

Document de Design

Document réalisé par

Jérôme Collin
Maxime Giard
Elrick Carrier-Picard
Cédric Lafond-Mercier

Document remis à

Philippe Voyer

ANI2012 - Programmation en animation

27 octobre 2019

Table des matières

Sommaire	3
Interactivité	4
Fonctionnalités	4
1. Interactivité	4
2. Texte	4
3. Dessin vectoriel	5
4. Traitement d'image	5
5. Séquence d'images	5
6. Audio	5
7. Musique	6
8. Système de particule	6
Ressources	7
Présentation	8-9

Sommaire

Nous avons décidé de recréer une “visionneuse interactive” d’images et de vidéo pour notre projet. Nous nous sommes donc basés sur l’idée d’un viseur d’appareil photo. Pour rendre l’expérience intéressante, nous avons voulu guider l’utilisateur dans l’interface afin qu’il ou elle puisse s’aventurer dans l’ensemble des interactivités possibles facilement dans un environnement agréable à la vue et à l’ouïe.

Dans cette optique, nous avons décidé d’utiliser une interface simple d’utilisation avec des instructions implémentée sous forme de textes dans notre programme. L’interface est composée de deux sections: une affichant les commandes et une autre affichant la scène où sont positionnés les images et les vidéos. Chaque scène aura la possibilité d’être altérée selon des méthodes de traitement d’image. Nous avons décidé d’ajouter une façon de changer le degré d’exposition de l’image, de rendre l’image floue-voulant simuler un ajustement du focus d’appareil photo et finalement de pouvoir inverser les couleurs .

De plus, nous avons choisi d’utiliser des scènes variées pour offrir le plus de possibilités à l’utilisateur en ce qui concerne les couleurs ainsi que les méthodes de traitement d’image. Ainsi, nous avons utilisé des scènes avec des couleurs contrastées entre elles.

L’ajout de sons est utilisé afin de confirmer à l’utilisateur que sa commande a bel et bien été prise en compte lorsque l’on appuie sur une touche ou sur les boutons de la souris. Cela offre donc une expérience plus agréable puisque le retour est immédiat après avoir effectué une commande.

Interactivité

La première forme d'interactivité est le lancement de notre logiciel en cliquant sur le logo et en le gardant enfoncé jusqu'à la fin du chargement. Il est ensuite possible de faire apparaître quatre scènes. Deux qui montrent des photos embrouillées, une qui montre une vidéo et une dernière qui montre une photo d'une forêt avec un système de particules qui simule des lucioles. Sur les deux premières photos, il est possible de faire un focus sur celle-ci en cliquant avec la souris. Il est aussi possible d'inverser les couleurs des photos en appuyant sur la touche "v". Sur les quatre scènes affichables, il est possible de changer l'exposition des images avec les flèches "Haut" et "Bas".

Fonctionnalités

1. Interactivité

Il sera possible d'interagir sur l'animation d'abord (1) lorsqu'on appuie sur le logo avec la souris. (2) Puis, avec les touches 1-2-3-4 du clavier, il sera possible de changer de scène. On utilise un switch pour faire changer les fenêtres selon la touche appuyée. (3) Les touches haut et bas sont utilisées pour changer l'exposition de la scène. Pour cela, un "if statement" est utilisé pour détecter la touche. (4) Le clic de la souris sur la scène sert à effectuer un focus sur la scène. Le programme utilise les fonctions "mousePressed" et "mouseReleased" pour changer le statut de la variable booléenne "isMousePressedLeft". Si le clic est détecté, une nouvelle image sans blur est affichée sur l'image brouillée. (5) La touche "v" permet d'inverser les couleurs sur les deux premières scènes.

2. Texte

Il y a 7 lignes de texte qui soient affichées dans le programme. La première est "Clique sur le logo". Ce texte est destiné à guider l'utilisateur dans l'interface. Si on clique sur le logo, le premier texte disparaît et fait place au deuxième texte: "chargement". En utilisant plusieurs "if statements", on a pu compter le nombre de frames depuis que l'utilisateur a cliqué sur le logo. Ce nombre, stocké dans la variable "frame", est ensuite utilisé pour vérifier l'activation de plusieurs booléens. Grâce à ce système, l'utilisateur peut arrêter l'animation, soit lâcher le bouton de la souris, et le texte va retourner à son état initial. Quand le logo finit son animation, la variable booléenne "animFin" devient vraie. Le menu contenant le reste du texte peut alors apparaître. Il contient les cinq autres lignes de texte. Le titre "Menu Clavier" est destiné à simplifier la compréhension des instructions puisque plusieurs personnes qui ont essayé le programme cliquaient sur le texte plutôt que sur les touches. La police de caractère utilisée pour le titre n'est pas la même que le reste du texte. L'animation du texte apparaît quand l'utilisateur utilise la fonction pour inverser l'image. La ligne de texte "v: inverser l'image" devient alors bleu.

3. Dessin vectoriel

La plupart du dessin vectoriel se situe dans la classe “viewFinder”, ou on utilise plusieurs formes pour recréer ce qu’on voit sur l’écran d’un appareil photo. Dans cette classe on retrouve dix cercles, huit rectangles et un carré. Les autres formes sont situées dans la partie de gauche du programme. Trois points apparaissent à côté du texte “chargement” lors de l’animation du logo. L’affichage de ces points utilise le même système que l’affichage des textes de la partie de chargement du programme. C’est-à-dire que les frames sont comptés et stockés à partir du clic de la souris et réinitialisés si l’animation n’est pas terminée. La dernière forme est la ligne à gauche du menu. Celle-ci apparaît avec le texte du menu et ajoute à l’esthétique du programme.

4. Traitement d’image

La première forme de traitement d’image est le “blur” qui est appliqué sur les images des deux premières fenêtres du programme. La deuxième est une inversion de couleurs qu’il est possible d’appliquer à ces deux mêmes images. Le flou est appliqué automatiquement sur les images et doit être enlevé par l’utilisateur en cliquant dessus. On recrée ainsi un focus d’appareil photo. Pour l’inversion, le programme vérifie si la touche “v” est activée et si c’est le cas, les couleurs de l’image sont inversées. Il est aussi possible de modifier l’exposition des quatre fenêtres avec les flèches du haut et du bas sur le clavier. L’effet est obtenu en avec un rectangle noir par dessus les images avec lequel il est possible de modifier l’alpha. Ainsi, lorsque l’alpha est près de 0 on a une exposition plus claire et lorsque l’alpha est près de 255 on ne voit tout simplement plus l’image.

5. Séquence d’images

La séquence d’images est le logo qui doit être activé pour voir le programme. La séquence a été animée dans After Effects, puis exportée en jpg. Pour jouer la séquence d’animation, une variable appelée “frame” est incrémentée de 1 par frame qui joue lorsque l’utilisateur clique dans la zone du logo. Si l’utilisateur cesse de cliquer avant la fin de l’animation, la variable “frame” est remise à zéro. Ainsi, le logo peut être joué autant de fois que l’on veut, mais si l’animation se termine, on ne peut plus la faire jouer. La séquence d’image est chargée selon cette variable puisque si on utilisait la variable “FrameCount”, le logo ne recommencerait pas du début.

6. Audio

Il y a trois éléments audio. Le premier est l’effet sonore grave qui joue durant l’animation du logo. Celui-ci utilise le même système que le logo pour jouer et se réinitialiser. Un “click filtré” peut être entendu lors de la transition entre les fenêtres d’affichage et un autre quand on pèse sur les touches haut et bas (exposition). Si on tient les touches enfoncées, l’effet sonore joue très rapidement, ce qui aide à comprendre que l’exposition est encore en changement et qu’elle peut être modifiée en continu (touche maintenue).

7. Musique

La séquence de “musique” est constituée de 3 notes qui sont activées dans le temps. Il ne s’agit pas d’un seul élément audio, mais bien de trois fichiers qui sont synchronisés aux trois points de chargement. Il s’agit d’une approche minimaliste, mais qui ajoute à l’immersion du programme.

8. Système de particule

Un système de particules est utilisé par-dessus une photo d’une forêt la nuit pour donner l’impression qu’il y a des lucioles. Ces particules ont été générées à partir d’un système d’itérations, c’est-à-dire que chaque particule a un état enregistré en mémoire qui change avec le temps et, lorsqu’il n’y a plus d’état à afficher pour cette particule, celle-ci disparaît et est remplacée par une nouvelle. Pour représenter un mouvement constant d’une luciole, une vitesse angulaire a été utilisée de façon à créer une accélération et une décélération sur l’axe des ordonnées. La taille des particules varie aléatoirement de façon à créer une impression de profondeur dans l’image.

Ressources

Vidéo de la forêt: Prise par Cédric

Image de la forêt: Image libre de droits trouvés sur Internet par Elrick

Photo Skate: Prise par Maxime Giard

Photo Percé: Prise par Maxime Giard

Polices d'écritures: Prises sur un site de polices de caractères libres de droits.

Logo : Fais par Jérôme dans After Effects et Photoshop.

Effets sonores: Créé par Maxime Giard dans Ableton.

Son grave du logo: Créé par Maxime Giard dans Ableton.

Notes de musique: Créé par Maxime Giard dans Ableton.

Librairies utilisées: Librairie vidéo de Processing (G-streamer) et Librairie audio de Processing

Présentation

Nous sommes quatre étudiants au BASA en deuxième année.

Cédric veut s’orienter vers le domaine du jeu vidéo, il voit donc ce cours comme une façon de se familiariser avec la programmation, qui est omniprésente dans ce domaine. Il a un peu de connaissances en analyses de base de données.

Tâches accomplies:

Cédric a contribué tout au long du projet à assurer que le programme soit fonctionnel et a proposé des idées et des solutions. Il a travaillé au design et à l’idéation des fonctions du programme au tout début du processus de création. Il a modifié un des systèmes de particules vus dans le cours pour créer l’illusion de lucioles. Il est aussi responsable de la classe “SoundEffects” et de l’intégration des sons dans le programme. Cédric a aussi contribué à la rédaction du document de design. Il est également responsable du fonctionnement de la classe “Viewer3” (vidéo).

Elrick n’a pas de bases en programmation, mais il souhaite apprendre pour pouvoir l’utiliser dans le monde des effets visuels au cinéma. Notamment dans les logiciels Maya et Houdini.

Tâches accomplies:

Elrick a réussi à faire apparaître l’image de forêt derrière les particules et il a modifié la taille et le placement du système pour que le tout s’affiche dans la fenêtre de visionnement 4. Il a aussi trouvé comment rendre la taille des particules aléatoire. Il a finalement travaillé au prototype du système de flou.

Maxime a déjà fait un peu de programmation avec des langages visuels comme Max/MSP et Pure Data. Il souhaite approfondir ses connaissances en programmation plus classique pour pouvoir l’utiliser dans ses projets.

Tâches accomplies:

Maxime a contribué tout au long du projet à assurer que le programme soit fonctionnel et a proposé des idées et des solutions. Il a participé au design visuel et il a fait une grande partie du prototype avec Jérôme. Ensuite, c’est celui qui a créé l’architecture du programme et qui s’est assuré de l’intégration adéquate de chaque nouvelle partie de code. Maxime est responsable de la majorité de ce qui est écrit dans le premier onglet “tp01” et des éléments dans les autres onglets. Il s’est aussi occupé de fournir différents éléments externes originaux (photos, sons et musique). Finalement, il a vérifié une dernière fois l’ensemble du code pour s’assurer que chaque élément soit utilisé et soit à son endroit respectif.

Jérôme est intéressé par la programmation pour sa versatilité. Il a déjà fait le cours GLO-1901 Introduction à la programmation avec Python, donc il a de bonnes bases, mais il souhaite apprendre à les appliquer dans un programme ou dans des scripts.

Tâches accomplies:

Jérôme a contribué tout au long du projet à assurer que le programme soit fonctionnel et a proposé des idées et des solutions. Il a créé l'esthétique du programme et le logo. Il a fait une très grande partie du prototype avec Maxime. C'est celui qui est responsable de la création de la majorité des classes (logo, viewFinder, UI, Viewer1-2). Jérôme, grâce à ses bases en python, a grandement contribué au code et a développé l'interactivité du programme. Il s'est occupé de la création du logo (Photoshop) et de son animation (After Effects). Finalement, il a vérifié une dernière fois l'ensemble du code pour s'assurer que chaque élément soit utilisé et soit à son endroit respectif. Jérôme a aussi contribué à la rédaction du document de design.