 2021.3.9 F1
Environnement de développement	<ul style="list-style-type: none">● Visual Studio 2022 (code en C#)
Système de gestion de version	<ul style="list-style-type: none">● GitHub Desktop
Gestion de projet	<ul style="list-style-type: none">● Discord● Google Sheets

* Les outils utilisés pour créer les prototypes seront les mêmes que pour le développement du jeu.

Architecture logicielle

Le jeu est conçu selon une approche orientée objet. Chaque élément (obstacles, ascenseur, fruit, etc.) est un objet responsable de ses propres fonctionnalités et états. La gestion des relations entre les objets se fait par des contrôleurs. Ceux-ci utilisent le système d'événement de Unity pour faire correspondre les diverses interactions possibles avec les objets appropriés. C'est également un contrôleur de ce type qui gère les commandes du joueur.

Afin d'optimiser l'utilisation des ressources, le système de « Prefabs » de Unity est utilisé. Celui-ci se base sur des principes inspirés du patron de conception « Fabrique » pour limiter le nombre de chargements des ressources en mémoire.

Le projet fait aussi bon usage des divers outils offerts par Unity afin de diminuer le temps de développement et d'améliorer la qualité de l'expérience du joueur. Par exemple, le roulement du fruit sur l'ascenseur ainsi que la détection des collisions sont gérés par le système de physique de Unity. Afin de simplifier l'implémentation de la physique, l'équipe a également opté pour le « renderer » 2D de Unity.

Présentation de l'équipe

Nicolas Beaulieu

Étudiant de première année au baccalauréat en art et science de l'animation à l'université Laval, Nicolas sera un collaborateur externe au projet. Il agira à titre d'artiste 2D et utilisera ses compétences en dessin afin de créer plusieurs éléments visuels de l'interface utilisateur.

Frédéric Bélanger

Dans le cadre d'un changement de carrière, il a fait un retour aux études en 2018. Il a complété un DEC technique intensif en informatique au cégep du Vieux Montréal, et termine présentement un baccalauréat en informatique à l'université Laval. Il a effectué un premier stage comme programmeur 3D chez Ubisoft, où il occupera un poste à temps plein dès l'été 2023. Ses deux objectifs à moyen et long terme sont de travailler dans l'industrie du jeu vidéo et d'enseigner.

Frédéric s'est inscrit au cours IFT-3113 — Projet de jeu vidéo afin d'ajouter un projet supplémentaire à son portfolio. Il a eu l'idée de départ pour la mécanique de base du jeu et est l'instigateur du projet. Ayant des expériences professionnelles en gestion et une bonne connaissance des méthodes de développement agile, il entend soutenir l'équipe à titre de directeur et réalisateur. Il utilisera également sa bonne maîtrise des mathématiques pour assister ses coéquipiers dans la conception et la programmation du jeu.

Henri Bernard-St-Laurent

Henri Bernard est un étudiant en informatique qui termine actuellement son baccalauréat à l'Université Laval. Il a opté pour la concentration en jeu vidéo et a complété le cours IFT-2103 à la dernière session. Cela lui donne une bonne connaissance de base d'Unity et de plusieurs aspects de la conception de jeu vidéo.

Le cours IFT-3113 lui permettra d'embellir son portfolio et d'améliorer ses compétences en programmation de jeux vidéo. Étant le membre le plus expérimenté de l'équipe avec Unity, il sera le directeur technique du projet. Il participera également à la conception et à la programmation du jeu.

Alexis Normand-Bélanger

Alexis est passé par les sciences de la nature et les sciences humaines au cégep de la Gaspésie et des îles en 2015. Il a poursuivi son parcours à l'institut d'agriculture biologique au cégep de Victoriaville en 2018. Finalement, il est passé à l'université Laval pour débiter un baccalauréat en Art et Sciences de l'animation en 2021.

Il s'est inscrit au cours IFT-3113 afin d'approfondir son expérience en modélisation et animation 3D. Bien qu'à sa deuxième année dans le BASA, c'est une première occasion pour lui de

pouvoir exercer son art dans un contexte de création de jeu vidéo et ainsi peaufiner son portfolio. Il sera le directeur artistique du projet et produira plusieurs éléments visuels à titre d'artiste 3D.

Mathis Rhéaume

Mathis a complété son DEC en cinéma au cégep de Sainte-Foy pour ensuite poursuivre en Art et sciences de l'animation à l'Université Laval. Il se spécialise en ce moment comme artiste 3D.

Le cours IFT-3113 — Projet de jeu vidéo représente une première expérience en jeu vidéo et lui permettra d'en apprendre davantage sur les processus de création liés au domaine. Il espère entre autres observer comment le « workflow » diffère de celui d'un film.

Ses trois objectifs principaux dans le cadre du cours sont de développer ses connaissances en 3D, s'initier au jeu vidéo et agrémenter son portfolio. Il utilisera ses aptitudes en modélisation et animation pour occuper la fonction d'artiste 3D.

François Veilleux-Lessard

À 28 ans, François a fait un retour aux études à la faculté de musique de l'université Laval où il a complété un certificat en réalisation audionumérique. Il a ensuite entamé un baccalauréat en informatique à l'hiver 2022 et souhaite approfondir ses connaissances pour pouvoir travailler dans le domaine du jeu vidéo.

Avec son expérience en musique et en traitement audio dans Unity, il occupera le rôle de directeur audio et compositeur pour le projet. Ses tâches principales seront de composer la musique originale et de mettre en place tous les sons du jeu.

Dans l'éventualité où cette tâche serait achevée avant la fin du projet, François pourra offrir son support aux programmeurs et concepteurs de jeu pour implémenter de nouvelles fonctionnalités.

Annexe 1 — analyse technologique

Modèle 3D avec textures

- Pour fin de correction, ce prototype est celui de Mathis Rhéaume.
- Après quelques recherches, il s'est avéré que le format de fichier « fbx » semble être idéal pour les jeux vidéo. Le prototype a confirmé que le chargement des modèles texturés enregistrés sous ce format se fait correctement dans Unity.
- Pour une meilleure utilisation des ressources matérielles, les modèles sont chargés une fois en mémoire et clonés au besoin. Ce modèle suit les principes du patron de conception « fabrique » et utilise le système de « Prefabs » de Unity.
- L'équipe a décidé d'utiliser les cartes de textures albédo, normale et rugosité pour le projet. Vu le style artistique minimaliste, la plupart des modèles produits ne feront pas un grand usage des cartes de normale et de rugosité. C'est le cas de ceux produits pour le prototype. Le modèle d'écureuil sera plus gros et plus complexe. Il bénéficiera sans doute davantage de ces cartes.
- Unity n'emploie pas directement les cartes de rugosité, mais plutôt une combinaison de métallicité et douceur (« metal smoothness »). La douceur étant l'inverse de la rugosité, il a été nécessaire d'apprendre comment inverser cette dernière et la combiner à une carte de métallicité nulle dans Substance 3D. Quelques essais ont été requis afin de trouver la bonne configuration dans Substance 3D.
- Vu la difficulté mentionnée précédemment, un second modèle a été implémenté plus rapidement que prévu afin de mieux évaluer le temps de conception de ceux-ci. Un modèle de petit fruit rouge a donc aussi été incorporé au prototype.
- Le style visuel « low poly » plaît bien aux membres de l'équipe et correspond à l'effet initialement désiré. Le style visuel est donc bien figé, et les deux modèles produits serviront de référence pour l'ensemble du projet.

Animation 3D

- Pour fin de correction, ce prototype est celui de Alexis Normand-Bélanger et Frédéric Bélanger.
- Après quelques recherches, il s'est avéré que le format de fichier « fbx » semble idéal pour les jeux vidéo. Le prototype a confirmé que le chargement des modèles animés enregistrés sous ce format se fait correctement dans Unity.
- Trois animations ont été ajoutées au modèle d'abeille. La première fait battre l'aile gauche, la seconde l'aile droite et la troisième combine les deux ailes.
- Les trois animations ont été enregistrées en une longue séquence dans un seul fichier fbx. Une fois chargées dans Unity, elles ont été découpées en 3 clips individuels. Pour le développement du projet, l'équipe a décidé que chaque animation serait enregistrée dans un fichier individuel. Procéder avec un seul fichier a le désavantage de nécessiter un redécoupage des clips à chaque fois qu'une animation est modifiée.
- Dans Unity, une composante « Animator » à laquelle on a assigné un « Animator Controller » a été ajoutée au modèle d'abeille. Les 3 clips précédemment découpés ont été transférés dans ce contrôleur et liés par des transitions contrôlables à l'aide de variables booléennes. Ces variables sont modifiées par trois boutons radio demandant respectivement l'activation de l'aile gauche, de l'aile droite ou des deux simultanément.
- Toujours dans Unity, un script a été assigné au modèle d'abeille. C'est dans celui-ci que le lien entre les boutons radio et les variables booléennes contrôlant les animations est effectué.
- Une glissière a été ajoutée pour faire varier la vitesse de déroulement des animations afin de mieux évaluer quelle fréquence de battement d'ailes donnerait un visuel plus adéquat.
- Toujours pour améliorer l'apparence visuelle du résultat, des petites variations aléatoires de la position du modèle sur l'axe Y ont été ajoutées dans le script de l'abeille.
- Les deux derniers ajouts ont permis à l'équipe d'apprendre comment peaufiner l'apparence d'une animation sans avoir à faire de multiples allers-retours entre Blender et Unity.
- L'équipe est satisfaite de l'apparence du résultat final. Le résultat composé de l'ensemble du modèle, des textures et des animations pour l'abeille servira de référence pour l'ensemble du projet.

Ascenseur et fruit

- Pour fin de correction, ce prototype est celui de Henri Bernard-St-Laurent.
- Dans ce prototype, un prisme rectangulaire positionné horizontalement est utilisé pour représenter l'ascenseur, et une sphère pour le petit fruit rouge.
- Au début de l'exécution, la sphère est déposée sur le prisme. Un bouton a été ajouté pour ramener la sphère et le prisme à leurs positions originales. On simule ainsi la réinitialisation d'un niveau à la suite d'une perte de vie du joueur.
- Deux sphères invisibles ont été jointes aux extrémités de l'ascenseur à l'aide de « Wheel Joint 2D ». Les positions verticales de celles-ci sont modifiées soit à l'aide d'un « gamepad », soit par les touches clavier « w-s » et « i-k ». C'est ainsi que l'on contrôle la position et l'inclinaison de l'ascenseur.
- La sphère et l'ascenseur ont tous les deux des corps rigides. Les déplacements de la sphère sur ce dernier sont entièrement gérés par le système physique d'Unity. Des prismes rectangulaires invisibles avec des corps rigides ont été ajoutés à la scène pour délimiter les mouvements de ces deux composantes. Par exemple, deux de ces prismes sont positionnés à droite et à gauche de l'ascenseur pour empêcher la sphère de tomber lorsqu'elle arrive aux extrémités.
- L'équipe a décidé d'opter pour le « renderer » 2D de Unity afin de simplifier l'implémentation de la physique
- Pour trouver les réglages idéaux de la vitesse de l'ascenseur et de la physique appliquée à la sphère, plusieurs glissières ont été ajoutées à l'interface du prototype.
- Il est à noter que dans le prototype, il est possible de donner une vitesse trop élevée à l'ascenseur. Lors de mouvements verticaux brusques, la sphère peut alors rester suspendue dans les airs un bref instant avant de retomber au bas de l'écran. Si cela se produit, on peut utiliser le bouton qui ramène ces deux éléments à leurs positions initiales. L'équipe a fait de nombreux essais pour s'assurer que ce problème ne se produit pas à la vitesse choisie pour le jeu.

Tests sonores

- Pour fin de correction, ce prototype est celui de François Veilleux-Lessard.
- Le « audio listener » est placé sur un objet (fruit rouge) que l'on peut déplacer avec les commandes haut, bas, gauche et droite du clavier.
- Il est possible de naviguer et de s'approcher de 4 sources sonores différentes : un son d'abeille, un ensemble d'exemples de son des personnages du jeu, et 2 thèmes musicaux.
- Un mélangeur audio est instancié et permet de diviser les sons en 2 canaux : SFX et musique.
- On peut ajuster à l'écran le volume de chaque canal et le volume du master.
- Les objets qui ont un son SFX associé sont ajustés pour que la proximité du fruit ait un impact sur le volume du son (plus on s'approche, plus le son est intense).
- Les objets ayant un extrait musical associé sont simplement placés de façon à ce qu'on puisse changer de thème en déplaçant le fruit à droite ou à gauche de l'écran.
- L'utilisation du mélangeur audio est un succès et les extraits élaborés confirment l'esthétique sonore du jeu.

Annexe 2 — rapport d'avancement 1

Tâche	Réalisation	Date prévue	Date livrée	Commentaires
Création d'un modèle 3D d'abeille avec textures	Mathis	07/02	21/02	<p>La texture de rugosité doit être transformée en « metal smoothness » pour Unity (07/02)</p> <p>Le modèle est mal orienté et trop réfléchitif dans Unity (14/02)</p> <p>Nous avons des difficultés à regrouper les textures dans Substance. L'enseignant a dit que nous ne serons pas pénalisés. (21/02)</p>
Création des sections « Boucle de jeu » et « Mécaniques prévues » du document de design	Frédéric	14/02	17/02	Date déplacée au 17/02 (07/02)
Ajout de 2 animations au modèle d'abeille	Alexis	10/02	10/02	Date déplacée au 10/02 (07/02)
Composition de 2 musiques et recherche de sons pour le prototype « Tests sonores »	François	14/02	17/02	Les musiques sont complétées, il reste les sons (14/02)
Création du prototype « Ascenseur et fruit »	Henri	14/02	17/02	Il reste quelques petits bogues à régler (14/02)
Rédaction de la section « Ressources de jeu » du document de design	Frédéric	14/02	17/02	Date déplacée au 17/02 (07/02)
Création d'une image 2D résumant le jeu pour la page titre du document de design et la section « crédits » du jeu	Nicolas	14/02	21/02	Première version montrée, il reste des améliorations à faire (14/02)

Tâche	Réalisation	Date prévue	Date livrée	Commentaires
Création du prototype « Tests sonores »	François	21/02	22/02	Petite correction à faire sur l'audio positionnelle dans le prototype (21/02)
Création du prototype « Génération procédurale »	Henri	21/02	17/02	Il reste quelques petits bogues à régler (14/02)
Création du prototype « Modèle 3D avec textures »	Frédéric	21/02	21/02	<p>Réalisation changée pour Frédéric (14/02)</p> <p>Le « metal smoothness » doit être appliqué manuellement sur les différents matériaux dans Unity (17/02)</p> <p>Ajout des « prefabs » de Unity pour maximiser l'utilisation des ressources (17/02)</p> <p>Les modèles pour le petit fruit rouge et le tronc d'arbre ont également été ajoutés au prototype (21/02)</p>
Création du prototype « Animation 3D »	Frédéric	21/02	17/02	
Créations des exécutables pour les prototypes « Modèle 3D avec textures », « Ascenseur et fruit » et « Génération procédurale »	Frederic	22/02	23/02	<p>Tous les prototypes ont été combinés en un seul exécutable (14/02)</p> <p>Responsable changé pour Frederic (21/02)</p>
Création de l'exécutable pour le prototype « Animations 3D »	Frédéric	22/02	23/02	Tous les prototypes ont été combinés en un seul exécutable (14/02)
Création de l'exécutable pour le prototype « Tests sonores »	Frederic	22/02	23/02	Tous les prototypes ont été combinés en un seul exécutable (14/02)

Tâche	Réalisation	Date prévue	Date livrée	Commentaires
Rédaction de l'annexe 2 « Rapport d'avancement 1 » pour le TP3	Frédéric	22/02	17/02	
Création de l'annexe 3 « Galerie d'art de production » pour le TP3	Alexis	22/02		
Création de la page titre du document de design pour le TP3	François	22/02	21/02	Réalisation changée pour François (07/02)
Création du diaporama des mécaniques de jeu	Alexis et Matis	22/02	22/02	Réalisation changée pour Frédéric et Henri (17/02)
Rédaction du document de design pour le TP3	Frédéric	22/02	17/02	
Assemblage final des documents et des fichiers requis pour le TP3	Frédéric	23/02		
Remise du TP3	Frédéric	24/02		
Création de modèles 3D avec textures pour le tronc d'arbre, le petit fruit rouge et la branche morte (l'ascenseur) avec textures	Alexis	25/02		Fruit rouge complet (21/02) Branche complet (07/02) Il faut enlever une fente sur le tronc d'arbre pour améliorer l'aspect visuel (21/02)
Création d'un effet visuel lumineux « magique » pour les extrémités de l'ascenseur	Frédéric	28/02		Réalisation changée pour Frédéric (07/02)
Création des animations pour le modèle 3D d'abeille	Alexis	28/02	21/02	Les abeilles auront seulement des battements d'ailes et des oscillations. Donc, pas de mouvement des pattes ou des yeux (21/02)

Tâche	Réalisation	Date prévue	Date livrée	Commentaires
Création d'un modèle 3D de vers sortant d'un trou avec animations et textures	Mathis	07/03		
Création d'un modèle 3D de patte d'ours avec textures et animations	Mathis	07/03		Première version montrée avec quelques changements suggérés (21/02)
Recherche d'un décor d'arrière-plan 3D dans l'Asset Store de Unity (pour les niveaux du jeu et le menu principal)	Henri	07/03		Réalisation changée pour Henri (07/02)
Création d'un niveau de base dans Unity avec l'ascenseur, le petit fruit, le tronc d'arbre ainsi que l'ajout aléatoire des trous et des abeilles	Henri	07/03		
Création des sons pour les mouvements verticaux de l'ascenseur et le roulement du fruit	François	07/03		
Création des icônes 2D pour l'interface utilisateur	Nicolas	07/03		
Ajout de l'obstacle « vers gluant » au jeu	Frédéric	12/03		
Ajout de l'obstacle « patte d'ours » au jeu	Henri	12/03		
Création des sons pour l'abeille, le vers et la patte d'ours	François	14/03		
Création du menu principal du jeu	Henri	14/03		
Création de l'interface utilisateur	Henri	14/03		

Tâche	Réalisation	Date prévue	Date livrée	Commentaires
Création d'un modèle 3D d'écureuil avec textures et gréement (« rigging »)	Alexis	14/03		
Création d'une animation « impatience » pour l'écureuil	Alexis	17/03		
Création d'une animation « mort dramatique » pour l'écureuil	Alexis	17/03		
Création d'une animation « en amour avec son fruit » pour l'écureuil	Mathis	17/03		
Création d'une animation « peur d'un danger » pour l'écureuil	Mathis	17/03		
Ajout de l'arrière-plan 3D au niveau	Henri	17/03		
Création des musiques pour le menu principal et les niveaux du jeu	François	17/03		
Création des sons liés à l'interface utilisateur	François	17/03		
Création des écrans « niveau gagné », « nouveau niveau » et « perte de vie »	Henri	17/03		
Création de l'écran « partie perdue »	Frédéric	17/03		
Création des sons pour « niveau gagné », « nouveau niveau », « perte de vie » et « partie perdue »	François	21/03		

Tâche	Réalisation	Date prévue	Date livrée	Commentaires
Mise à jour du document de design pour le TP4	Frédéric	21/03		
Rédaction des annexes 1 et 2 « Rapport d'avancement 2 » et « Rapport d'avancement 3 » pour le TP4	Frédéric	21/03		
Mise à jour du diaporama des mécaniques de jeu pour le TP4	Mathis	21/03		
Mise à jours de l'annexe 3 « Galerie d'art de production » pour le TP4	Alexis	21/03		
Assemblage final des documents et des fichiers requis pour le TP4	Frédéric	23/03		
Remise du TP4	Frédéric	24/03		
Ajout de l'écureuil animé qui réagit aux actions du joueur dans Unity	Henri	28/03		
Création d'un modèle 3D de lapin magicien avec texture et gréement	Alexis	28/03		
Création d'un modèle 3D avec textures pour des « vies bonus »	Alexis	28/03		
Création d'une animation « magicien qui fait de la télépathie » pour le lapin	Mathis	28/03		

Tâche	Réalisation	Date prévue	Date livrée	Commentaires
Création des sons pour l'écureuil animé	François	28/03		
Création de l'exécutable pour le TP5	Henri	02/04		
Préparation de la présentation du projet pour le TP5	Équipe	02/04		
Présentation du projet pour le TP5	Équipe	04/04		
Discussion sur l'évaluation des autres projets pour le TP5	Équipe	04/04		
Rédaction de l'évaluation des autres projets pour le TP5	Frédéric	05/04		
Assemblage final des documents et des fichiers requis pour le TP5	Frédéric	05/04		
Remise du TP5	Frédéric	07/04		
Ajout du lapin magicien animé dans Unity	Henri	14/04		
Ajout des niveaux multiples avec incrémentation de la difficulté	Henri	14/04		
Ajout de « vies bonus » pouvant être amassées par le joueur	Frédéric	14/04		
Modifications selon les commentaires reçus lors de la présentation du projet	Équipe	21/04		
Tests, corrections de bogues et polissage	Équipe	24/04		

Tâche	Réalisation	Date prévue	Date livrée	Commentaires
Mise à jour du document de design pour le TP6	Frédéric	24/04		
Rédaction de l'annexe 1 « Rapport d'avancement 4 » pour le TP6	Frédéric	24/04		
Rédaction de l'annexe 2 « Post-mortem » pour le TP6	Frédéric	24/04		
Rédaction de l'annexe 4 « Réponses aux commentaires » pour le TP6	Frédéric	24/04		
Mise à jour du diaporama des mécaniques de jeu pour le TP6	Mathis	24/04		
Mise à jour de l'annexe 3 « Galerie d'art de production » pour le TP6	Alexis	24/04		
Création de l'exécutable pour le TP6	Henri	24/04		
Assemblage final des documents et des fichiers requis pour le TP6	Frédéric	27/04		
Remise du TP6	Frédéric	28/04		

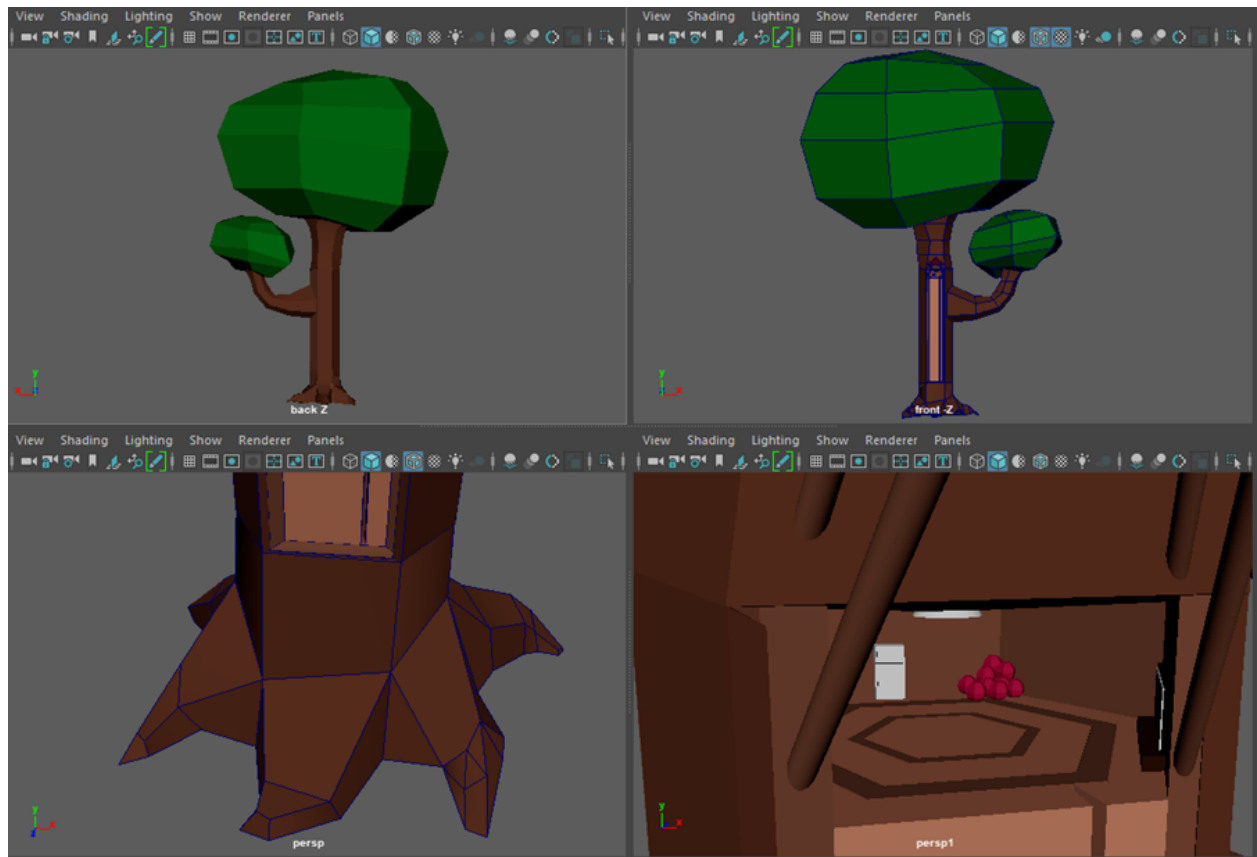
Annexe 3 — galerie de production

Arbre de l'écureuil

L'arbre est l'élément central des différents niveaux du jeu. Il se retrouve également en arrière-plan du menu principal ainsi que sur d'autres cinématiques.

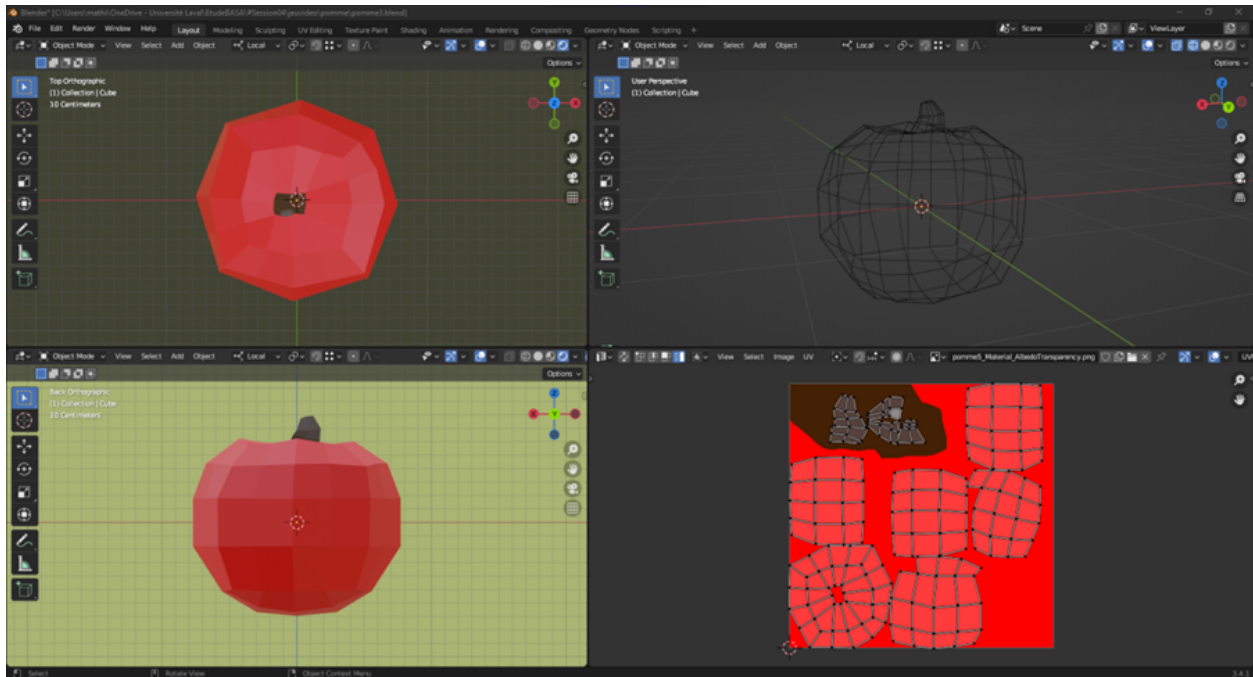
Ce modèle est composé de 2 sections simples. La partie principale est constituée du tronc de l'arbre et contient l'espace où se déroule le jeu. Sa longueur doit donc être proportionnelle à celle des niveaux.

La partie restant est utilisée à des fins esthétiques. Par exemple, au bout du tronc se trouve le repère de l'écureuil. Des éléments de décorations typiques d'une maison humaine ont été ajoutés pour donner de la personnalité à l'espace et augmenter l'aspect ludique du personnage.



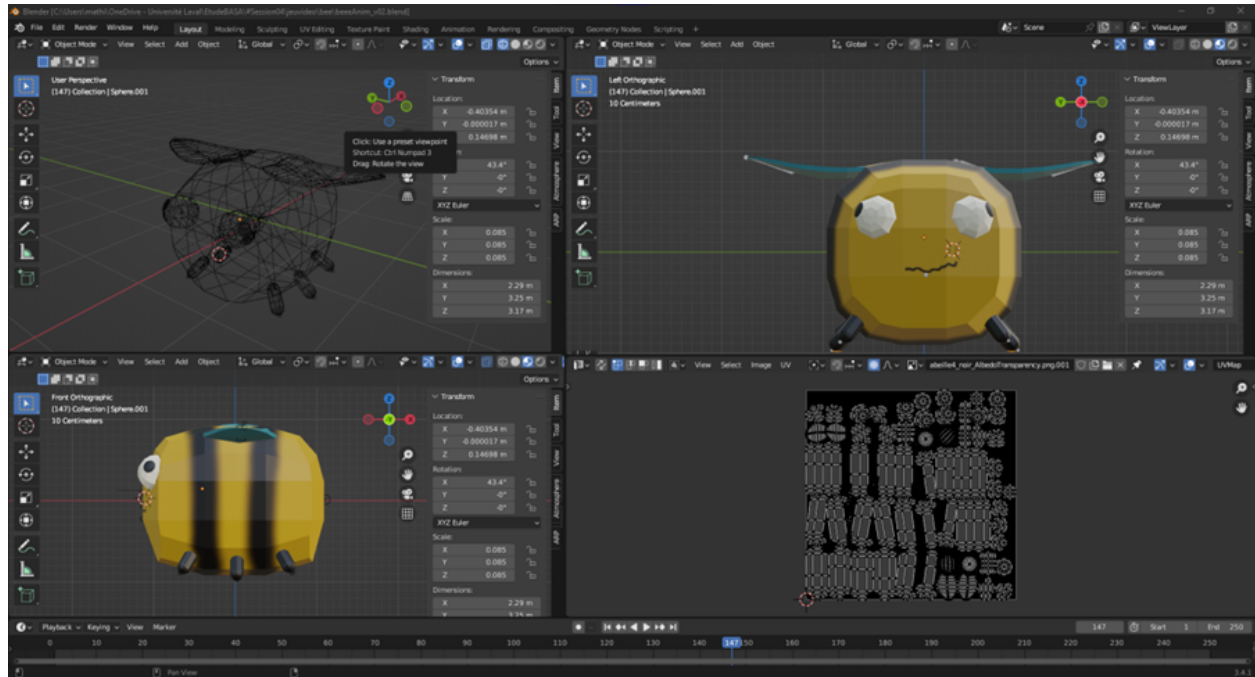
Petit fruit rouge

Le petit fruit rouge est l'objet qui doit être remonté au haut de l'arbre par le biais de l'ascenseur. Le modèle sera également recoloré et utilisé pour représenter les vies supplémentaires que le joueur peut ramasser le long d'un niveau.



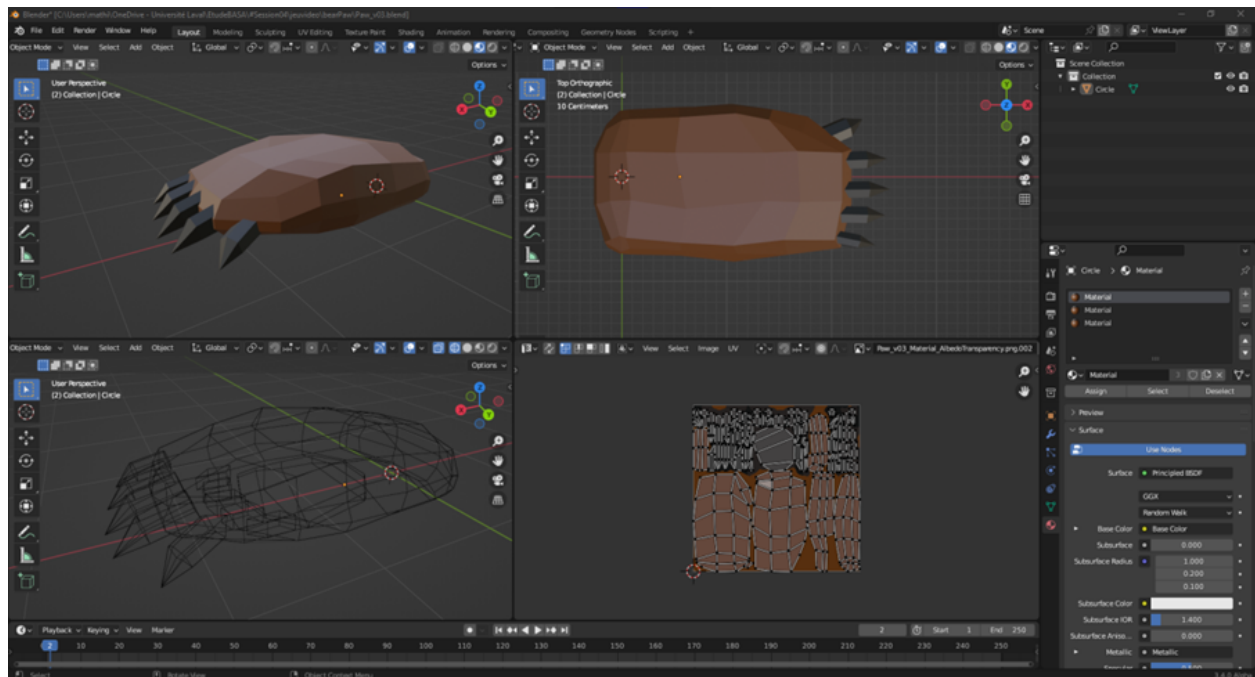
Abeille

Un des principaux obstacles sur l'arbre. Celles-ci feront des allers-retours entre les trous dans l'écorce afin de faire tomber le petit fruit et bloquer le joueur dans sa progression.



Patte d'ours

L'ours est l'un des obstacles les plus dangereux. Rapide et impitoyable, il peut apparaître à tout moment pour écraser le fruit transporté d'un seul coup de patte. Une ombre sur le tronc apparaît peu avant l'impact, laissant du temps au joueur pour réagir.



Menu principal (logo, UI et arrière-plan)

Plusieurs versions du menu principal ont été produites afin d'évaluer différents styles visuels. C'est la première des images ci-dessous qui présente la version retenue.

D'autres optimisations sont à prévoir afin que ce premier contact visuel avec le jeu soit le plus agréable et pratique possible. Par exemple, l'écureuil sera dans son repère et donnera des commentaires au joueur pour l'aider dans sa navigation du menu.





Annexe 4 — prototype supplémentaire

Génération procédurale

- Dans ce prototype, les trous dans le tronc de l'arbre sont représentés par des cylindres noirs longeant l'axe Z. Ils apparaissent comme des cercles noirs à l'écran. Les abeilles prennent quant à elles la forme de sphère jaune.
- Les générations des trous et des abeilles sont contrôlées à l'aide de deux scripts. Dans ceux-ci, on s'assure qu'il y a une distance minimale entre les obstacles pour ne pas créer de niveau impossible à franchir.
- Il est possible de sélectionner le nombre d'abeilles et de trous générés à l'aide de glissières. Pour chacun de ces éléments, on peut lancer ou annuler la génération à l'aide de deux boutons.
- Lorsqu'un grand nombre d'obstacles est demandé, il peut être difficile de trouver aléatoirement des positions convenables pour tous les éléments. Pour ne pas créer de boucle sans fin où le contrôleur cherche en permanence une position libre pour un obstacle, le nombre d'essais permis est limité. Il peut donc y avoir des situations où le nombre d'obstacles est très légèrement moins élevé que ce qui avait été demandé. Le joueur ne sera pas conscient de ceci.
- L'équipe est très satisfaite du résultat obtenu. Des centaines d'essais ont démontré que la génération est solide. La répartition des obstacles est uniforme et il y a toujours un chemin possible pour la sphère. Il y aura donc une génération procédurale des niveaux dans le jeu.
- Note concernant les chiffres blancs : si l'équipe a le temps, elle envisage d'ajouter un mode « legacy » au jeu. Celui-ci serait inspiré du cabinet d'arcade mécanique du jeu Ice Cold Beer duquel est inspiré le projet.

Dans ce mode, l'objectif ne serait plus d'atteindre le haut d'un arbre, mais de faire entrer le fruit dans un trou identifié comme cible. Lorsque le joueur progresse d'un niveau à l'autre, cette cible est de plus en plus haute. Il doit donc éviter un plus grand nombre d'obstacles pour l'atteindre.

Dans le prototype, les chiffres blancs au-dessus de certains trous représentent une progression potentielle pour ces cibles. Celles-ci sont aussi générées procéduralement.

Annexe 5 — lien vers le code source

https://github.com/hebernard/projet-jeu-video/releases/tag/TP3_prototypes