

BEAULIEU, William;

BÉDARD, Hugo;

JULLIEN, Jérémie;

TANGUAY, Thomas.

Programmation en animation

ANI-2012

Travail pratique #1 :

Document de design

Travail présenté à

VOYER, Philippe

Université Laval

Automne 2017

Table des matières

2.1 Sommaire-----	p. 2
2.2 Interactivité-----	p. 3
2.3 Fonctionnalités-----	p. 4
- 2.3.1 Interactivité-----	p. 4
- 2.3.2 Graphe de scène-----	p. 4
- 2.3.3 Texte-----	p. 5
- 2.3.4 Dessin vectoriel-----	p. 5
- 2.3.5 Séquence d'image-----	p. 6
- 2.3.6 Audio-----	p. 6
- 2.3.7 Vidéo-----	p. 7
- 2.3.8 Musique-----	p.
- 2.3.9 Système de particules-----	p.
- 2.3.10 Système dynamique-----	p.
2.4 Ressources-----	p.
2.5 Présentation-----	p.

2.1 Sommaire

Dans ce travail pratique, nous avons développé par programmation, à l'aide de la technologie Processing, une animation interactive en 2 dimensions. L'animation que nous avons créée est quelque peu semblable à un jeu-vidéo, c'est-à-dire que l'utilisateur sera amené à participer durant l'exécution du programme. Le but de l'animation est donc, de pousser l'utilisateur à faire certains choix en interagissant avec les différentes fonctionnalités qui lui sont proposés du début à la fin et non d'exécuter le programme et de n'être simplement qu'un spectateur. Différentes possibilités sont offertes à l'utilisateur pour cheminer à travers l'animation cependant, il n'y a qu'un seul commencement et une seule alternative à la fin. Le travail que nous présentons reflète bien les différentes notions qui ont été enseignées dans les modules qui ont été explorés durant la première partie du cours de Programmation en animation. Les notions apprises sont toutes intégrées dans l'animation sous différentes formes et dans certains cas, qui utilisent des fichiers provenant de ressources externes au logiciel.

2.2 Interactivité

L'interactivité correspond à une partie extrêmement importante au sein de l'animation que nous avons développée. Elle constitue principalement le lien entre l'utilisateur et les différentes instances implémentées dans le programme. L'interactivité correspond entre autres, au premier critère fonctionnel qui était nécessaire au fonctionnement de ce projet, il était donc celui à laquelle nous avons apporté une attention plus particulière puisqu'il était toujours question de celui-ci au moment de définir la façon qui allait être utilisée pour présenter les autres critères fonctionnels. En pratique, l'interactivité correspond donc à certaines manipulations effectuées, pour la plupart, par l'utilisateur qui entraîne des résultats (action-réaction). Plusieurs touches du clavier servent à engendrer une instance différente lorsqu'elles sont relâchées. Lorsque la touche clavier «b» est pressée, deux résultats sont causés soit, la composante audio du robinet ainsi que la composante visuelle du robinet qui correspond au système de particules. Ensuite, les touches « z,x et c » sont utilisées pour ouvrir chacun des coffres en plus d'être accompagné d'une composante audio à chaque ouverture. Pour la télévision, il y a deux touches qui ont des actions en lien avec celle-ci mais qui agissent différemment. La touche «v» par exemple, active la lecture du fichier vidéo à l'intérieur du cadre de la télévision, tandis que la touche «m» active le système dynamique ainsi que la composante audio qui représentent l'effet statique dans une télévision. De plus, le clic de la souris est une autre interactivité qui sera utilisée afin d'afficher le texte en début de jeu. Une touche du clavier très importante est le «d» puisqu'il sert à afficher le début du jeu, il est donc indispensable de l'utiliser afin de profiter du reste de l'animation.

2.3 Fonctionnalités

2.3.1 Interactivité

Pour le critère d'interactivité, nous avons utilisé deux contrôleurs interactifs, soit le clavier et la souris. Le clic de la souris est une action liée au texte. Pour le clavier, nous avons utilisé les touches « b,v,m,z,x,c et d » qui correspondent toutes à des actions bien définies dans le programme. Voici en exemple les codes pour les touches « d et z » :

```
void keyPressed() {
  if (key == 'd' || key == 'D') {
    BG();
  }

  while (key == 'z' || key == 'Z') {
    if (player2.isPlaying()) player2.pause();
    else player2.play(0);
    strokeWeight(2);
    fill(51, 51, 0);
    rect( 295, 675, 70, 50 );
    fill(0);
    text("Il n'y a rien ici, appuyez sur 'X' pour ouvrir le deuxième coffre. ", 100, 660);
    break;
  }
}
```

2.3.2 Graphe de scène

Le graphe de scène est un critère particulièrement en lien avec les coffres puisque ce sont des composantes qui sont mutables au cours de l'animation. Lors de l'ouverture des coffres, l'aspect physique du coffre va passer d'un coffre fermé à un coffre ouvert.

```
while (key == 'x' || key == 'X') {
  BG();
  if (player2.isPlaying()) player2.pause();
  else player2.play(0);
  strokeWeight(2);
  fill(51, 51, 0);
  rect(465, 675, 70, 50 );
  fill(0);
  text("Il n'y a rien ici non plus, appuyer sur 'C' pour ouvrir le dernier coffre. ", 75, 660);
  break;
}

while (key == 'c' || key == 'C') {
  BG();
  if (player2.isPlaying()) player2.pause();
  else player2.play(0);
  strokeWeight(2);
  fill(51, 51, 0);
  rect(635, 675, 70, 50 );
  fill(0);
  text("Vous avez trouvé! Appuyer sur 'V' pour jouer la vidéo ", 160, 628);
  text("et sur 'B' pour ouvrir le gicleur", 300, 670);
  break;
}
```

2.3.3 Texte

Pour répondre à ce critère, le programme comporte de nombreux éléments textuels fixes, tel un message de départ à la suite de la séquence d'image, les explications du déroulement du programme, ainsi que l'écriture des touches à peser pour interagir avec l'animation. De plus, il y a aussi un texte animé, qui apparaît à la toute fin pour annoncer au participant qu'il a complété toutes les tâches. On peut observer que le message s'écrit lettre par lettre.

```
void mousePressed() {  
  fill(0);  
  rect(50, 50, 900, 900);  
  textFont(Font, 32);  
  fill(255);  
  text("Bienvenue dans notre petit jeu interactif animé. ", 220, 400);  
  text("Pour faire progresser l'interactivité,", 290, 450);  
  text("il suffit d'appuyer sur les touches affichées", 240, 500);  
  text("en dessous de chaque éléments.", 310, 550);  
  text("Appuyez sur \"D\" pour débiter", 360, 625);  
}
```

2.3.4 Dessin vectoriel

Le décor qui a été créé pour le programme comporte de nombreuses formes vectorielles, plus précisément de 5 formes différentes. On retrouve plusieurs rectangles, quelques triangles, des lignes, des ellipses, ainsi qu'un arc. Ils servent à représenter des objets réels, comme des coffres, une télévision, et un gicleur à incendie. Voici quelques exemple de formes qui ont été implantées :

```
fill( 170 );  
rect( 490, 0, 20, 35 );  
rect( 497, 35, 6, 10 );  
rect( 485, 45, 30, 5 );  
line( 500, 45, 500, 50 );  
line( 493, 45, 493, 50 );  
line( 507, 45, 507, 50 );  
  
fill(150);  
arc(500, 552, 150, 100, 0, PI);  
  
noFill();  
strokeWeight(4);  
ellipse(330, 825, 60, 60);  
  
strokeWeight(4);  
line(320, 815, 340, 815);  
line(340, 815, 320, 835);  
line(320, 835, 340, 835);  
}
```

2.3.5 Séquence d'images

Pour ce qui est de la séquence d'image, on la retrouve au tout début de l'animation. Elle sert à introduire le décor où se retrouve toute les différents éléments d'interactivité, comme le vidéo, les fichiers audio, le système dynamique, etc. Effectivement, cette séquence en noir et blanc montre toutes les étapes de création du décor, puisque les formes géométriques sont ajoutés les unes après les autres, dans un ordre logique de création.

```
void draw() {  
  
  if (Sequence) {  
    frame = (frame+1)%numFrames;  
    image(images[frame], 0, 0);  
    if (frame == 7) {  
      Sequence = false;  
      textFont(Font, 32);  
      text("Cliquez pour continuer", 365, 400);  
    }  
  }  
}
```

2.3.6 Audio

En vue de répondre aux caractéristiques de ce critère fonctionnel, l'idée de base a été d'implémenter des effets sonores aux diverses animations de notre programme. Effectivement, nos trois fichiers audios viennent compléter les différentes autres fonctions, tel le système de particules, le système dynamique, ainsi que le dessin vectoriel. Un des problèmes qui a surgi lors de l'écriture de cette fonction a été la superposition des fichiers audios, mais voici comment cela a pu être réglé :

```
player1 = minim.loadFile("SprinklerSound.mp3");  
player2 = minim.loadFile("ChestOpening.wav");  
player3 = minim.loadFile("TVStatic.aiff");
```

(...)

```
while (key == 'c' || key == 'C') {  
  BG();  
  if (player2.isPlaying()) player2.pause();  
  else player2.play(0);  
}
```

2.3.7 Vidéo

Le critère fonctionnel de vidéo est implémenté dans notre animation en étant cadrer à l'intérieur de certaines limites (la télévision). Il est lié à une touche clavier, soit la touche «v» qui enclenche la lecture du vidéo qui a été importée au début du code dans Processing dans un format mp4.

```
void movieEvent(Movie video) {  
    video.read();  
}
```

(...)

```
if ((key == 'v' || key == 'V') && coffre3 == false && tv == true) {  
    frameRate(30);  
    video.play();  
    tv = false;  
}
```

2.3.8 Musique

La musique est implémentée dans l'animation sous forme de séquence de notes qui sont synchronisées avec le texte à la fin. Il sera déclenché au moment où le participant aura fini d'interagir avec tous les éléments de l'animation.

```
if (coffre1 == false && coffre2 == false && coffre3 == false && tv == false && stat == false && gicleur == false){  
    frameRate(5);  
    textSize(150);  
    fill(255);  
    howManyLettersToShow();  
    typeIt(test, 160, 158);  
    while (play == true){  
        out.playNote( 0.0, 0.1, "Ab" );  
        out.playNote( 0.2, 0.1, "Ab" );  
        out.playNote( 0.4, 0.1, "Ab" );  
        out.playNote( 1.0, 1.5, "A#" );  
        out.playNote( 1.0, 1.5, "D#" );  
        play = false;  
    }  
}
```


2.3.9 Système de particules

Le système de particule qui a été utilisé dans cette animation vient découle d'un exemple du site processing.org. Il a été utilisé pour créer l'effet d'un de nos interactivités, puisqu'il simule l'eau qui s'échappe du gicleur.

```
ps = new ParticleSystem(new PVector(width/2, 50));

if(spr == true && coffre3 == true){
  BG();
  frameRate(60);
  ps.addParticle();
  ps.run();
  fill(0);
  text("Pas de chance, encore rien. Appuyer sur 'C' pour ouvrir le troisième coffre. ", 50, 660);
  gicleur = false;
```

2.3.10 Système dynamique

Le système dynamique est une notion qui a été apprise lors de la semaine 4 du cours. Le système dynamique implémenté dans l'animation présente correspond à un automate cellulaire qui agit sensiblement comme l'effet statique dans une télévision. Le code a été réalisé sensiblement de la même façon que dans l'exemple du cours, c'est-à-dire sous formes de classe mais la partie la plus importante est dans l'intégration de cette instance au reste du programme.

```
ca = new Automaton(Automaton.TYPE_CELL_NOISE, ceil(520 / resolution), ceil(325 / resolution), resolution, frequency, alpha);
}

if (mo == false){
  player3.play();
  frameRate(60);
  translate(240, 212);
  ca.update();
  ca.update();
  ca.update();
  ca.update();
  stat = false;
}
```

2.4 Ressources

Sources audios :

- Coffre : YouTube. (22 juillet 2016). Chest Opening SOUND Effect [Fichier Vidéo].
Récupéré de <https://www.youtube.com/watch?v=mMBHFswOBik>
- Sprinkler : Youtube. (14 août 2013). Sprinkler Sounds Effects [Fichier Vidéo].
Récupéré de https://www.youtube.com/watch?v=C1K6W_FkH8Q
- Bruit : Youtube. (23 décembre 2016). TV Static Sound Effect- Download [Fichier Vidéo]. Récupéré de <https://www.youtube.com/watch?v=USdqmhBnWkk>

Source vidéo :

- Youtube. (29 novembre 2016). WhatsApp funny video 2017 [Fichier Vidéo].
Récupéré de <https://www.youtube.com/watch?v=SEavdHi5x2U>

Source des codes :

- <https://processing.org>
- <https://github.com/philvoyer/ANI2012A17>

2.5 Présentation

Équipe 1:

Jérémie Jullien:

Diplômé d'une technique en graphisme au cégep de Cégep de Sainte-Foy. Je me suis inscrit dans le CASA afin d'acquérir le plus de bases en animations possible pour me lancer dans le BASA l'an prochain. Je suis passionné d'art de toute sorte depuis toujours, mais j'ai eu la piqure pour l'animation durant ma technique au cégep. Le cours de programmation est un grand début en programmation et en animation par programmation pour moi. Toutefois je trouve toujours appréciable et bénéfique d'apprendre de nouvelles compétences, ce qui rend le cours plutôt intéressant pour moi.

Hugo Bédard:

Je suis diplômé en Sciences de la nature au Cégep de Mont-Tremblant. Je me suis inscrit au certificat en art et science de l'animation après avoir fait 1 an et demi dans le baccalauréat en sciences géomatiques à l'Université Laval. Le baccalauréat en sciences géomatiques m'a permis de toucher pour la première fois à la programmation avec le logiciel MatLab. Par contre, ce cours est pour moi, la première expérience de programmation dans le domaine de l'animation. Les films d'animations et les jeux-vidéos m'ont toujours intéressé, c'est pourquoi je me suis dirigé dans ce domaine en ayant comme objectif d'apprendre le plus possible et de m'améliorer, surtout dans la programmation.

Thomas Tanguay:

J'ai fait mes études collégiales en Sciences, Lettres et Arts au cégep de Jonquière. Ayant de grands intérêts pour les films d'animation, ainsi que les arts en général, comme le cirque, la musique et les films, je me suis lancé en Automne 2017 dans le CASA, certificat en art et science de l'animation. La programmation est une matière tout à fait nouvelle pour moi, mais ça reste néanmoins une matière très intéressante. C'est donc pour moi un grand défi à relever.

William Beaulieu:

Diplômé d'un DEC sans mention, je suis étudiant au Bac en informatique. Je suis inscrit au cours de programmation en animation parce que les jeux vidéos me passionnent et que ce cours est un bon cours complémentaire a mon Bac en informatique puisque je veux continuer vers le jeu vidéo ensuite. J'ai déjà suivi un cours de programmation Java au Cégep auparavant, mais l'animation est toute nouvelle pour moi. J'ai donc grandement à apprendre dans ce domaine de programmation.