

Rapport Bureau d'Étude

Moteur de Fusion Multimodale



Bastian Muratory
Clément Pagés
Antoine Paulus

3ème année S.R.I
BE réalisé entre le 18 Octobre et le 9 Novembre 2022

Introduction	3
Conception du moteur de fusion	4
1. aspects temporels de la fusion multimodale	4
Création de forme	4
Déplacement d'une forme	4
Suppression d'une forme	5
Modifications sur une forme à l'écran.	5
2. automate de la grammaire	6
3. conception logicielle du système	8
4. machine à états	9
Etat INITIAL	9
Etat AFFICHER_FORMES	9
Etat ATTENTE_PAROLE	9
Etat ATTENTE_CLIC	10
5. Reconnaissance de gestes	11
Conclusion	12

Introduction

L'objectif de ce bureau d'étude est de concevoir et d'implémenter un moteur de fusion permettant à un utilisateur d'interagir avec une palette de dessin ne disposant d'aucun bouton. Ainsi, pour pouvoir effectuer des actions comme créer une forme (carré, cercle ou triangle), la déplacer ou encore changer sa couleur, l'utilisateur a recours à la reconnaissance de parole et de geste et au clic souris. Le moteur de fusion implémenté a pour objectif de fusionner les informations issues de ces différentes modalités d'interactions afin que l'application effectue l'action demandée par l'utilisateur. La conception de ce moteur est décrite en première partie de ce rapport. Dans un second temps, des cas d'utilisation de l'application sont présentés à travers d'exemples illustrés.

L'intégralité du code associé à cette application ainsi que les différents agents utilisés sont disponibles au lien suivant : https://github.com/clement-pages/BE_IHM_fusion.git

Conception du moteur de fusion

1. aspects temporels de la fusion multimodale

Pour modéliser les différentes interactions entre le système et l'utilisateur avant de réaliser le moteur, nous avons eu recours à des chronogrammes. Chaque chronogramme va décrire une interaction en précisant les actions de l'utilisateur attendues dans chaque modalité.

Création de forme

On attend ici que l'utilisateur prononce "Créer ça ici" en faisant le geste lié à la figure qu'il souhaite créer, puis qu'il clique à l'endroit où il souhaite insérer la forme souhaitée. Étant donné que la reconnaissance de gestes a été ajoutée de manière itérative après l'intégration de la palette et de la reconnaissance vocale, il est également possible d'utiliser la reconnaissance vocale pour dire "Créer un carré ici" par exemple. Dans le cas d'un conflit entre geste et voix, la priorité est donnée à la voix. Le cas d'utilisation tel que décrit en phase de conception est le suivant :

CRÉER UNE FORME

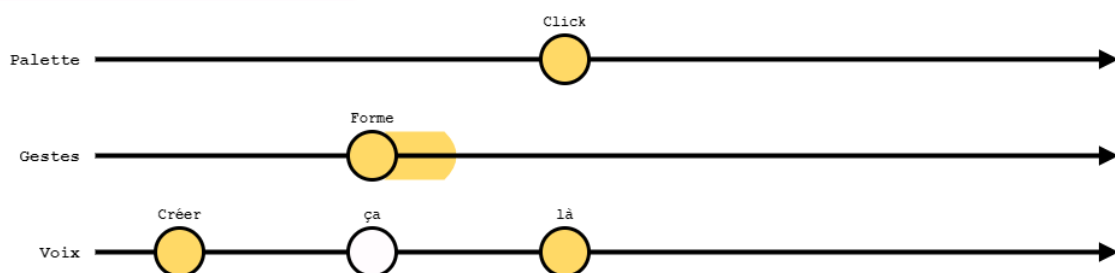


Figure 1 : Chronogramme pour la création d'une forme

Déplacement d'une forme

Nous avons ici déterminé deux cas possibles d'utilisation. Un où l'utilisateur fait chronologiquement le déplacement ("Mettre cette forme ici"). Une autre où l'utilisateur donne d'abord la destination, suivie de l'objet à déplacer. Ces deux interactions sont définies dans les figures 1 et 2.

DÉPLACER UNE FORME

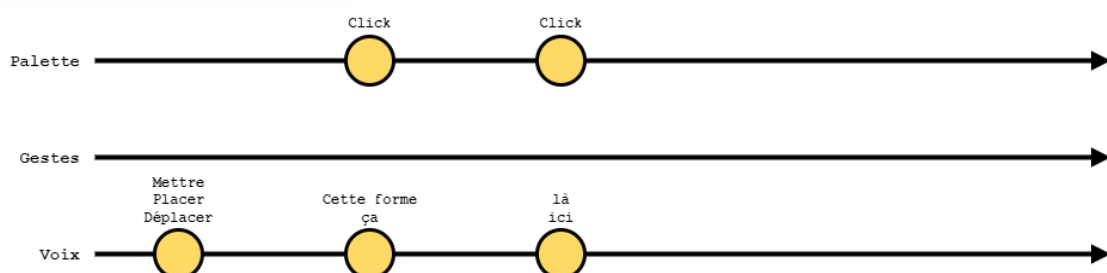


Figure 1: Chronogramme pour le déplacement d'une forme (cas origine - destination)

DÉPLACER UNE FORME (ALT)

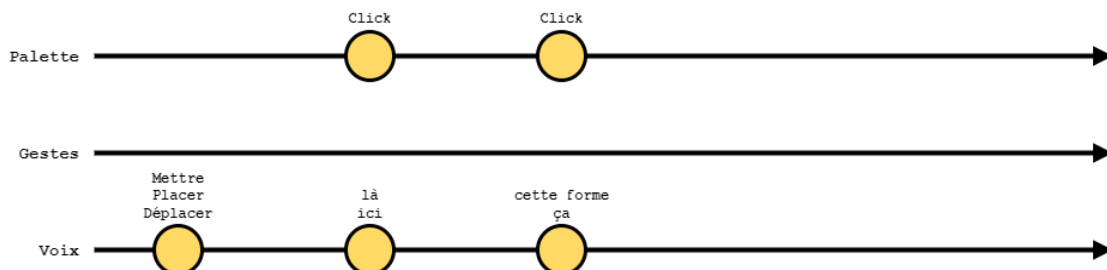


Figure 2: Chronogramme pour le déplacement d'une forme (cas destination - origine)

Suppression d'une forme

La suppression d'une forme s'effectue à l'aide d'une commande vocale associée à un clic sur la palette pour désigner la forme à enlever de l'écran. Le mécanisme est décrit en figure 3

SUPPRIMER UNE FORME

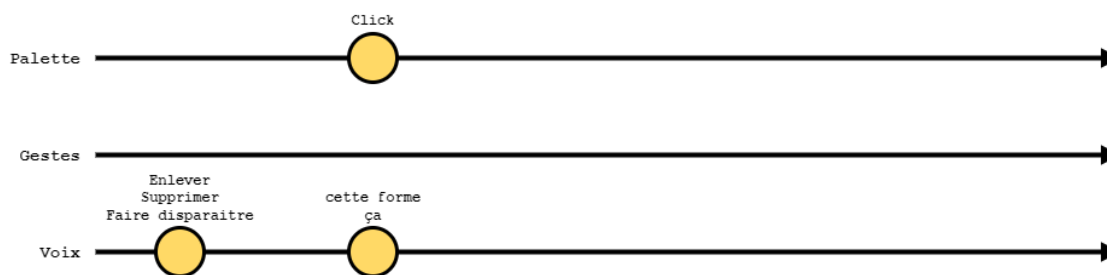


Figure 3: Chronogramme pour la suppression d'une forme

Modifications sur une forme à l'écran.

Deux modifications ont été implémentées dans le moteur. La première pour changer la couleur, on sélectionne à l'aide de la palette la forme tout en demandant de changer de couleur. La couleur peut être énoncée clairement (voir en partie 2), ou non. Elle est dans ce cas sélectionnée aléatoirement. Les figure 4 et Figure 5 illustrent respectivement ces deux modes.

CHANGER LA COULEUR D'UNE FORME

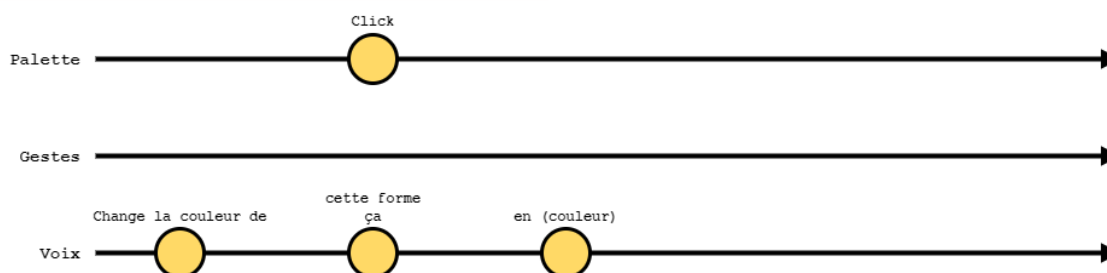


Figure 4: Chronogramme pour le changement de couleur d'une forme (défini)

CHANGER LA COULEUR D'UNE FORME

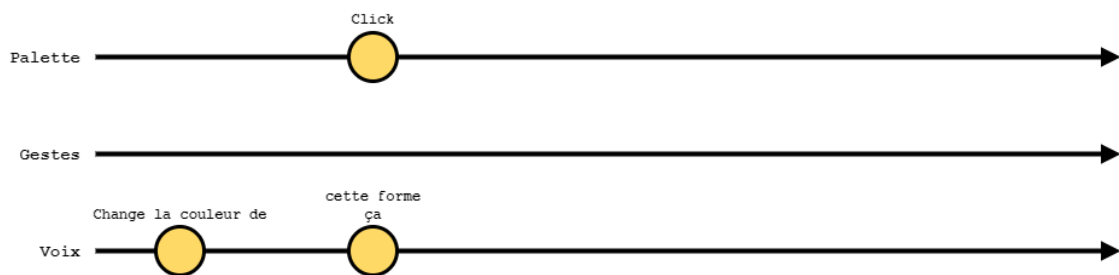


Figure 5: Chronogramme pour le changement de couleur d'une forme (aléatoire)

Enfin, il est également possible d'agrandir ou de diminuer la taille d'une forme en utilisant le même procédé (commande et clic). Nous avons pensé à utiliser les gestes pour donner un facteur d'agrandissement ou de diminution, mais sa mise en œuvre s'est révélée trop complexe. Le chronogramme de la fonction implémentée est décrit en figure 6.

CHANGER LA TAILLE

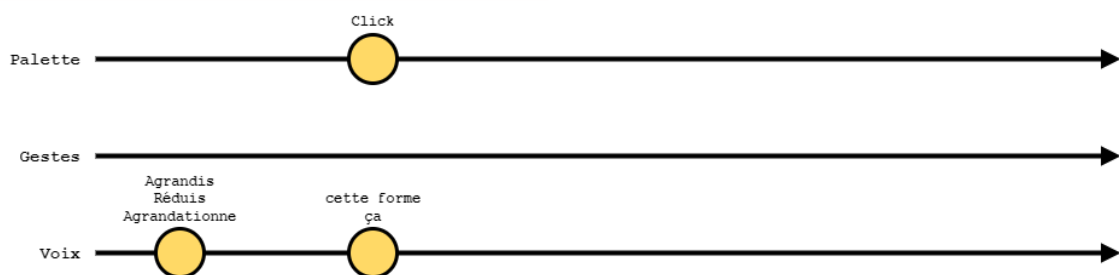


Figure 6: Chronogramme pour le changement de taille d'une forme.

2. automate de la grammaire

La reconnaissance vocale de commande émise par l'utilisateur repose sur la grammaire dont l'automate est représentée en figure 7 . L'automate prend en compte les formules de politesse de type "s'il-vous-plait" ou "peux-tu" ajoutées en début ou fin de commande selon les cas. Ces éléments sont représentés par les arcs *filler_debut* et *filler_fin* dans la figure ci-après. Le premier élément important concernant la commande est le verbe employé par l'utilisateur. En effet, c'est à partir de ce verbe que l'application est capable de déterminer l'action à effectuer sur la palette. La liste des verbes pouvant être utilisés dans la reconnaissance vocale sont listés dans le tableau suivant.

Verbes	Action correspondante
Créer / Dessiner	Ajoute une nouvelle forme à la palette
Supprimer	Supprime une forme de la palette
Mettre	Selon le reste de la commande, déplace une

	forme ou change la couleur de cette forme
Changer	Change la couleur d'une forme
Agrandir	Augmente la taille d'une forme sur la palette
Réduire / Rapetisser	Réduit la taille d'une forme sur la palette

Table 1: Correspondance verbe et action sur la palette

La grammaire laisse la possibilité à l'utilisateur d'utiliser ces verbes, soit à l'infinitif, soit conjugué à la troisième personne du singulier, à l'impératif. De plus, comme on peut le constater, la reconnaissance vocale ne permet pas de déterminer à quelle forme s'applique l'action de la commande donnée par l'utilisateur. Cette information est donnée par la position du curseur de souris lors d'un clic réalisé par l'utilisateur. La forme de la palette à laquelle s'applique l'action est alors celle se trouvant sous le curseur.

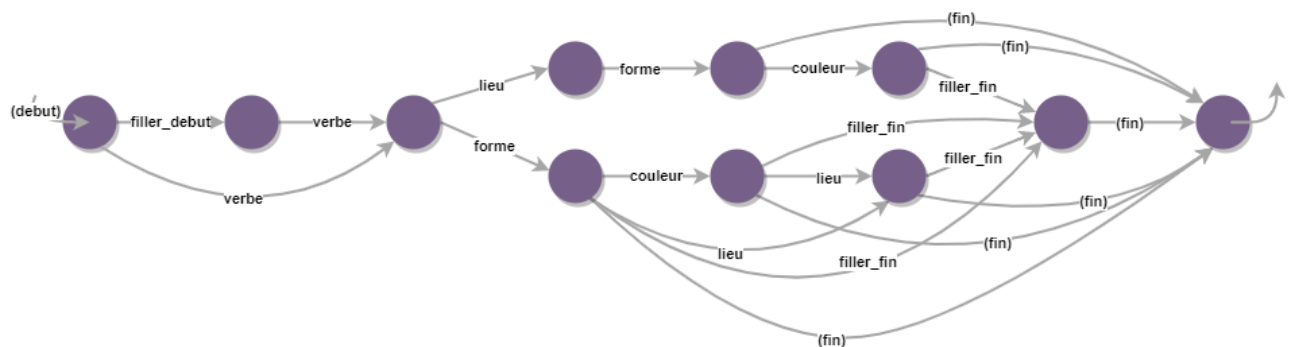


Figure 7: Automate de la grammaire utilisée pour la reconnaissance vocale de commande

Après le verbe, la grammaire laisse deux possibilités à l'utilisateur :

- Donner d'abord la forme, avec éventuellement une couleur selon l'action à réaliser, puis un lieu sur la palette (optionnel). Cette possibilité est désignée comme étant l'ordre classique.
- Donne d'abord un lieu sur la palette puis une forme avec éventuellement une couleur selon l'action à réaliser.

Si l'utilisateur peut spécifier le type de la forme parmi cercle, carré ou triangle, notamment lors de la création d'une forme (dans le cas où l'utilisateur n'utilise pas la reconnaissance de geste), la plupart du temps, le mot utilisé pour désigner une forme sera "FORME". Tout comme précédemment, cela signifie que la forme à laquelle l'action va s'appliquer sera identifiée via un clic souris sur la palette. Les couleurs reconnues par la grammaire sont indiquées dans le tableau suivant :

Couleur	Correspondance
Rouge	RGB = (255, 0, 0)
Vert	RGB = (0, 255, 0)
Bleu	RGB = (0, 0, 255)

Couleur	Couleur de la forme indiqué par un clic ou couleur aléatoire si aucune forme n'est spécifiée
---------	--

Table 2: Correspondance entre couleur grammaire et couleur sur la palette

Enfin, le lieu n'est pas une autre information qui n'est pas fourni par la grammaire, mais par la position du curseur lors d'un clic souris sur la palette. Ainsi, l'utilisateur ne peut pas indiquer via la reconnaissance vocale l'endroit sur la palette où déplacer une forme, par exemple. Il peut seulement indiquer "ici" ou "là", la notion correspondant à ces termes étant donnée par le clic souris dans le moteur de fusion.

3. conception logicielle du système

La figure 8représente le diagramme de classe décrivant le système conçu pour la palette.

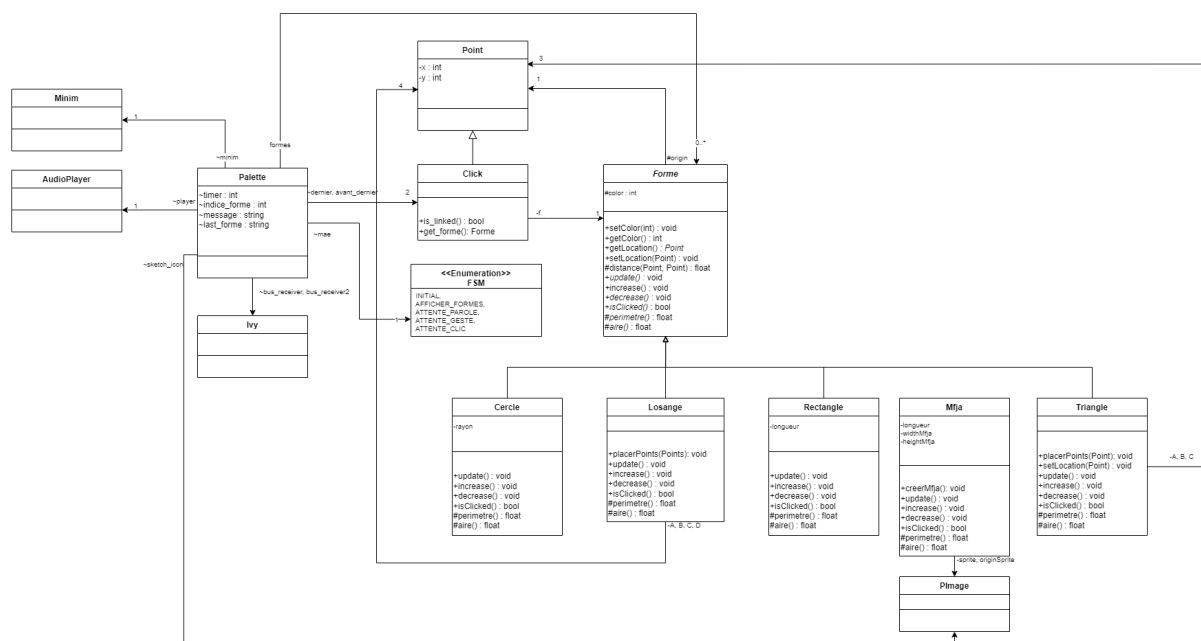


Figure 8: Diagramme de classe du système représentant la palette

Le principal fichier du système, celui comportant les fonctions *setup* et *draw* est le fichier *Palette*. Ce dernier comporte les liens vers l'ensemble des classes de l'application permettant de l'initialiser et d'interagir avec l'utilisateur. On retrouve en particulier :

- Une instance de la classe Ivy. Cette instance permet d'écouter sur le bus Ivy les messages en provenance des deux agents utilisés dans le cadre de l'application : sra5 et OneDollar. Elle permet également de gérer la réception d'un message en provenance de ces agents.
- Un ArrayList de Formes. Cette liste correspond à l'ensemble des formes se trouvant à un instant donné sur la palette. Une forme peut être un Cercle, un Rectangle / Carré, un Losange ou un Triangle (ainsi que Mfja). L'ajout ou la suppression d'une forme de cette liste a donc pour conséquence d'ajouter ou de supprimer une forme sur l'affichage. Par défaut, une

forme comporte un point d'origine, à partir duquel la forme est construite, ainsi qu'une couleur, sous la forme d'un entier correspondant à un code RVB. Certaines formes, telles que le Triangle et le Losange ont d'autres points comme attributs, ces points participant à la construction des formes.

- La gestion de la machine à état régissant le fonctionnement de la Palette. Les états correspondent à la situation dans laquelle se trouve l'application à un instant particulier : affichage des formes, attente d'une entrée utilisateur... Ce dernier point est décrit plus en détail dans la partie suivante.
- La gestion des clics. Les deux derniers clics réalisés par l'utilisateur sont sauvegardés en conservant leur position ainsi que la forme qui leur est associée. Cela permet notamment de réaliser les commandes de déplacement et de modifications des formes sur l'interface de la Palette.

4. machine à états

L'application Palette fonctionne avec une machine à états composée de 4 états : état *INITIAL*, *AFFICHER_FORMES*, *ATTENTE_PAROLE*, *ATTENTE_CLIC*. La machine à état est représentée en figure 9. Ces différents états définissent le comportement de l'application ainsi que les entrées qu'elle est susceptible de recevoir en provenance de l'utilisateur pour changer d'état. L'ensemble des états sont décrits dans les paragraphes suivants.

Etat INITIAL

L'état INITIAL n'est présent qu'au début de l'application, il va de lui-même vers le prochain état *AFFICHER_FORMES*. La liste étant vide initialement, aucune forme n'est affichée sur la palette.

Etat AFFICHER_FORMES

AFFICHER_FORMES est l'état dans lequel l'application reste en attente des interactions avec l'utilisateur. Il s'agit de l'état principal du système.

Depuis l'état *AFFICHER_FORMES* il est possible de rejoindre les états suivants :

- *ATTENTE_PAROLE* (après un clic souris)
- *ATTENTE_CLIC* (après réception d'un message gestuel depuis Ivy)

Etat ATTENTE_PAROLE

ATTENTE_PAROLE est l'état dans lequel l'application a reçu un ou plusieurs clics souris et attends donc la commande de l'utilisateur qui va déclencher l'action voulue. Les actions possibles ont

été définies dans la partie précédente sur la grammaire. Dans cet état il est possible de transitionner vers l'état suivant :

- AFFICHER_FORMES (après un délai de quelques secondes) : Ici on considère que l'utilisateur a seulement cliqué sur l'écran sans réelles intentions, cela peut être une erreur donc on revient dans l'état principal.
- AFFICHER_FORMES (après une commande vocale reconnue) : cette transition est réalisée quand l'utilisateur a correctement donné une commande vocale. Dans ce cas là, la commande reconnue est exécutée.

Etat ATTENTE_CLIC

ATTENTE_CLIC est l'état dans lequel l'application a reçu un geste ou une commande vocale depuis Ivy et attend l'emplacement où devra aller la forme reconnue. Dans cet état il est possible de transitionner vers l'état suivant :

- AFFICHER_FORMES (après un délai de quelques secondes) : Ici on considère que l'utilisateur a seulement effectué un geste sans réelles intentions, cela peut être une erreur donc on revient dans l'état principal.
- AFFICHER_FORMES (après un clic) : Dans ce cas là, l'utilisateur a bien précisé la localisation sur la palette où la commande doit être exécutée et / ou la forme à laquelle s'applique la commande.

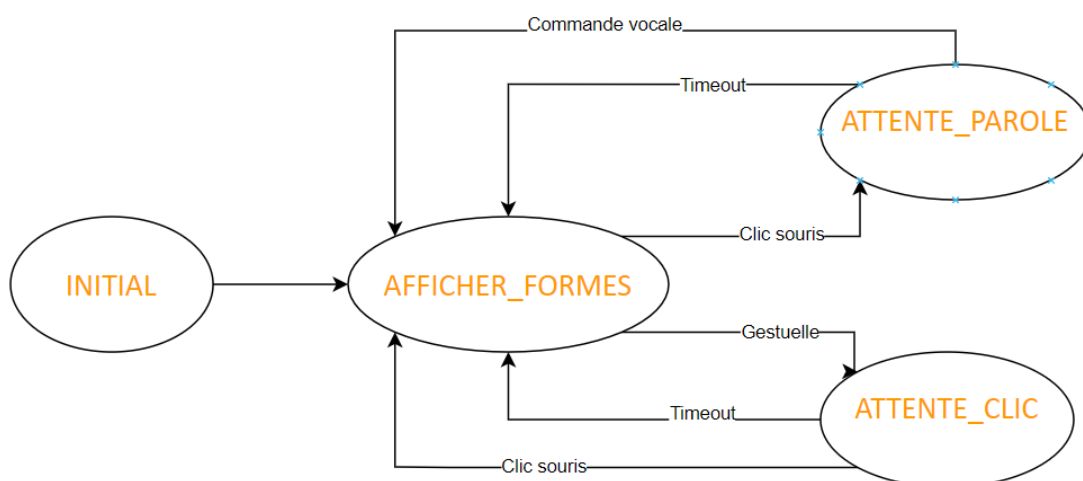


figure 9: Diagramme de la machine à état du comportement du logiciel palette.

5. Reconnaissance de gestes

La reconnaissance de geste se fait grâce au logiciel Gestuelle qui utilise le système du \$1 Recognizer (<https://depts.washington.edu/accelab/proj/dollar/index.html>). Ce système tourne en parallèle de la Palette et reconnaît les gestes suivants :

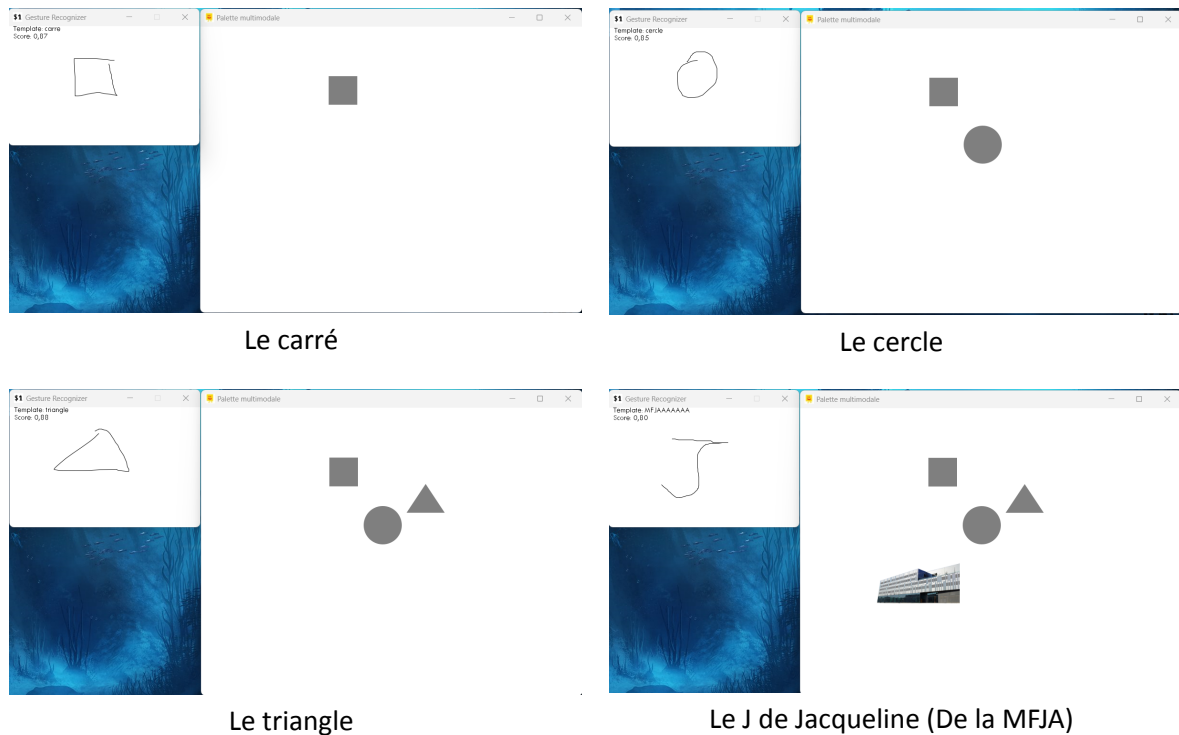


Figure 10: Exemple de création de forme à partir de la reconnaissance de geste

Ces différentes formes peuvent ensuite être placées sur la palette à l'endroit voulu par l'utilisateur.

Conclusion

L'objectif de ce bureau d'étude a été de réaliser un moteur de fusion multimodale permettant la gestion d'un ensemble de formes affichées sur une palette. Les modalités utilisées sont la reconnaissance vocale, la reconnaissance de geste ainsi que la détection de clics souris sur l'interface graphique. Chacune de ces modalités apporte une partie des informations nécessaire à l'exécution d'une commande utilisateur, le moteur de fusion, comme son nom l'indique, ayant pour objectif de fusionner ses différentes informations afin de mettre à jour l'interface. Pour se faire, le comportement du système est contrôlé par une machine à états permettant de récupérer toutes les entrées de l'utilisateur et de mettre à jour la palette. Cette logique permet à l'utilisateur de réaliser un certain nombre d'interactions avec le système tel que :

- Créer des formes avec la voix puis sélectionner leur emplacement avec un clic. Cette action peut aussi se faire via la reconnaissance gestuelle, en dessinant la forme à créer sur la palette
- Déplacer une forme, avec un clic sur la forme à déplacer puis un second clic vers la destination
- Des interactions spécifiques à une forme : agrandir, rétrécir, changer de couleur.
- Supprimer une forme de la palette