Interaction Homme-Machine

Projet de réalité augmentée

Objectifs

- identification de tâches d'interaction et création de séquences d'interaction
- mise en œuvre de critères ergonomiques
- mise en œuvre de prototypage rapide
- conception et mise en œuvre d'interfaces avec JavaFX

1 Préambule

« La réalité augmentée est la superposition de la réalité et d'éléments (sons, images 2D, 3D, vidéos, etc.) calculés par un système informatique en temps réel. Souvent, elle désigne les différentes méthodes qui permettent d'incruster de façon réaliste des objets virtuels dans une séquence d'images. Elle s'applique aussi bien à la perception visuelle (superposition d'image virtuelle aux images réelles) qu'aux perceptions proprioceptives comme les perceptions tactiles ou auditives. Ces applications sont multiples et touchent de plus en plus de domaines, tels que les jeux vidéo, l'éducation par le jeu, les chasses au trésor virtuelles, le cinéma et la télévision (post-production, studios virtuels, retransmissions sportives...), les industries (conception, design, maintenance, assemblage, pilotage, robotique et télérobotique, implantation, étude d'impact...etc.) ou le champ médical. » [Wikipédia [1]]

Pour ce projet nous utiliserons la réalité augmentée pour incruster de manière réaliste (respect des perspectives) des objets virtuels dans des images réelles.

2 Contexte et problématique

Ce projet s'inscrit dans le contexte de l'aménagement et de la rénovation de l'habitat. Il est difficile pour une majorité de personnes de se représenter mentalement le résultat de travaux avant leur réalisation. L'objectif est de développer une interface permettant de pré-visualiser en réalité augmentée le résultat de travaux. L'objectif est de permettre à l'utilisateur de faire des choix avisés quand différentes alternatives pour un projet de travaux s'offrent à lui.

Le projet se limitera à l'incrustation d'objets de formes rectangulaires (fenêtres, portes, tableaux...) dans des photos, par utilisation de la projection perspective qui permet d'associer un quadrilatère quelconque à un autre tout en s'assurant que les droites avant transformation restent représentées par des droites après transformation.

3 Base de code JavaFX

JavaFX met à votre disposition la classe PerspectiveTransform [2] afin de vous éviter le calcul de matrices de transformation. Un exemple minimal de code est disponible ci-dessous qui vous illustre comment incruster une fenêtre virtuelle à la place d'une fenêtre réelle, tout en respectant les perspectives. Les coordonnées renseignées dans l'image réelle ont été obtenues en déterminant manuellement les coordonnées des quatre coins de la fenêtre réelle.

```
public class HelloPerspective extends Application {

  public void start(Stage stage) {
    VBox root = new VBox();
    Canvas canvas = new Canvas (500, 504);
    GraphicsContext gc = canvas.getGraphicsContext2D();
```

1. https://fr.wikipedia.org/wiki/Réalité_augmenté
2. https://docs.oracle.com/javafx/2/api/javafx/scene/effect/PerspectiveTransform.html

```
    Image imageReelle = new Image("resources/imageReelle.png");
    Image imageVirtuelle = new Image("resources/imageVirtuelle.png");

    PerspectiveTransform perspectiveTransform = new PerspectiveTransform();
    perspectiveTransform.setUlx(140);
    perspectiveTransform.setUly(54);
    perspectiveTransform.setUrx(323);
    perspectiveTransform.setUry(106);
    perspectiveTransform.setLrx(334);
    perspectiveTransform.setLry(382);
    perspectiveTransform.setLlx(151);
    perspectiveTransform.setLly(427);

    gc.drawImage(imageReelle, 0, 0);
    gc.setEffect(perspectiveTransform);
    gc.drawImage(imageVirtuelle, 0, 0);
    gc.setEffect(null);
    root.getChildren().add(canvas);

    Scene scene = new Scene(root);
    stage.setTitle("HelloPerspective");
    stage.setScene(scene);
    stage.show();
  }

  public static void main(String[] args) {
    Application.launch(args);
  }
}
```

FIGURE 1 – Résultat obtenu (à droite) avec l'exemple de code HelloPerspective, à partir d'une photo (gauche) et d'une image de fenêtre (milieu).

4 Utilisateurs visés et scénarios envisagés

Votre application s'adresse à un large public, c'est-à-dire potentiellement n'importe qui avec des projets de travaux à réaliser. La facilité d'utilisation donc est un critère essentiel à prendre en compte dans votre conception.

Un travail préliminaire auprès d'utilisateurs potentiels de l'application a permis d'identifier les scénarios d'usage suivants :

Scénario 1 : *Marie veut remplacer la porte de sa maison. Avec son smartphone, elle commence par prendre plusieurs photos de sa porte actuelle sous différents angles, photos qu'elle importe ensuite sur son ordinateur portable. Marie a repéré sur différents sites Internet des modèles de portes qui pourraient convenir mais elle hésite encore entre plusieurs modèles. Pour se décider, elle commence par ajouter ses*

photos dans votre application ainsi que les images des différents modèles de portes repérées sur Internet, sur lesquelles elle effectue éventuellement un travail de rognage. Après avoir identifié les coins de la porte sur chacune des photos, elle visualise le rendu de chacune des portes sous les différents points de vue. Satisfaite par deux modèles, elle sauvegarde plusieurs rendus pour les montrer à d'autres personnes et solliciter leur avis avant de prendre une décision finale.

Scénario 2 : *Paul veut changer un poster accroché sur un mur de sa cuisine et il hésite entre plusieurs modèles. Il réalise des opérations similaires à Marie pour enfin visualiser côte à côte les rendus des différents modèles, ce qui lui permet de choisir celui qui suit le mieux avec le reste de la cuisine.*

Scénario 3 : *Pierre veut changer la porte de son garage. Il trouve des modèles d'occasion sur Internet, cependant ceux-ci ne sont pas pris selon un point de vue face aux portes. Après avoir pris des photos sous différents angles de sa porte de garage actuelle, Pierre utilise l'application pour importer les photos prises sur Internet et identifier les quatre coins des portes d'occasion. Une fois cette étape réalisée il intègre les portes d'occasion pour comparer les rendus.*

5 Résultats attendus

L'application sera développée en utilisant JavaFX.

Vous devrez rendre une archive **ZIP** contenant :
- un répertoire src avec les sources du projet
- un répertoire resources avec les ressources nécessaires pour exécuter votre projet
- un jar exécutable **qui fonctionne dans les salles TP** (indispensable pour la séance d'évaluations croisées)
- un export au format pdf de vos mockups Balsamiq : dans votre pdf, il foit être possible de cliquer sur les éléments de l'interface pour enchaîner les écrans et ainsi explorer une séquence d'interaction.
- un compte rendu **au format pdf** contenant
  - vos noms et prénoms
  - une capture d'écran de l'application finale
  - une justification de vos choix de conception au regard des critères ergonomiques, guide de conception etc...
  - toute autre information utile permettant de mettre en valeur votre travail
- Une vidéo de présentation de votre projet, conforme aux exigences disponibles ici [3].

Le projet est à réaliser en binômes (pas de trinômes). Binômes inter-groupes non autorisés.

Barème indicatif de notation :

Critère	**Points**
Respect des consignes : archive zip et pas un autre format, capture d'écran, compte rendu au format pdf, rendu code source, jar ...	1
Le jar exécutable fonctionne en salle TP	1
Qualité des mockups Balsamiq	2
Ergonomie de l'interface	3
Qualité justifications des choix de conception dans le rapport	3
Qualité des évaluations croisées : critiques argumentées des autres interfaces sur la base de critères ergonomiques ...	2
Utilisation correcte des gestionnaires de placement	3
Gestion correcte des événements	2
Qualité du code	1
Qualité de la vidéo de présentation du projet et conformité aux instructions	2

TABLE 1 – Barème indicatif de notation

3. `https://git-iut.univ-lille1.fr/gery.casiez/videomaker`