

### Synthèse de l'article

#### « *Interactions and systems for augmenting a live dance performance* »

d'Alexis Clay, Nadine Couture, Laurence Nigay, Jean-Baptiste de la Rivière, Jean-Claude Martin, Matthieu Courgeon, Myriam Desainte-Catherine, Emmanuel Orvain, Vincent Girondel et Gaël Domenger

L'article étudié présente le travail effectué au sein du projet CARE (*Cultural Experience: Augmented Reality and Emotion*), dont l'objectif est de développer, adapter et intégrer des outils liés à la réalité augmentée, dans le but d'améliorer l'émotion produite par une performance culturelle. Le projet s'est achevé sur un spectacle de danse augmentée, mêlant reconnaissance de mouvements et d'émotions, réalité augmentée et génération de fond musical. Afin de mieux comprendre le sujet, rappelons qu'une danse augmentée est une forme de danse qui réinvente la notion de mouvement, en insérant en temps réel un ou plusieurs éléments numériques. Les auteurs de l'article présentent plus particulièrement le travail réalisé autour de ce spectacle, ainsi que les résultats de recherche du projet obtenus. Cette synthèse commencera ainsi par un résumé de l'article, puis s'y intéressera d'un point de vue « adaptation des IHM ».

Le sujet concernant l'augmentation d'un spectacle de danse, un objectif applicatif apparaît clairement : celui d'augmenter la performance d'un danseur en ajoutant des éléments audiovisuels virtuels, et fournir aux interprètes de nouveaux moyens d'interagir avec ces éléments.

Différentes techniques d'interaction ont donc été développées et intégrées à des outils et techniques de réalité augmentée, dans le but de créer un système complet, pour un spectacle de danse augmentée et interactif.

Ce système consiste en un assemblage de cinq modules, permettant de créer différents types d'interactions et augmentations :

- La capture de mouvement (*Motion tracking*) : elle permet de connaître la conformation exacte du corps du danseur, ainsi que sa position sur la scène en temps réel.
- La reconnaissance d'émotions : elle permet d'identifier des émotions à partir de l'analyse des mouvements du danseur.
- La génération de sons et musiques en adéquation avec une émotion donnée.
- La génération d'un avatar correspondant à un robot humanoïde, effectuant des mouvements en adéquation avec une émotion donnée.
- La génération d'éléments virtuels ainsi que l'infrastructure physique et logicielle permettant l'affichage de ces augmentations.

Les différentes interactions développées à partir de ce système sont basées sur les mouvements. On distingue des interactions directes, telles que la manipulation et la génération d'éléments virtuels à partir des mouvements du danseur, ainsi que des interactions indirectes, possédant un haut niveau d'abstraction. Ces dernières correspondent à la traduction des mouvements du danseur en émotions et à la génération d'éléments audiovisuels en adéquation avec les émotions reconnues.

Actuellement, avec l'évolution constante du nombre d'utilisateurs, de plateformes et d'environnements, il devient difficile de créer une interface pour chaque situation. C'est pourquoi il est devenu essentiel de rendre ces interfaces plastiques, c'est-à-dire capables de s'adapter à un changement de contexte d'usage, dans le respect de leur utilisabilité. Il est intéressant de se demander

quel est le besoin en plasticité dans le cas présent. Pour cela, nous essaierons de répondre à différents critères de positionnement qui s'appliquent ici. Nous commencerons par identifier le contexte d'usage, déterminer à quel moment l'adaptation de l'interface s'effectue, proposer une solution permettant de répondre au besoin en plasticité, puis analyser cette solution.

Dans notre cas, le contexte d'usage est surtout centrée utilisateur. En effet, il s'agit ici d'un spectacle de danse augmentée dont les interactions sont basées sur les mouvements du danseur. La plateforme est constituée de deux supports, le mur du fond recouvert d'un écran blanc et un revêtement au sol blanc, ainsi que de deux dispositifs correspondant à deux vidéoprojecteurs, projetant à partir d'un ordinateur, sur l'écran du fond et sur le sol de la scène. Il s'agit plus précisément d'un système de projection stéréoscopique, visant à reproduire une perception du relief à partir de deux images planes, perceptible par les spectateurs grâce au port de lunettes. L'environnement correspond quant à lui à la scène de spectacle.

Comme il a été annoncé précédemment, les interactions sont basées sur les mouvements du danseur. L'interface étant alors adaptable et dynamique, les modalités et leurs usages sont choisis par l'utilisateur et ces choix ont lieu à l'exécution.

Le système existant consiste plus particulièrement en l'assemblage des cinq modules suivants :

- La capture de mouvement (*Motion tracking*) : elle permet d'extraire l'information de mouvement du danseur et d'enregistrer la position exacte de celui-ci et de chacun des membres de son corps sur la scène, grâce au port d'un costume MVN de la société XSens. Il s'agit d'une combinaison en Lycra équipée de 17 capteurs de mouvements à inertie, capable de communiquer le moindre déplacement du danseur à un ordinateur. Celui-ci traite les mouvements pour en contrôler une contrepartie virtuelle (modèles 3D, avatar), et afficher la restitution visuelle de ces mouvements en temps réel.
- La reconnaissance d'émotions à partir de l'analyse des mouvements du danseur : elle permet d'extraire certaines caractéristiques des mouvements fournis par le module de « capture de mouvement » et d'y associer des émotions. Leur système de reconnaissance d'émotions nommé *eMotion* a été construit autour de 6 émotions universelles : le bonheur, la tristesse, la peur, la colère, la surprise et le dégoût.
- La génération de sons et musiques en adéquation avec une émotion donnée. Ce système nommé MuZICO, reçoit une émotion envoyée par le module de « reconnaissance d'émotions », l'évalue en fonction de deux axes : sa valence (négative, positive) et son intensité (ampleur de l'émotion), puis la traduit en coordonnées dans cet espace à deux dimensions. A partir de ces coordonnées, le système génère ensuite des compositions musicales, le tout en temps réel. La valence étant traduite en un ensemble d'échelles musicales et l'intensité en rythmique.
- La génération d'un avatar correspondant à un robot humanoïde, effectuant des mouvements en adéquation avec une émotion donnée. Cette génération se fait grâce au logiciel 3DVIA Virtools, qui permet la création d'applications 3D en temps réel, pour des environnements immersifs.
- La génération d'éléments virtuels ainsi que l'infrastructure physique et logicielle permettant l'affichage de ces augmentations. De même que précédemment, cette génération est rendue possible grâce à l'utilisation de l'outil 3DVIA Virtools.

L'assemblage de ces modules a permis le développement de différentes interactions basées sur les mouvements :

- 3 interactions directes
  - La manipulation directe des éléments virtuels par le danseur
  - L'animation d'un avatar qui copie les mouvements du danseur
  - La génération de rubans 3D à partir des mouvements des mains du danseur
- 2 interactions indirectes possédant un haut niveau d'abstraction
  - La traduction des mouvements du danseur en émotions
  - La génération de « musiques émotionnelles » et de « mouvements émotionnels » pour l'avatar

Il est maintenant intéressant de se demander comment cette solution réagirait à un changement de contexte. Nous proposons de faire varier un des 3 points qui composent le contexte d'usage (plateforme/environnement/utilisateur) et de montrer ce qu'il faudrait modifier afin de répondre au besoin de plasticité. Nous pouvons nous demander s'il faudrait changer les techniques d'interactions et comment ?

Pour commencer, il serait intéressant d'adopter une « démarche générique » afin d'automatiser le changement. Pour ce faire, il faudrait monter dans le niveau d'abstraction et identifier un modèle applicable sur n'importe quel support.

Dans le cadre d'un spectacle de danse augmentée dont les interactions sont basées sur les mouvements du danseur, nous adopterons une démarche utilisateur. Nous partons donc du modèle de tâches, qui exprime explicitement les objectifs, et qui s'appuiera ensuite sur les outils existants afin d'atteindre ces objectifs, quelle que soit l'évolution du système.

Puisque la modélisation se fait par rapport à un objectif, il est essentiel d'identifier celui-ci. De plus, le modèle identifié doit être capable de répondre aux problématiques du système actuel. Il est donc nécessaire d'identifier les critères permettant cela. Rappelons l'objectif principal du projet, qui est d'augmenter un spectacle de danse en ajoutant des éléments audiovisuels virtuels à la scène, et de fournir aux artistes de nouveaux moyens d'interagir avec ces éléments.

Afin de mieux déterminer les critères sur lesquels nous nous baserons pour constituer un modèle, déclinons l'objectif principal en termes de fonctionnalités :

- Reconnaissance de mouvements en temps réel
- Reconnaissance d'émotions à partir de l'analyse de ces mouvements en temps réel
- Génération de sons et musiques en adéquation avec une émotion donnée en temps réel
- Génération de mouvements en adéquation avec une émotion donnée pour un avatar en temps réel
- Génération d'éléments virtuels en temps réel

Définissons maintenant l'objectif en termes de critères. Nous pouvons identifier dans le cas présent, que l'interaction qui est de type gestuelle, correspond aux mouvements du danseur et que la technique d'interaction utilisée correspond à des capteurs de mouvements, disposés sur une combinaison portée par le danseur. Il s'agira donc ici de modéliser la capacité des capteurs par rapport aux fonctionnalités à effectuer.

Nous pouvons ainsi établir un modèle de correspondance des données vers les représentations visuelles. En effet, qu'importe la manière dont les données sont acquises par le système (ici, les

mouvements du danseur), ce dernier dispose de données et de « vues », puis à partir des tâches et interactions effectuées par le danseur, le système traite ces données et affiche les générations audiovisuelles virtuelles sur les vues.

Une fois le modèle identifié, nous pouvons nous intéresser aux autres techniques d'interactions existantes et trouver une technique d'interaction utilisant le même modèle. Cela permettant de vérifier la validité du modèle.

Actuellement, la technique d'interaction utilisée correspond à des capteurs de mouvements à inertie, placés sur une combinaison portée par le danseur. Une multitude de capteurs différents existent aujourd'hui et évoluent constamment. Nous choisirons d'appliquer une technique d'interaction différente, celle des capteurs infrarouges.

Nous nous proposons d'analyser cette nouvelle solution en indiquant ses avantages et inconvénients. Tout d'abord, l'utilisation de capteurs infrarouges impose quelques modifications au niveau des outils utilisés. En effet, il s'agirait de se concentrer sur l'éclairage de la scène, en utilisant une lumière infrarouge proche du spectre visible. Il faudrait également utiliser une caméra dotée d'un capteur, qui alimenterait un système d'analyse d'images, détectant la gestuelle, la position, la vitesse, ainsi que l'accélération des danseurs. La partie du système concernant le traitement de ces informations et la projection vidéo de ces générations resterait inchangée.

La solution initiale possède un avantage par rapport aux méthodes classiques, puisqu'elle permet une souplesse d'utilisation plus grande. En effet, aucune caméra n'est nécessaire pour saisir la performance du danseur, puisque cela est rendu possible grâce au costume équipé de capteurs.

Cependant, le port d'une combinaison n'est pas forcément pratique lors d'une performance, et peut influencer visuellement, l'émotion produite par cette performance.

La nouvelle solution permettrait ainsi d'éviter le port d'une combinaison, qui peut être contraignante. De plus, elle offrirait la possibilité d'une performance composée de plusieurs danseurs, sans avoir à se procurer autant de costumes équipés de capteurs, que de danseurs. Dans ce cas, cela permettrait de réduire grandement le coût de mise en œuvre, car ces costumes coûtent chers.

Cependant, étant donné que la solution trouvée repose sur l'éclairage, l'utilisation de cette technologie imposerait de nouvelles contraintes au danseur. En effet, certaines zones de la scène ne pourront pas être utilisées car la projection ne sera plus possible.

## Etude technologique sur le *Responsive Web Design* Bootstrap VS Foundation

Cette étude comparative portant sur le *Responsive Web Design* (RWD), rappelons qu'il s'agit d'une approche de conception web, ayant pour but d'élaborer des sites web adaptatifs offrant une expérience de lecture et de navigation optimales pour l'utilisateur, quel que soit le support (ordinateurs, smartphones, tablettes, télévisions). Afin que cette expérience utilisateur soit réussie, un certain nombre de critères doivent être satisfaits. A titre d'exemple, une expérience réussie implique un minimum de redimensionnement, de recadrage et de défilements multidirectionnels de pages.

Les critères liés à la notion de plasticité seront présentés dans la suite du document, lors d'une étude comparative des *frameworks* CSS Bootstrap et Foundation. Avant de commencer cette comparaison, nous commencerons par nous positionner par rapport à la plasticité dans le cadre des solutions *Responsive Web Design*, puis nous définirons la notion de *framework* CSS avant de présenter Bootstrap et Foundation. Nous terminerons par une étude comparative de ces deux solutions.

Identifions tout d'abord le contexte d'usage. Ici, le type d'utilisateur n'a pas d'influence. Cependant, selon les actions de l'utilisateur, celui-ci peut par exemple décider d'orienter son smartphone en mode portrait ou paysage. De ce fait, il y a un besoin d'adaptation des contenus selon l'orientation du dispositif. Un autre exemple concerne un utilisateur qui redimensionne la fenêtre de son navigateur. On observe également un besoin d'adaptation des contenus selon la largeur du dispositif. Concernant la plateforme, l'un des objectifs du RWD est de permettre le développement de sites web multiplateformes. En effet, selon la plateforme, la largeur de l'écran n'est pas la même et le RWD permet l'adaptation des contenus selon la largeur du dispositif.

Dans le cas du RWD, l'adaptation s'effectue à la fois à la conception et à la fois à l'exécution. En effet, certains éléments adaptables sont d'abord conçus, avant de pouvoir être fournis aux développeurs, ce qui leur facilite le travail. Cependant, l'adaptation peut également s'effectuer à l'exécution, de par les actions réalisées par l'utilisateur.

Avant de poursuivre sur l'étude de Bootstrap et Foundation, il est nécessaire de définir la notion de *framework* CSS. Un *framework* en général constitue un ensemble de bibliothèques et d'outils offrant une aide à la programmation. Comme son nom l'indique, un *framework* CSS utilise et simplifie la mise en place de la charte graphique d'un site web grâce au CSS. Le principe de ces *frameworks* consiste en des classes CSS prédéfinies dans un fichier CSS. Ces classes permettent une mise en page rapide, grâce à un système de grilles. Elles permettent également l'intégration simple de différents composants tels que des boutons, des barres de navigation, des formulaires ou encore des menus. Il existe actuellement de très nombreux *frameworks* de présentation de contenu web. Nous proposons d'étudier ici la version 3 de Twitter Bootstrap et la version 5 de ZURB Foundation.

- Twitter Bootstrap [1] est le *framework* CSS le plus populaire, utilisé pour construire des sites et des applications web. La version 3 de Bootstrap fonctionne sur un système de grille à 12 colonnes, avec 4 types d'affichages possibles selon la largeur de l'écran de l'utilisateur. Bootstrap 3 est totalement *responsive*. En effet, les éléments s'adaptent à la largeur de l'écran en glissant les uns en dessous des autres lorsque la largeur diminue. L'adaptation se fait également par un affichage différent selon l'écran comme énoncé précédemment. Bootstrap 3 propose des composants de base tels que des boutons, des formulaires, des barres de navigation et plus encore. Il propose également l'utilisation du préprocesseur LESS, permettant de faciliter la personnalisation du CSS. Le site de Bootstrap propose également de

nombreux exemples de *templates* directement utilisables, ainsi que des composants jQuery tels que des menus déroulants, des fenêtres modales etc.

- Le principe reste le même pour ZURB Foundation [2]. Il fournit des classes CSS, des composants etc. Cependant, celui-ci fournit un système de grilles allant jusqu'à 16 colonnes (12 par défaut). L'adaptation à la taille de l'écran est cependant différente. En effet, l'adaptation se fait non seulement avec des blocs, mais également avec des listes. De plus, Foundation fonctionne avec le préprocesseur SASS.

La mise en place du *Responsive Web Design* nécessite en général les technologies et méthodes suivantes [3] :

- Une grille de mise en page fluide (s'adaptant notamment à la largeur du dispositif).
- Des images, des médias et des contenus flexibles.
- Une adaptation de l'affichage au *viewport* du terminal. Rappelons que le *viewport* correspond à la surface d'affichage du navigateur web. La notion de *viewport* d'un terminal mobile est différente de celle d'un écran d'ordinateur. Sur mobile, le navigateur ne dispose pas de fenêtre réelle, ni de barre de défilement.
- Des *media queries* (qui sont des spécifications de CSS3), permettant d'appliquer différentes règles de styles CSS selon la largeur de l'écran, du type de support, de l'orientation (portrait ou paysage) ainsi que de la densité de pixels.
- Des plugins JavaScript ou jQuery responsive et compatibles.
- Une approche « *Mobile First* » facilitant l'accessibilité, la compatibilité et la performance des pages produites. Le principe de cette approche consiste à partir des écrans les plus petits (smartphones) pour aller vers de plus grands écrans (ordinateurs) dans la conception.

L'étude comparative se concentrera donc principalement sur ces critères [4][5].

	Bootstrap 3	Foundation 5
Grille	<ul style="list-style-type: none"> <li>• Fluide</li> <li>• Non responsive : 960 px</li> <li>• Plusieurs points d'arrêts responsive               <ul style="list-style-type: none"> <li>○ mobile &lt; 768px</li> <li>○ tablette &gt; 768px</li> <li>○ écran &gt; 992px / &gt;1200 px</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Fluide</li> <li>• Un point d'arrêt responsive : 768px</li> <li>• Création d'une grille qui inclut 2 points d'arrêts et 3 grilles (petit, moyen, grand)</li> </ul>
CSS	Similaire : navigation, fils d'Ariane ( <i>breadcrumbs</i> ), typographie, tables, formulaires, boutons, classes de visibilité (affichage ou non d'éléments selon la taille de l'écran, l'orientation, le support du tactile par le dispositif), pagination, alertes, barres de progression, vignettes	
	Images, badges, listes avec un contenu personnalisable ( <i>list groups</i> )	Grilles de blocs ( <i>block grid</i> ), vidéos flexibles ( <i>flex video</i> ), listes horizontales ( <i>inline lists</i> ), tableaux de prix ( <i>pricing tables</i> )
Préprocesseur CSS	LESS, SASS	SASS
JavaScript	jQuery	Zepto/jQuery
Composants JavaScript	Similaire : fenêtres modales, menus déroulants, <i>pop-overs</i> , barres de défilement, sections/onglets/accordéons, bulles d'informations ( <i>tooltips</i> ), fenêtres modales responsive affichant des images ( <i>lightbox</i> )	

		<ul style="list-style-type: none"> <li>• Interchange (images adaptives)</li> <li>• Joyride (visite guidée du site)</li> <li>• Abide (validation de formulaire)</li> </ul>
<b>Navigateurs supportés</b>	<ul style="list-style-type: none"> <li>• Chrome (Mac, Windows, iOS et Android)</li> <li>• Safari (Mac, iOS)</li> <li>• Firefox (Mac, Windows)</li> <li>• Opera (Mac, Windows)</li> <li>• IE8+</li> </ul>	<ul style="list-style-type: none"> <li>• Chrome</li> <li>• Firefox</li> <li>• Safari</li> <li>• IE9+</li> </ul> Mobile: <ul style="list-style-type: none"> <li>• iOS (iPhone)</li> <li>• iOS (iPad)</li> <li>• Android 2, 4 (Phone, Tablet)</li> <li>• Windows Phone 7+</li> <li>• Surface</li> </ul>
<b>Avantages</b>	<ul style="list-style-type: none"> <li>• Large communauté et support</li> <li>• Beaucoup de plugins, d'extensions et de styles</li> <li>• Activement développé et supporté</li> <li>• Support d'IE8</li> </ul>	<ul style="list-style-type: none"> <li>• Allégé, styles de base plus neutres</li> <li>• Permet une personnalisation du CSS plus simple</li> <li>• Documentation complète</li> </ul>
<b>Inconvénients</b>	<ul style="list-style-type: none"> <li>• Personnalisation du CSS compliquée</li> </ul>	<ul style="list-style-type: none"> <li>• Ne supporte pas IE8</li> </ul>

## Références

[1] Twitter Bootstrap. Available at <http://getbootstrap.com>

[2] ZURB Foundation. Available at <http://foundation.zurb.com>

[3] Marcotte, E. (2011). Responsive web design. Editions Eyrolles.

[4] Responsive CSS Framework Comparison. Available at <http://responsive.vermilion.com>

[5] Bootstrap vs. Foundation: Which Framework is Better? Available at <http://bootstrapbay.com>