



UNIVERSITÉ
LAVAL

ANI-2012 Programmation en animation

Document de design

préparé par Équipe 2 :

Mathieu Dubreuil

Virginie Juteau

Charlotte Sarrazin

Tatyana Shevchenko

François Veillette

Présenté à Philippe Voyer

15 décembre 2019

Table des matières

1. Introduction	3
2. Sommaire	4
3. Présentation de l'équipe	5
4. Interactivité	6
5. Fonctionnalités	6
5.1 Scène Maya	6
5.2 Attribut	7
5.3 Sélection	7
5.4 Poses clés	7
5.5 Génération procédurale	8
5.7 Intégration (Sketchfab et Unity Asset Store)	9
5.7.1 Flappy Bird (3dBird de Sketchfab)	9
5.7.2 Maison (Baker_house du Unity Asset Store)	9
5.7.3 Momie (Mummy_Mon du Unity Asset Store)	9
5.7.4 Chat (Cartoon_Cat du Unity Asset Store)	9
5.7.5 Plancher (3dBird de Sketchfab)	10
5.7.6 Arbre (Fir_Tree du Unity Asset Store)	10
5.7.7 Nuage (3dBird de Sketchfab)	10
5.7.7 Cloture (3dBird de Sketchfab)	10
5.7.7 Tube (3dBird de Sketchfab)	10
5.7.8 Roche (3dBird de Sketchfab)	10
5.8 Animation avec une ligne de temps	10
5.9 Animation avec une machine à états	10
5.10 Effets visuels	11
5.10.1 Effets visuels en boucle	11
5.10.2 Effet visuel par le clic	11
5.10.3 Effet de particules	12
6. Ressources	13
6.1 Ressources originales	13
6.2 Ressources externes	13

1. Introduction

Le document de design présente une vue d'ensemble du travail pratique 2 du cours de programmation en animation. Le but du travail pratique 2 est de développer une animation interactive contenant des éléments visuels en trois dimensions. Les notions théoriques utilisées sont de la programmation avec deux technologies complémentaires, soit Maya pour la production de contenu visuel et animé, ainsi que Unity qui permet d'animer ce contenu, de le rendre interactif et de le publier vers différentes plateformes.

Le travail pratique 2 est inspiré du jeu populaire de *Flappy Bird*. Le jeu est une application mobile publiée en 2013 par le développeur et graphiste Vietnamien Dong Nguyen. Le jeu a eu un succès exponentiel dans ses débuts grâce à son effet addictif. Par contre, le jeu *Flappy Bird* a été retiré des différentes boutiques d'applications mobiles par son créateur Dong Nguyen, qui avait un sentiment de culpabilité avec la popularité de son jeu un peu trop addictif. Le jeu est donc populaire grâce à son succès, mais aussi grâce à son petit « scandale » lors de son retrait. Le travail présentera une adaptation d'un jeu similaire à celui-ci.

Le document de design est séparé en cinq grandes sous-sections pour survoler tous les aspects du jeu : le sommaire, la présentation de l'équipe, l'interactivité de l'application, les fonctionnalités et les ressources.

***** Afin que le jeu fonctionne correctement, les «packages » suivants doivent être installés :**

- com.unity.2d.sprite: Package [com.unity.2d.sprite@1.0.0]
- com.unity.2d.tilemap: Package [com.unity.2d.tilemap@1.0.0]
- com.unity.modules.androidjni:Package[com.unity.modules.androidjni@1.0.0]
- com.unity.package-manager-ui:Package[com.unity.package-manager-ui@2.2.0]
- com.unity.timeline: Package [com.unity.timeline@1.1.0]
- com.unity.ugui: Package [com.unity.ugui@1.0.0]

2. Sommaire

Après un brainstorm entre les membres de l'équipe, nous avons décidé de créer un jeu de *Flappy Bird*. Le but ultime du jeu est de voyager à travers le niveau sans entrer en collision avec le sol ou un tuyau. Une collision avec un de ses éléments provoque la fin de la partie et un retour à la case de départ. Le gameplay repose sur l'agilité du joueur à faire avancer un oiseau avec le clavier dans un environnement à défilement horizontal sans heurter les obstacles présents dans la scène. Les tuyaux peuvent autant provenir du haut ou du bas de l'écran. Afin de rendre le tout visuellement plus alléchant, nous avons décidé d'ajouter des éléments 3D à la scène tel qu'un chat, des sapins, des nuages, une momie, des clôtures et des maisons. Certains des éléments présents dans la scène Unity ont été créés par notre équipe et d'autres sont des ressources externes (section 6). Le processus de création sera dévoilé en détail tout au long des prochaines sections.

3. Présentation de l'équipe

La table 1 ci-dessous présente les membres de l'équipe. Elle contient une brève présentation, une liste de l'expertise et un sommaire des tâches accomplies et des responsabilités pour chaque membre de l'équipe.

Table 1: Présentation des membres de l'équipe

Nom	Brève Présentation	Expertise	Responsabilités
Mathieu Dubreuil	Développeur-Analyste chez Revenu Québec DEC en informatique et gestion Baccalauréat en informatique(En cours)	- Développement Front-end (Angular) - Développement Back-end (.net) - Développement Java - Méthode agile (XP, scrum)	-Modélisation de la Scène Unity (5.6) - Intégration des divers asset dans la Scène (5.7) - Génération d'une animation avec une ligne de temps (5.8) - Génération d'une animation avec une machine à états (5.9)
Virginie Juteau	Certificat en art et science de l'animation en cours Baccalauréat en sciences biologiques DEC en sciences de la nature Illustratrice et concept artist	-Adobe Photoshop, Clip studio paint -Dessin et design -Modélisation 3D et animation (3dsmax et Blender), animation dans Maya (brièvement) -Programmation en R	- Modélisation de la scène Maya (5.1) - Génération des poses clés (5.4) - Génération procédurale (5.5) - Aide dans génération de l'attribut (5.2) - Aide dans génération de la sélection (5.3) - Rédaction du document de design
Charlotte Sarrazin	Développeur Front-end dans une entreprise de solutions réseau local DEC en science informatique et mathématique Baccalauréat en génie logiciel	- Développement Front-end (Angular) - Développement d'application mobile (Objective-C, Swift) - Développement Java - Méthode agile (XP, scrum)	- Rédaction du document de design - Ajout de menu dans le jeu - Système de particule - Effets visuels (5.10)
Tatyana Shevchenko	Baccalauréat en design graphique (en cours) Travaille comme assistante au coordonnateur des communications chez CSPQ	- Design graphique (Suite Adobe : Photoshop, Illustrator, InDesign, XD) - Animation Motion Design - Méthode agile (XP, scrum)	- Génération de l'attribut (5.2) - Génération de la sélection (5.3) - Mise en page initial et la rédaction du document de design
François Veillette	Travaille comme technicien en audiovisuel et Baccalauréat multi en audiovisuel (en cours)	-Tournage et montage vidéo -Animation Motion Design -Création audio-numérique	- Rédaction du document de design - Recherche d'assets en ligne

4. Interactivité

La table 2 présente un survol des effets qui sont enclenchés par l'utilisateur à l'aide des touches du clavier.

Jeu de *Flappy Bird*

Table 2: Les formes d'interactivité possibles dans le jeu de <i>Flappy Bird</i>	
Touche/Souris	Effets
Touche : <i>Espace</i> enfoncée (Key Press)	L'oiseau s'oriente vers le haut en gagnant graduellement de l'altitude
Touche : <i>Espace</i> relâchée (Key Down)	L'oiseau s'orient vers le bas en perdant graduellement de l'altitude
Clic gauche de la souris	La sélection est activée dans le menu.

5. Fonctionnalités

La section de fonctionnalités contient les critères fonctionnels du travail pratique. Les critères sont détaillés et permettent une meilleure compréhension du travail qui a été effectué.

5.1 Scène Maya

Le projet contient un fichier de scène pour le logiciel *Maya* en format binaire (.mb). Le fichier *flappy_bird_scene.mb* contient des lumières, des plans, les tuyaux et l'oiseau *Flappy Bird*. Les ressources de la scène *Maya* vont permettre l'exportation dans un fichier de format .fbx, qui sera utilisé dans le logiciel *Unity*. Ce fichier est la base de la partie développée avec *Maya* pour ce travail pratique. Autrement dit, il est le fichier le plus important pour la manipulation ou la génération de contenu à l'aide des différents scripts en *Python*.

À noter que les assets de cette scène ne seront pas utilisés pour le jeu en tant que tel, mais restent tout de même dans la thématique.

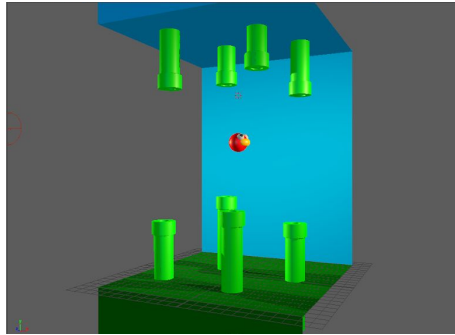


Figure 1: Scène Maya de Flappy Bird.

5.2 Attribut

Le fichier *select_attribute.py* contient un code permettant de modifier les attributs de *Flappy Bird* suite à une sélection. Notamment, le code permet d'effectuer la lecture et ensuite l'écriture d'une valeur d'un attribut assigné à un noeud. Les attributs assignés sont la translation sur les trois axes (x, y et z) ainsi que la rotation sur l'axe vertical (y) et une légère mise à l'échelle.

À noter que pour utiliser *select_attribute.py*, il faut avoir préalablement ouvert *flappy_bird_animation.py* dans une scène Maya.

5.3 Sélection

Le fichier *select_attribute.py* contient un code de sélection. Il permet de sélectionner uniquement l'oiseau *Flappy Bird* ainsi que ses clés d'animation. Si la scène est vide, l'utilisateur sera averti à l'aide d'une impression dans le terminal qu'il n'y a rien dans la scène présente. L'avantage d'un fichier de sélection est qu'il permet d'exporter toute la sélection dans un même fichier .fbx, qui sera utilisé dans *Unity*. À l'aide d'une sélection plus précise, il est possible d'exporter tous les éléments d'un seul objet de la scène, sans sélectionner les autres éléments de la scène. Il est donc facile d'isoler un seul objet d'une scène avec de nombreux éléments.

5.4 Poses clés

Le fichier *flappy_bird_animation.py* est un fichier de génération des poses clés d'animation sur l'oiseau *Flappy Bird* de la scène. Tout d'abord, le script génère tous les éléments nécessaires pour modéliser l'oiseau : les formes géométriques, les différentes couleurs, les shaders et la fusion de l'ensemble des éléments. Par la suite, le script génère une animation qui est assignée à l'oiseau modélisé. Elle est plutôt simple et présente un rebondissement avec déformations, comme une balle. L'animation est générée de la manipulation des attributs *translate*, *scale* et/ou *rotate* sur notre élément. Le résultat est une séquence de poses clés d'animation sur l'élément de *Flappy Bird* de notre scène.

À noter que cette animation ne sera pas utilisée dans le jeu dans *Unity* et que le modèle ne possède pas de rig.

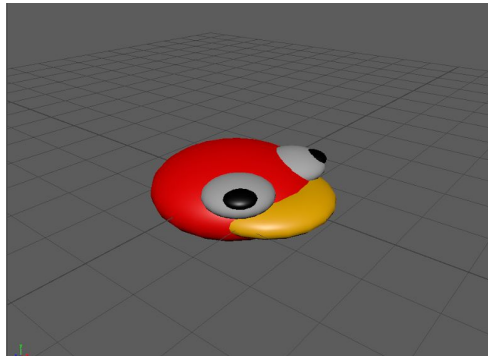


Figure 2: Première pose clé de l'animation Flappy Bird.

5.5 Génération procédurale

Le fichier *pipe_generator.py* fait de la génération procédurale de contenu dans la scène *Maya*. Le script génère dynamiquement plusieurs instances de tuyaux verts qui peuvent être utilisés comme des obstacles dans le jeu. Tout d'abord, les formes, la couleur et le shader *blinn* d'un tuyau sont créés et fusionnés. Par la suite, l'ensemble est groupé et assigné à une position aléatoire qui se situe entre deux bornes pour chaque axe (x, y et z). Cela permet aux tubes de ne pas trop s'éloigner des limites de la surface du sol et de ne pas être trop hauts ni trop bas dans la scène, mais de tout de même conserver une diversité de positions.

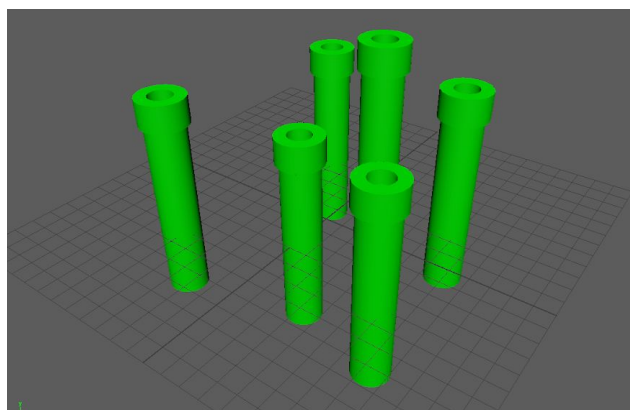


Figure 3: Génération procédurale de tuyau verts.

5.6 Scène Unity

Le projet contient une scène *flappybird.unity* qui contient l'ensemble du jeu. Dans cette scène, plusieurs assets de *Sketchfab* et du *Unity Asset Store* sont utilisés, incluant les modèles, les shaders et les textures, mais on y retrouve aussi des lumières et une caméra. Les assets modélisés sont l'oiseau, un arbre, une clôture, un tuyau, une maison, un sol, une pierre, un chat, un nuage et une momie. Ces éléments sont répétés horizontalement pour que le joueur puisse voir la scène défiler durant la partie. L'oiseau *Flappy Bird* est animé et peut changer de position lorsqu'on appuie et relâche la touche espace afin de permettre au joueur d'éviter une collision avec des obstacles.



Figure 4: Scène Unity en action, incluant les modèles du Asset Store.

5.7 Intégration (*Sketchfab* et *Unity Asset Store*)

5.7.1 Flappy Bird (3dBird de *Sketchfab*)

La scène *Unity* contient un modèle .fbx de *Flappy Bird* riggé et animable. C'est avec ce personnage que le joueur peut effectuer sa partie. Le modèle contient déjà des matériaux et textures.

5.7.2 Maison (Baker_house du *Unity Asset Store*)

La scène *Unity* contient un modèle .fbx de maison qui fait partie du décors de fond. Le modèle contient déjà des matériaux et textures.

5.7.3 Momie (Mummy_Mon du *Unity Asset Store*)

La scène *Unity* contient un modèle .fbx de momie qui fait partie du décors de fond. Le modèle contient déjà des matériaux et textures.

5.7.4 Chat (Cartoon_Cat du *Unity Asset Store*)

La scène *Unity* contient un modèle .fbx de chat qui fait partie du décors de fond. Le modèle contient déjà des matériaux et textures.

5.7.5 Plancher (3dBird de *Sketchfab*)

La scène *Unity* contient un modèle .fbx de plancher qui défile sous *Flappy Bird*. Le modèle contient déjà des matériaux et textures.

5.7.6 Arbre (Fir_Tree du *Unity Asset Store*)

La scène *Unity* contient un modèle .fbx d'arbre qui fait partie du décors de fond. Le modèle contient déjà des matériaux et textures.

5.7.7 Nuage (3dBird de *Sketchfab*)

La scène *Unity* contient un modèle .fbx de nuage qui fait partie du décors de fond. Le modèle contient déjà des matériaux et textures.

5.7.7 Cloture (3dBird de *Sketchfab*)

La scène *Unity* contient un modèle .fbx de clôture qui fait partie du décors de fond. Le modèle contient déjà des matériaux et textures.

5.7.7 Tube (3dBird de *Sketchfab*)

La scène *Unity* contient un modèle .fbx de tube qui fait partie du décors de fond. Le modèle contient déjà des matériaux et textures.

5.7.8 Roche (3dBird de *Sketchfab*)

La scène *Unity* contient un modèle .fbx de roche qui fait partie du décors de fond. Le modèle contient déjà des matériaux et textures.

5.8 Animation avec une ligne de temps

Le projet contient des animations produites avec l'éditeur de lignes de temps de *Unity* (Timeline) et qui sont déclenchée avec le scripting en *C#*. Notamment, le fichier *transform.cs* contient le code qui fait avancer l'oiseau via une translation et joue l'animation de 'Vol' qui fait une rotation sur les ails et la queue de l'oiseau en fonction du temps. Le fichier *input.cs* contient le code qui joue l'animation de 'up', quand la touche espace est appuyée et de 'down' quand la touche espace est relâchée qui fait une rotation horaire et anti-horaire sur l'oiseau.

5.9 Animation avec une machine à états

Le projet contient les animations d'oiseau qui sont contrôlées par l'Animator, l'outil de machine à état de *Unity*. L'Animator permet de contrôler la trajectoire du vol d'oiseau selon la touche du clavier enfoncée. Le projet commence par l'état *fly* qui est son animation par défaut, donc l'oiseau commence à voler. Si aucune touche n'est appuyée, il va changer pour l'état *down* et après

l'animation de celle-ci, il va retourner à l'animation par défaut *fly*. Selon les actions du joueur (appuyer ou relâcher espace), l'état va changer pour *up* et l'oiseau va prendre de l'altitude ou, au contraire, l'état va changer pour *down* et l'oiseau va descendre. Quand l'oiseau rentre en collision avec un obstacle, le jeu est terminé et l'oiseau se retrouve donc en son état de sortie *exit*.

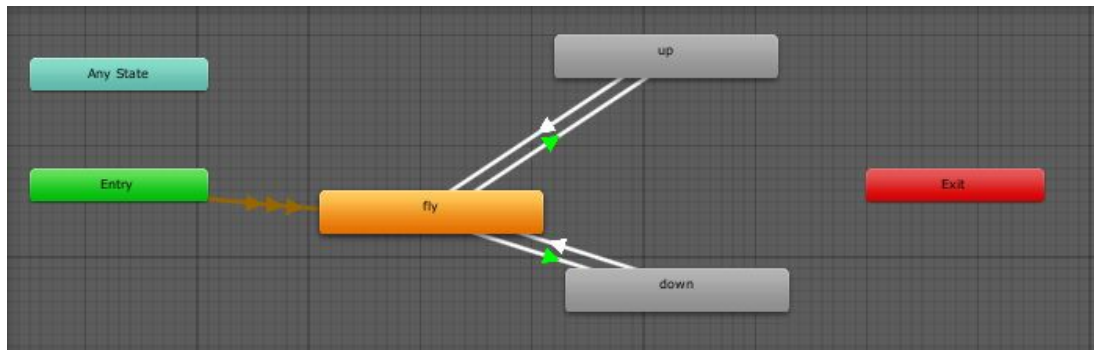


Figure 5: Schéma de la machine à états dans le projet Flappy Bird.

5.10 Effets visuels

5.10.1 Effets visuels en boucle

Les effets visuels en boucle de la page d'accueil sont dans la présentation du titre, le tuyau et dans l'oiseau Flappy. À l'aide d'un script en C#, tous les éléments sont en mouvement dans le menu lorsque nous sommes dans cette scène.

5.10.2 Effet visuel par le clic

Le fichier *MainMenu.cs*, s'occupe de la gestion du jeu. Il permet de commencer une partie, de visiter la page d'aide ou les instructions du jeu sont inscrite et de quitter. Lorsque la partie est terminée, le joueur est amené dans la scène du menu. Lorsque l'utilisateur clic sur un bouton, le bouton grandi avant d'exécuter l'action demandé par l'utilisateur.

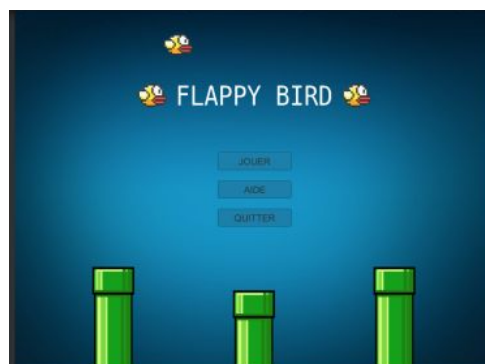


Figure 6: Effets visuels dans le projet Flappy Bird.

5.10.3 Effet de particules

L'effet de particules est ajouté sur l'oiseau dans le jeu. Lorsque l'utilisateur relâche la touche ESPACE, un système de particule est déclenché pour 1 seconde. Le système de particule est déclenché dans le fichier *transform.cs*, le système de particule provient du *GameObject* de l'oiseau, *FELPUDO_PARENT*. La figure ci-dessous présente un petit visuel sur le système de particule.



Figure 7: Système de particule

6. Ressources

6.1 Ressources originales

Les éléments de la scène Maya ont été modélisés par nous-mêmes.

6.2 Ressources externes

Les modèles utilisés pour le jeu *Unity* sont les suivants:

<https://sketchfab.com/3d-models/flappy-bird-em-realidade-aumentada-ar-tutorial-4b7602ce3dd1472d905d80a3e810c483>

Les images pour le menu du jeu *Unity*, proviennent de <https://www.cleanpng.com/> et de <https://www.pinterest.fr/>.