

Programmation en animation

(ANI-2012)

Travail Pratique 1 - Animation Interaction 2D

Document de Design

Nom du projet
Processing Basics

Par l'Équipe 1

Emma Verneuil
Éric Harvey
Hélène Carignan
Sandie Valiquette
William Brochu

Présenté à
Philippe Voyer

Remis le
29 octobre 2018

Faculté d'aménagement, d'architecture, d'art et de design, Université Laval

***** ATTENTION, il est important d'importer la librairie Minim (lecture des sons) et la librairie G-Streamer (lecture des vidéos) dans Processing pour que le programme puisse s'exécuter *****

***** Pour ouvrir la version compilée MacOSX64 vous devez avoir la version de Java8 *****

1.1. Sommaire

Pour décider quel type de projet nous allions réaliser, nous avons tenu plusieurs rencontres via Hangouts, où chaque membre de l'équipe devait participer. Chaque membre a d'abord imaginé un prototype pour le projet et en a fait la présentation à l'équipe. Nous avons exploré plusieurs avenues avant d'arrêter notre idée sur une interface éducative. Inspirée du modèle de *Hello Processing* (voir références externes) nous avons retenu l'idée d'Hélène.

Nous avons élaboré un programme interactif de style éducatif à travers lequel l'utilisateur découvre les différentes applications de base que l'outil *Processing* peut offrir dans la création d'éléments artistiques visuels et dans l'apprentissage du codage.

L'interface se veut ludique, fonctionnelle et épurée. Elle comporte plusieurs éléments interactifs, et elle est composée de formes simples et de couleurs variées.

Au lancement du programme, s'affichent : le titre du programme *Processing Basics*, un bouton *Restart*, un bouton de gestion du son (*ON/OFF*), une trame sonore, ainsi que cinq bulles avec dans chacune d'elles un contenu en mouvement ou statique, prêtes à recevoir des commandes. Un texte animé apparaît ensuite, se déplaçant de la gauche vers la droite.

Nous avons choisi le blanc pour l'arrière-plan afin de mettre les éléments de contenu en valeur. Les bulles dans l'interface sont disposées en étoile de façon à mettre le point focal sur le titre. Elles sont de différentes grandeurs pour briser le modèle linéaire et pour créer un effet de mouvement. Pour simuler un certain volume et suggérer de la profondeur, nous leur avons ajouté un contour de couleur grise et lors d'un mouse-over, on observe un effet de surbrillance, ainsi qu'un léger agrandissement de la bulle sur laquelle le mouse-over est effectué.

Dans la réalisation de notre programme, nous avons rencontré plusieurs obstacles et dû trouver des solutions pour chacun de ceux-ci. La première difficulté reposait dans le fait de devoir imaginer un programme d'une ampleur X, sans avoir encore toutes les connaissances pour connaître sa faisabilité, évaluer la grosseur du projet, et ce, en situation d'apprentissage. Deuxièmement, la gestion des différentes cadences (frame rate) du vidéo et du GIF nous ont demandé beaucoup de réflexion. Nous avons aussi dû faire des choix pour diminuer la lenteur du projet (en raison des nombreux éléments).

1.2. Interactivité

Lorsque l'utilisateur glisse la souris sur une des bulles, celle-ci change de teinte (grâce à une variation du degré de transparence, le pourcentage Alpha). Aussi lors du mouse-over sur les différentes bulles, on observe un léger agrandissement de celles-ci, sans agrandissement du contenu. Lorsque l'utilisateur clique sur une bulle, la bulle et son contenu s'agrandissent et dévoilent le nom des fonctionnalités, et dans certains cas des instructions (il s'agit d'une précision sous forme de texte) tout en continuant l'animation du contenu.

Quand l'utilisateur clique en dehors de la bulle celle-ci disparaît, accompagnée d'un son d'éclatement. Il peut alors voyager ainsi sur chacune des bulles jusqu'à ce qu'elles aient toutes été explorées. Avec ce principe simple, l'utilisateur sait quel contenu il a visionné ainsi que le celui qui lui reste à regarder. L'utilisateur peut toujours choisir de repartir l'animation s'il le désire en cliquant sur le bouton *Restart*.

Grâce au bouton *ON/OFF*, il peut choisir d'arrêter la musique de fond (de style instrumental) lorsqu'il le désire. Quand la musique est à OFF, le bouton change d'aspect (l'image change et devient une note de musique avec un trait rouge). L'utilisateur peut aussi décider de remettre la musique quand il le veut (l'image change et redevient une simple note de musique).

L'utilisateur peut aussi en apprendre davantage sur *Processing* en cliquant sur le titre du programme (*Processing Basics*) ce qui le guidera vers le site internet de *Processing* (processing.org).

Dans deux des cinq bulles, l'utilisateur est appelé à intervenir davantage.

Dans la bulle *Fractal Tree*, dès le démarrage du projet, ce dernier est tout «épanoui», mais avec le mouvement de la souris (mouse-over) l'utilisateur peut faire varier le degré « d'épanouissement » de l'arbre.

Dans la bulle contenant le « Virtual drum » l'utilisateur peut composer une musique à l'aide de son clavier. Par exemple, la touche H active le son du Hi-Hat (pour une description plus détaillée des éléments jouables, voir la section « Musique »). L'utilisateur peut alors créer son propre rythme, sa propre composition. Chacune des formes vectorielles est associée à un son et à une touche du clavier. Quand une de ces formes est appelée par l'utilisateur, elle joue le son qui lui est attribué et prend une couleur rouge.

1.3. Fonctionnalités

Pour aller jusqu'au bout de l'exercice éducatif, nous avons choisi de travailler avec toutes les fonctionnalités, c'est-à-dire: l'interactivité, les éléments textuels, le dessin vectoriel, le traitement d'image, la séquence d'images, l'utilisation de l'audio, de la vidéo, de la musique avec une rétroaction visuelle, d'un système de particule ainsi que d'un système dynamique.

a. Interactivité

L'utilisateur peut interagir dans le programme par de nombreuses actions différentes. Les actions sont déclenchées lors d'événements des contrôleurs interactifs clavier et souris.

Les «clics» :

Un clic sur le bouton *ON/OFF* permet d'éteindre ou de repartir la musique de fond.

Un clic sur une bulle, l'agrandit elle et son contenu (sauf dans le cas de la bulle contenant le «Virtual drum»), appelle un son «pop» et fait aussi apparaître une description textuelle de son contenu.

Un clic à l'extérieur de la bulle (lorsqu'elle a été sélectionnée auparavant) la fait disparaître et appelle un son d'éclatement.

Un clic sur la bulle du vidéo active le volume de celui-ci.

Un clic sur la bulle *Fireworks* appelle un son de feux d'artifice.

Un clic sur le mot «Processing», premier mot de la ligne de texte au bas de l'interface, mène, grâce à un lien URL, au site de *Processing* (processing.com).

Un clic sur le bouton *Restart* permet de repartir l'animation du début (sauf le bouton du son *ON/OFF* qui reste en place puisqu'il s'agit davantage d'une configuration).

Les mouse-over :

Le glissement de la souris autour et sur la bulle *Fractal Tree* anime ce dernier (épanouissement de l'arbre).

Un mouse-over sur le bouton *ON/OFF* génère un son et applique un effet de surbrillance (une teinte).

Un mouse-over sur les bulles agrandit légèrement la bulle et applique un effet de surbrillance (teinte).

Un mouse-over sur le bouton *Restart* applique un effet de surbrillance (teinte).

Le clavier :

Certaines touches du clavier (C, H, R, S, B, F, 1 et 2) permettent de créer une musique dans la bulle contenant le «Virtual drum» (*Musical Sequence*) (et de colorer les éléments appelés).

Les difficultés rencontrées :

C'est durant la création de l'élément d'interactivité associé aux bulles que nous avons rencontré notre défi principal. Nous avons créé en parallèle les codes des contenus de chacune des bulles ainsi que celui de l'interface principale (celui qui contient les bulles, le titre, le texte défilant, les boutons *Restart* et celui de la musique). Le premier défi était de regrouper tous les codes dans un programme principal, puisque cela a généré beaucoup de lourdeur et d'interférences entre les différentes commandes. Le deuxième défi s'est présenté lorsqu'est venu le temps d'intégrer le contenu de chacune des bulles au dit programme principal. Il n'y avait pas une recette qui pouvait s'appliquer à toutes les bulles. Chaque intégration provoquait une série de réactions différentes et il a fallu travailler fort pour que les bulles et leur contenu adoptent un même comportement.

b. Textes

L'animation a six éléments textuels, quatre polices de caractères différentes ainsi qu'un élément textuel animé. Nous avons choisi d'utiliser des textes en anglais puisque cela permettait l'utilisation de textes plus courts et s'accordait mieux à l'espace disponible de notre projet.

Le titre du programme, « *Processing Basics* » (au centre de l'interface) est un texte statique, sa police est CafBrewery.

Processing Basics

Pour ce qui est de la ligne de texte animé «*a popular tool to learn how to code and simulate ideas quickly within the context of visual arts and technology*» (en bas de l'interface), sa police de caractère est Offerings.

Le premier mot, «*Processing*», quant à lui, est statique (il se distingue par une couleur bleue), sa police de caractère est aussi Offerings. Il contient le lien URL vers le site de *Processing* (processing.org).

Processing a popular tool to learn how to code and simulate ideas quickly within the context of visual arts and technology

Le texte du bouton *Restart* est statique et sa police est Meatloaf (la déclaration de cette police se fait dans la classe *Button*).

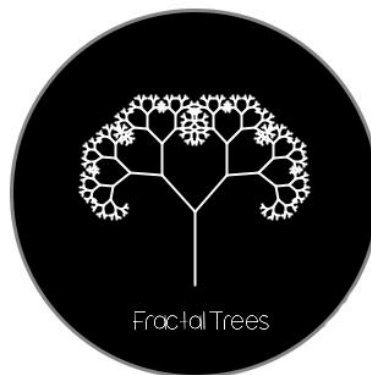


Sans mouse-over

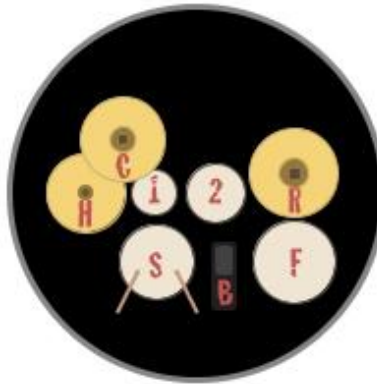


Avec mouse-over

Le texte qui apparaît dans chacune des bulles lorsque l'utilisateur clique sur l'une d'elles est statique et sa police est Offerings. La création des différents éléments textuels des bulles se trouve dans la classe *Bubble*.



Les lettres qui apparaissent sur les éléments de la bulle *Musical Sequence* sont statiques et leur police est Cracked. Ces éléments textuels sont créés directement dans la classe *DrawDrum*

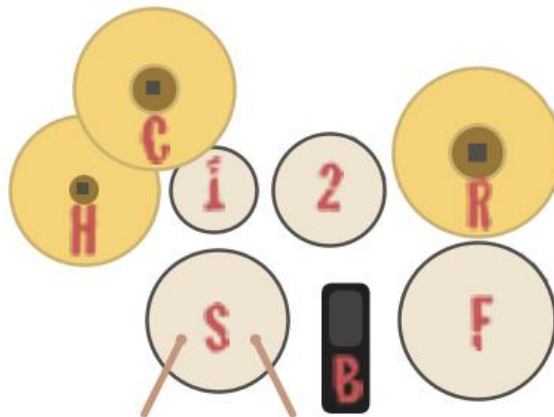


c. Dessin vectoriel

Le programme dessine cinq types de primitives vectorielles, dans la bulle *Musical Sequence* (le « Virtual drum ») (dans l'image ci-dessous, le texte est flou, car nous avons grossi l'image). Contrairement aux autres bulles, lors du clic sur celle-ci, le contenu ne s'agrandit pas afin de laisser la place à la création d'une légende pour aider l'utilisateur à comprendre son fonctionnement.

Les cinq primitives vectorielles utilisées :

- Ellipse : Crash (C), Hi-Hat (H), Snare (S), Tom 1(1), Tom 2 (2), Bass Drum (B) et Ride (R)
- Carré : un carré gris est dessiné au centre de la Ride, de la Crash et du Hi-Hat
- Rectangle : un rectangle gris et un noir dessinent le Bass Drum
- Ligne : la tige des baguettes
- Point : l'embout des baguettes



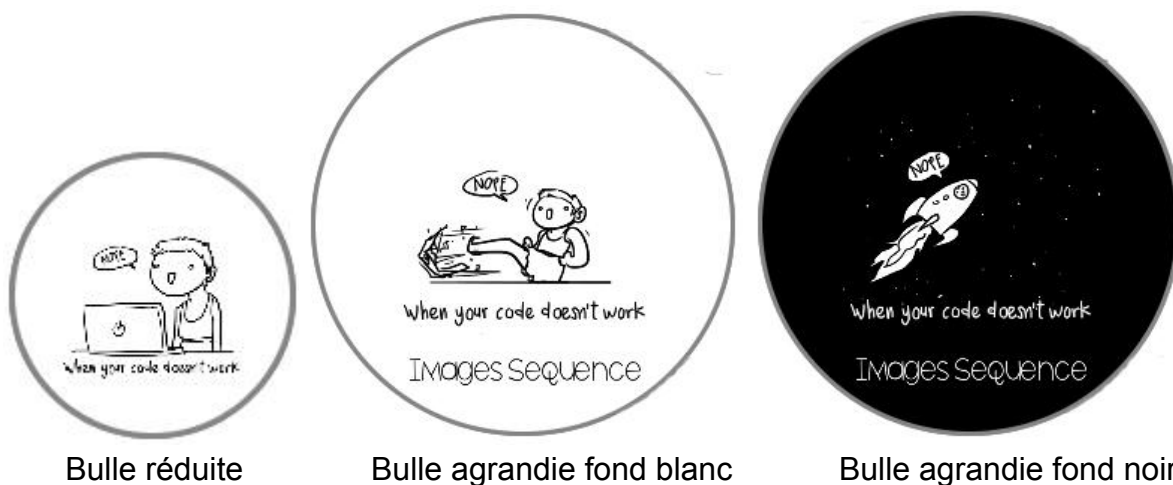
d. Traitement d'image

Une technique de traitement d'image a été utilisée pour modifier l'apparence du bouton *ON/OFF* de la musique de fond, du bouton *Restart*, ainsi que des bulles lors du mouse-over : une légère surbrillance.

e. Séquence d'images

Nous avons créé un GIF à partir d'une séquence d'images pour la bulle *Images Sequence*. Nous avons procédé à la décomposition d'un GIF existant (sur internet) grâce à des captures d'écran. Des ajustements et des modifications ont été opérés dans le logiciel Photoshop (ajout de texte, adaptation de la taille du GIF à notre projet et quelques modifications de couleurs).

Quand est venu le temps d'intégrer le GIF au programme principal et de le faire entrer dans sa bulle, nous avons choisi de garder le comportement de «zoom de la bulle», c'est-à-dire que le GIF s'adapte à la bulle agrandie. De plus, lors d'un mouse-over sur cette bulle, on observe un changement de couleur (effet de surbrillance) qui se distingue bien sur le fond noir, mais qui est difficile à voir sur le fond blanc. Nous avons aussi eu à jouer avec le *frame rate* pour que le GIF s'anime dans la cadence désirée, puisqu'il fallait le ralentir sans ralentir le reste du projet. Le texte de la bulle a subi une condition afin qu'il reste apparent dans le cas où le GIF est sur fond blanc (texte noir) et dans le cas où le GIF est sur fond noir (texte blanc).



f. Audio

L'animation utilise plusieurs éléments audio.

Premièrement, un élément sonore «pop» est déclenché lors du clic sur une bulle et un élément sonore «éclatement» est déclenché lors de la disparition de la bulle (un clic en dehors de la bulle sélectionnée).

Deuxièmement, une musique de fond accompagne l'animation lors du lancement du programme. L'utilisateur a le choix de l'arrêter par un clic sur l'image de la note de musique (OFF) ou de la redémarrer (ON) par un nouveau clic sur la note.

Troisièmement, un mouse-over sur le bouton *ON/OFF* produit un son «Bongo».



Bouton *ON/OFF*

Quatrièmement, une trame de son accompagne les feux d'artifice lorsque l'utilisateur clique sur la bulle *Fireworks*.

Cinquièmement, lors du clic sur la bulle contenant le vidéo, son volume s'active.

Dans le but de prévenir et d'éviter toute erreur possible lors du lancement du programme, nous avons converti les sons *.mp3* en *.wav* étant donné que *Processing* avait un problème de reconnaissance de propriété avec les fichiers *.mp3*.

g. Vidéo

Un fichier vidéo est utilisé comme source pour afficher du contenu dans la bulle *Vidéo*. La fréquence d'images (*frame rate*) a été modifiée à l'externe à l'aide du logiciel After Effect afin qu'il soit fluide une fois importé dans le projet. Nous lui avons attribué une fréquence de 60fps afin de contrer l'effet du *frame rate* appliquée au GIF.

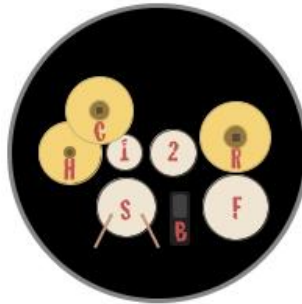
Le vidéo a été compressé et sa qualité a été réduite afin de ne pas alourdir le projet. Ainsi, le problème de lenteur que nous observions depuis son intégration a été presque réglé en totalité (persiste tout de même au démarrage du projet).

Au clic, le vidéo s'agrandit comme la bulle qui la contient et son volume s'active (puisque lorsque la bulle n'est pas cliquée, le son est muet pour ne pas interférer avec les différents sons du projet).

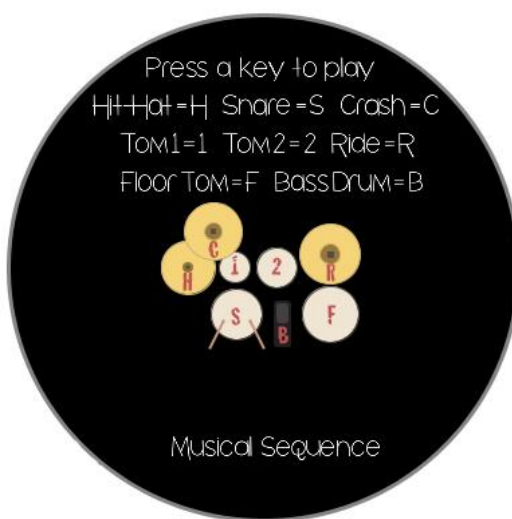


h. Musique

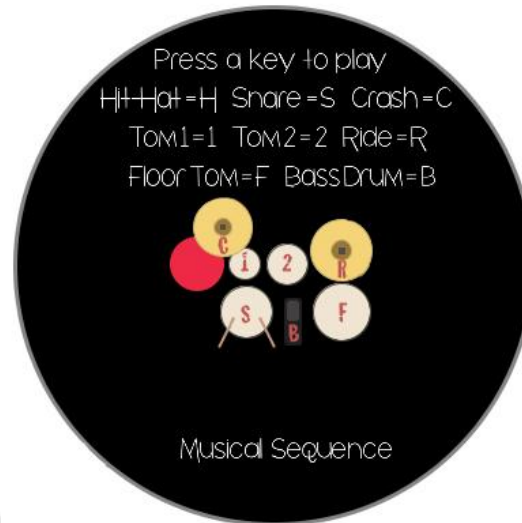
L'utilisateur peut créer une musique à partir d'une action interactive, le clavier. Lors du clic sur la bulle du *Musical Sequence*, une légende apparaît pour expliquer à quoi correspond chacune des lettres. À plusieurs touches du clavier sont associés un son et un élément (primitives vectorielles reconstituant un élément du *drum*). En appuyant sur une de ces touches (H, C, S, 1, 2, B, R, et F), cela appelle un son spécifique, ainsi qu'une rétroaction visuelle puisque la couleur de l'élément joué devient rouge (il y a là une interaction entre l'image et le son). Par exemple, lorsque la touche H (qui correspond au Hi-Hat) est appuyée, le Hi-Hat devient brièvement rouge pour montrer à l'utilisateur quel élément du *drum* il vient de jouer, et le son correspondant au Hi-Hat joue. Chaque «touche» à appuyer est indiquée sur chacun des éléments, mais aussi dans la légende textuelle.



Bulle réduite



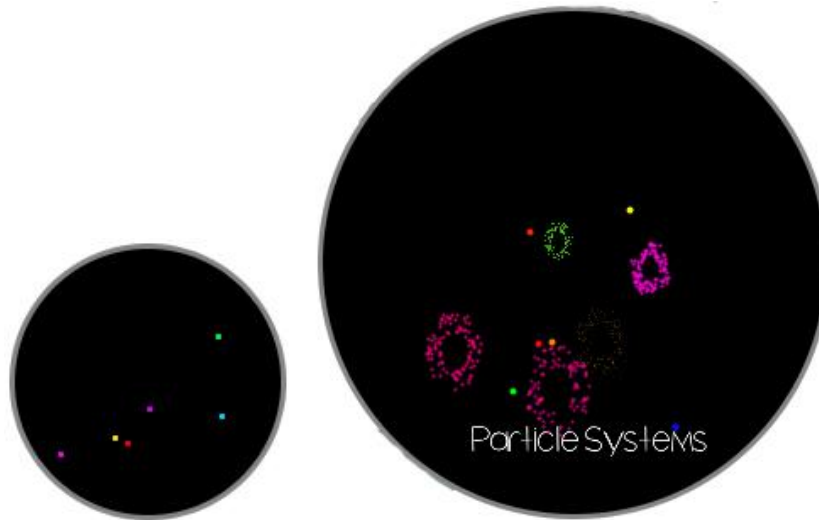
Bulle agrandie



Bulle jouée (keypressed)

i. Système de particules

L'animation utilise un système de particule pour la composition de la bulle *Fireworks*. Pour sa réalisation, nous nous sommes inspiré du code du même nom, créé par Daniel Shiffman (voir références externes), mais avons modifié l'accélération et la vitesse. Nous avons choisi d'enlever les traînées pour simplifier l'intégration au programme principal, mais surtout pour garder la performance du programme en raison des nombreux éléments qui le composent. Le clic sur la bulle *Fireworks* les fait s'agrandir pour s'adapter à la bulle et appelle un son de feux d'artifice.



Bulle réduite

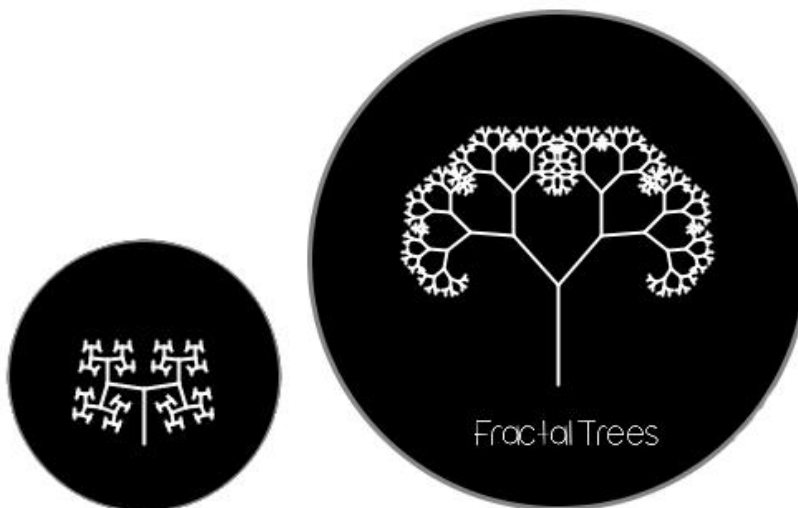
Bulle agrandie

j. Système dynamique

L'animation utilise un système dynamique dans la composition de la bulle *Fractal Tree*. Nous avons utilisé l'exemple *Recursive Tree* de Daniel Shiffman (voir les références externes) et avons ajusté la longueur du tronc (rapport corps/tête).

Lors du clic, l'arbre grandit comme la bulle qui le contient. Ainsi, pour qu'il s'adapte parfaitement nous avons changé la longueur et l'épaisseur des traits lorsque le zoom est effectué (bulle agrandie).

Un mouse-over autour de l'arbre crée une rétroaction puisqu'il permet de varier l'épanouissement de l'arbre.



Bulle réduite

Bulle agrandie

1.4 Ressources

a. Ressources externes

Dans cette section, vous retrouverez toutes nos inspirations pour la création des différents éléments de notre projet (l'entièreté, une partie ou même dans certains cas, une seule ligne de code ont été réutilisés). Dans le cas d'inspirations, vous trouverez le terme "source d'inspiration", dans le cas d'un code réutilisé, mais modifié et adapté, vous trouverez le nom du membre ayant fait les modifications, et dans le cas d'une réutilisation complète du code, vous trouverez clairement indiqué "code/source utilisé".

Source d'inspiration du projet :

Processing. (S.D). *Interface de base*. Récupéré le 19 octobre de <http://hello.processing.org/>

Virtual Drum :

Source des sons du drum:

Sans-Auteur. (S.D). *Kit Pearl Real*. Récupéré de 19 octobre de <https://sampleswap.org/filebrowser-new.php?d=DRUMS+%28FULL+KITS%29%2FPearl+Real+Kit%2F>

Source d'inspiration pour l'importation des fichiers audio :

Sans-Auteur. (2014). *Playing a sound for a pressed key*. Récupéré le 19 octobre de <https://forum.processing.org/two/discussion/7247/playing-a-sound-for-a-pressed-key>

Source d'inspiration pour l'utilisation du « KeyPressed » :

Voyer, Philippe. (2018). *Keyboard*.

Interface de base :

Source d'inspiration pour la création un bouton menant à une action :

Ockerby, C (2017). *Processing, Creating a button class*. Récupéré le 19 octobre de <https://www.youtube.com/watch?v=z903vXot-Lg>

Source d'inspiration pour la création d'un bouton « Restart » :

Sans-Auteur. (2017). *Reset/restart game*. Récupéré le 19 octobre de <https://forum.processing.org/two/discussion/24517/help-to-reset-restart-game>

Source d'inspiration pour la création d'un lien URL :

Sans-Auteur. (2009). *Hyperlink*. Récupéré le 19 octobre de https://processing.org/discourse/beta/num_1257794997.html

GIF :

Source du code utilisé :

Voyer, Philippe. (2018). *Image Animation 1*.

Bouton on /off :

Source d'inspiration pour mettre en pause de la bande sonore :

Librairie Minim "Pause" de Processing 3

Source d'inspiration pour la création d'un code «keep stay» lorsque «mousePressed» :

Sans-Auteur. (2018). *Replace an Image on MousePressed and making it stay*. Récupéré le 19 octobre de <https://forum.processing.org/one/topic/replace-an-image-on-mousepressed-and-making-it-stay.html>

Source de la musique utilisée pour la trame sonore :

Official Audio. (2018). *Unknown Longing*. Récupéré le 19 octobre de https://www.youtube.com/watch?v=LE_nW2AVEbE. Modifiée par Éric Harvey (2018).

Source du son joué lors du clic sur le bouton on/off :

Universal SoundBank. (S.D). *Percussions Bongos*. Récupéré le 19 octobre 2018 de <http://www.universal-soundbank.com/bongos.htm>

Polices d'écriture utilisées :

Murphy, B. (2012). *Café and Brewery*. Récupéré le 19 octobre de <https://www.dafont.com/cafe-brewery.font>

Murphy, B. (2017). *Meatloaf*. Récupéré le 19 octobre de <https://www.dafont.com/meatloaf.font>

7NTypes. (2017). *Offering*. Récupéré le 19 octobre de <https://www.dafont.com/offerings.font>

Xerographer. (2015). *Cracked*. Récupéré le 19 octobre de <https://www.dafont.com/cracked.font>

Feux d'artifice :

Source du code utilisé :

Shiffman, D. (2016). *Fireworks*. Récupéré le 19 octobre de <https://www.youtube.com/watch?v=CKeylbT3vXI>. Modifié et adapté par William Brochu (2018).

Arbre fractal :

Source du code utilisé :

Shiffman, D. (S.D.). *Recursive Tree*. Récupéré le 19 octobre de <https://processing.org/examples/tree.html>
Modifié et adapté par Hélène Carignan (2018).

Vidéo :

Sources d'inspiration pour l'intégration du vidéo :

Shiffman, D. (S.D.). *Movie*. Récupéré le 19 octobre de <https://www.processing.org/reference/libraries/video/Movie.html>

Sources d'inspiration pour l'importation du vidéo :

Voyer, Philippe. (2018). *Video Playback*.

Source de la vidéo :

Shiffman, D. (2015). *Introduction - Processing Tutorial*. Récupéré le 19 octobre de <https://www.youtube.com/watch?v=2VLalr5Ckbs>. Modification du frame en 60 fps (frame per second) pour les besoins du projet ainsi qu'une compression du fichier par William Brochu (2018) et modification des dimensions du vidéo à l'affichage du vidéo dans le projet par Hélène Carignan (2018).

Son de l'ouverture de la bulle Fireworks :

Universal SoundBank. (S.D). *Sons feux d'artifices*. Récupéré le 19 octobre de <http://www.universal-soundbank.com/feuxdartifices.htm>

Bulles :

Source d'inspiration pour la création de la classe bulle :

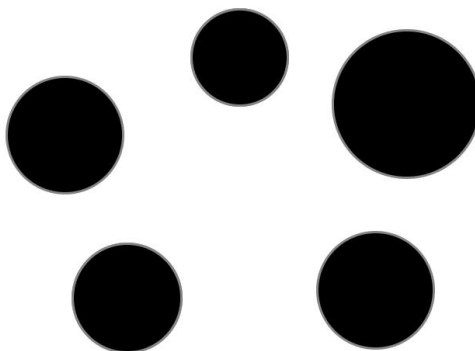
Shiffman, D. (2017). *Mouse Interaction with Objects*. Récupéré le 19 octobre de <https://www.youtube.com/watch?v=TaN5At5RWH8>

b. Ressources originales

Comme nous avons produit beaucoup d'éléments en nous inspirant de codes trouvés sur internet, nous avons choisi d'indiquer dans cette section, uniquement les éléments créés de "A à Z" par les membres de l'équipe.

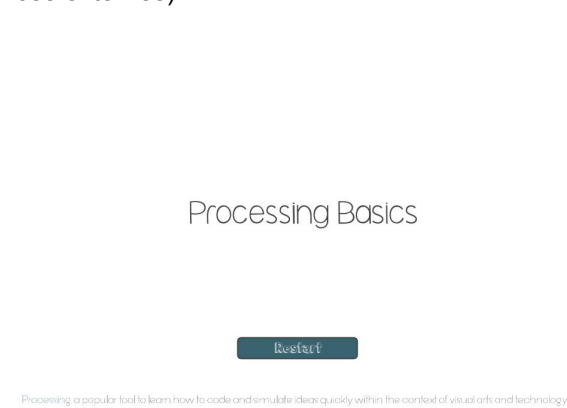
Bulles :

La création de la classe bulle, de leur géométrie des bulles, ainsi que leur texte (non apparents dans l'image ci-dessous) ont été créés par Hélène Carignan avec une inspiration du code de Daniel Shiffman (voir références externes).



Textes statiques et animation du texte:

L'entièreté des textes statiques et animés (sauf le texte des bulles) a été réalisée par Emma Verneuil (2018), l'animation du texte a été changée d'orientation par Hélène Carignan (2018) (voir sources d'inspiration dans les ressources externes).



Bouton Restart :

Le bouton Restart (interactivité et apparence) a été réalisé par Emma Verneuil (2018) et modifié par Hélène Carignan pour l'adapter à l'entièreté du projet (voir sources d'inspirations dans les ressources externes)



Bouton ON/OFF :

Icon 8. (2018). *White Music Icon*. Récupéré le 19 octobre de <https://iconsplace.com/white-icons/music-icon-18/>. Image modifiée (dessin d'un trait rouge lorsque le son est coupé) par Hélène Carignan (2018). L'action et l'interactivité du bouton ont été réalisées par Hélène Carignan (2018) (voir les sources d'inspirations dans les ressources externes)



Image originale



Image modifiée

GIF :

Sans-Auteur. (S.D). *Nope*. Récupéré le 19 octobre de <https://media.giphy.com/media/XUXHNme32gSqs/giphy.mp4>. Textes modifiés par Emma Verneuil (2018) et dimension de l'image modifiée par Hélène Carignan (2018)



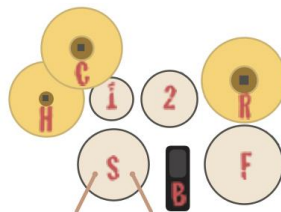
GIF original



GIF Modifié

Drum virtuel fabriqué à partir de primitives vectorielles :

Verneuil, E. (2018). *Virtual Drum*.



Son à l'ouverture et à la fermeture des bulles :

Harvey, Éric. (2018). *Pop*. [Sons enregistrés en studio]

1.5 Présentation

a. Présentation des membres

Le choix des membres de l'équipe a été fait pour regrouper des forces différentes, mais complémentaires, afin de mener à terme ce projet. Nous avons des membres expérimentés en programmation, en création d'interfaces usager, en animation 3D, en création multimédia, en composition et en rédaction. Chacun des membres avait des parties spécifiques qui lui étaient assignées. Toutes et tous participaient aux décisions, et aux réunions. Chaque semaine des réunions avaient lieu sur Hangouts ce qui nous permettait de valider le travail réalisé ainsi que d'évaluer celui qui restait à faire, mais surtout de répartir les différentes tâches entre chaque membre.

Emma Verneuil : Étudiante à temps plein au CASA. Dans le cadre de ses études, elle a déjà utilisé le langage SQL pour l'utilisation du logiciel Access, ainsi que la création de requêtes pour l'utilisation du logiciel ArcGIS. Cependant, elle n'a jamais fait de programmation à proprement parler avant ce cours.

Éric Harvey : Étudiant au CASA à temps partiel. Il est concepteur multimédia à l'école des technologies supérieures de Montréal au département des cycles supérieurs. Son travail consiste à aider les enseignants avec l'implantation d'éléments interactifs dans les cours et dans l'environnement d'apprentissage numérique Moodle. Il est débutant en programmation, il travaille quotidiennement Photoshop, Premiere, Audition, After effect et Illustrator.

Hélène Carignan : Étudiante en 2e année au CASA. Elle possède déjà une solide expérience dans le domaine de la programmation puisqu'elle a travaillé pendant environ 5 ans comme programmeur et ensuite 10 ans en tant que UI/UX Designer. Elle a beaucoup programmé en Delphi et elle était surtout spécialisée au niveau des interfaces (et des composants).

Sandie Valiquette : Étudiante à temps partiel au CASA depuis 2 ans, elle achève sa formation. Elle est bachelière de l'Université Concordia en production cinématographique, maman et entrepreneure dans le domaine des arts et des communications. Avant ce cours, elle n'avait jamais fait de programmation.

William Brochu : Étudiant en 3e année au BASA. Il est principalement intéressé par la 3D, mais aussi par l'animation de manière générale. Il n'a jamais été en contact de près ou de loin avec la programmation, ce cours est une véritable découverte.

b. Tâches réalisées par chacun des membres

Un **concept original** proposé par d'Hélène Carignan basé sur l'interface de hello.processing.org

William Brochu:

- Modification du frame du vidéo à 60 fps et compression du fichier vidéo
- Intégration du vidéo dans la bulle
- Ajout d'un fond de couleur à la bulle pour la faire correspondre avec le fond du vidéo (finalement supprimé pour un souci d'esthétisme)
- Annotation, modification et adaptation du code *Fireworks*
- Aide à la recherche pour la création des bulles

Hélène Carignan :

- Application d'un traitement d'image (teintes) sur le bouton *ON/OFF* et sur les bulles GIF et vidéo lors d'un mouse-over.
- Réglage du volume et du lancement du vidéo lors du démarrage du projet et lors du clic sur le bouton *Restart*
- Création des bulles avec avec effet «surbrillance», «loupe» et «disparition» pour l'interface de base
- Création de la classe *Bubble*
- Intégration et adaptation du code *Fractal Tree* et du code *Fireworks* dans le fichier principal
- Intégration du *Fractal Tree* et du *Fireworks* dans les bulles
- Modification du code *Fractal Tree*
- Création des sons à l'ouverture et à la fermeture des bulles
- Création des textes à l'ouverture de chaque bulle
- Intégration du GIF dans une bulle
- Modification du bouton *Restart*
- Modification du bouton *ON/OFF* (image, son et traitement d'image)
- Rectification des «bugs» du projet, amélioration et allègement du projet
- Aide à la rédaction générale (ajout d'éléments précis sur les codes)

Éric Harvey:

- Recherche de la trame sonore du projet
- Enregistrement en studio des sons pour l'ouverture et la fermeture des bulles
- Recherche du vidéo
- Rédaction du document de design
- Réalisation de l'image du bouton *ON/OFF* et intégration dans l'interface de base (finalement non utilisé en raison d'un problème de pixellisation de l'image)

- Organisation des séances de travail, planification des réunions, création des documents collaboratifs avec la suite *Google Apps*.

Emma Verneuil:

- Création du GIF
- Création du «Virtual Drum»
- Création du bouton *ON/OFF* de la musique de fond (modifié par Hélène Carignan pour alléger le projet)
- Création de l'interface de base (lien *URL*, animation du texte, et textes fixes sauf les textes des bulles)
- Création du bouton *Restart* (redémarre les bulles, l'animation du texte et la trame sonore, modifié et corrigé par Hélène Carignan)
- Intégration du «Virtual Drum» dans une bulle et création de la légende
- Intégration du code du «Virtual Drum» dans le fichier principal et création d'une classe *Drawdrum*
- Compilation du code de l'interface de base, celui des bulles et celui du bouton *ON/OFF* ensemble pour ne former qu'un seul grand code (revu et corrigé par Hélène Carignan en raison des lourdeurs et des bugs de tous les codes réunis)
- Rédaction des ressources utilisées dans le cadre du projet et aide à la rédaction générale (ajout d'éléments précis sur les codes et leurs actions)

Sandie Valiquette :

- Rédaction et réalisation du document de design
- Communication avec membres pour la création et la validation des informations pour le document de présentation (afin que chacun puisse donner les difficultés rencontrées, et expliquer l'action générée par leur code)
- Soutien moral pour les trois «programmeurs»