

STAUBLI CS 9 – Créer un cycle de pick and place sous SRS 2019

Ce document présente la méthodologie de réalisation d'un cycle de pick and place avec un robot Staubli 6 axes associé à un contrôleur CS9. On commencera par définir l'outillage, apprendre les repères utilisateurs et les points nécessaires puis définir une trajectoire de pick and place. Enfin, nous améliorons l'application en réalisant une interface graphique et en utilisant une architecture de programme organisée en tâche principale et sous tâches.

1- Création de la cellule sous SRS2019



Figure 1 : Création d'une cellule sous SRS

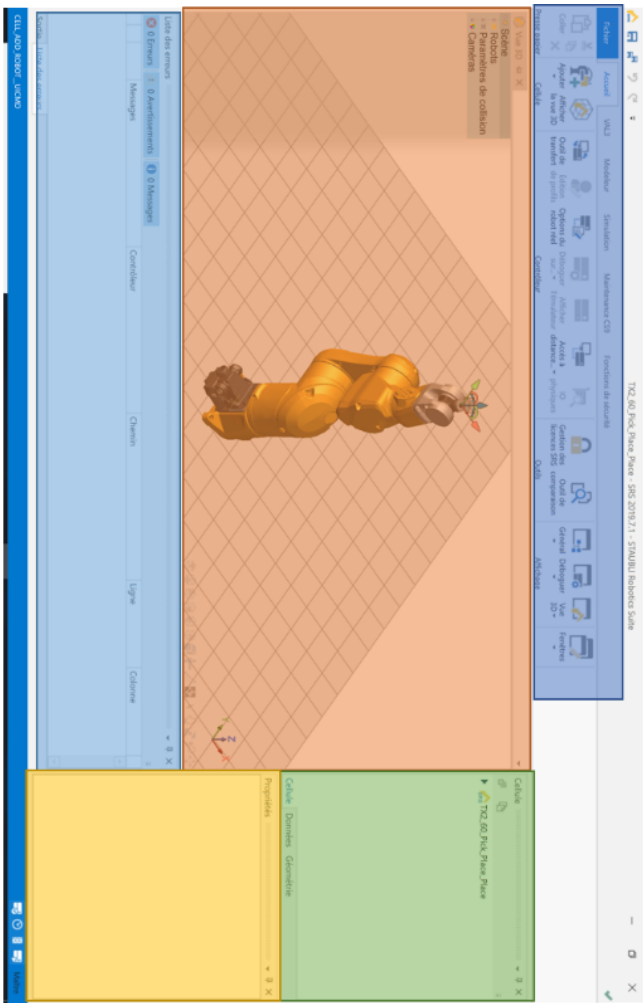


Figure 3 : Rappel de la vue principale de SRS

2- Ajout du préhenseur

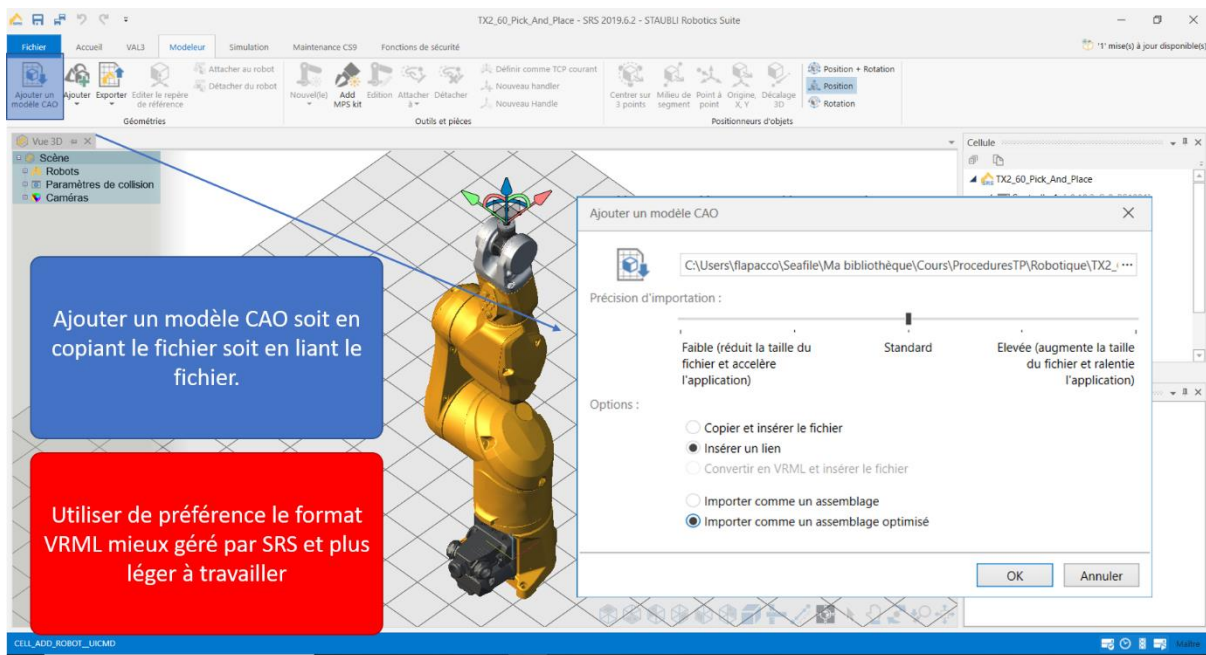
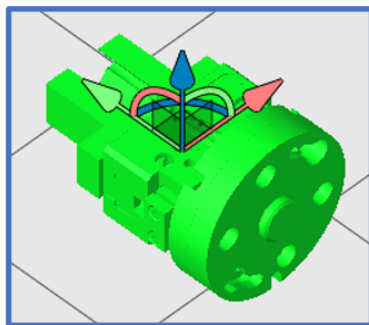
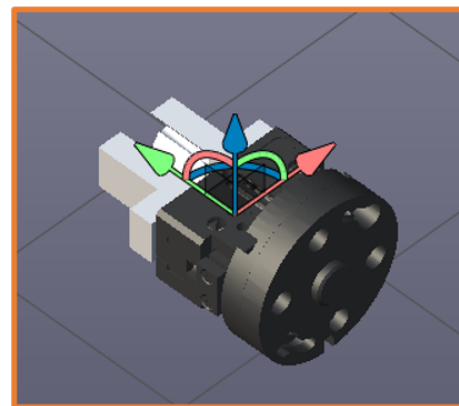
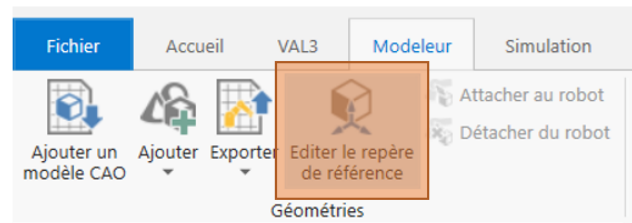


Figure 4 : Insérer un fichier CAO



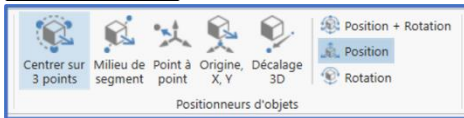
Préhenseur importé

Utiliser le trièdre pour déplacer le préhenseur dans une zone visible

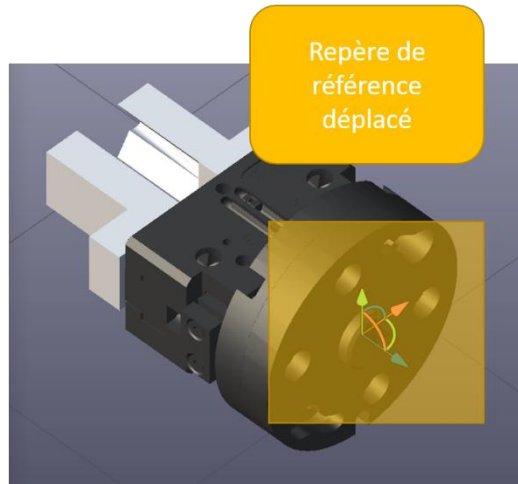
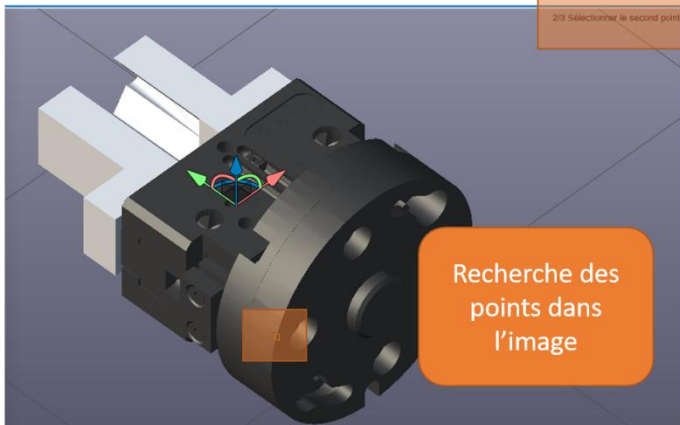


Edition du repère de référence

Figure 5 : Edition du repère de référence

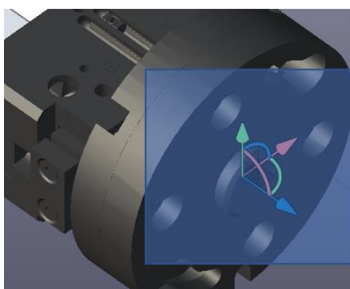


Positionneur d'objet

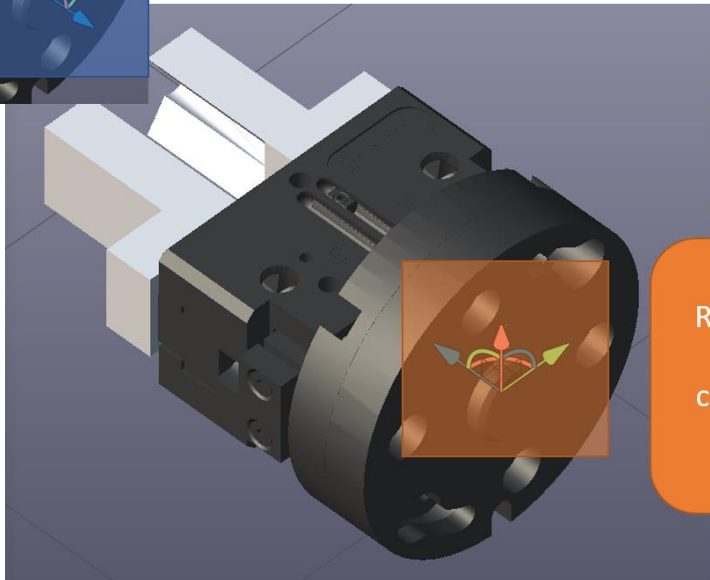


Utiliser le bon positionneur suivant le besoin et choisir le bon type de déplacement (position et/ou rotation)

Figure 6 : Utilisation du positionneur d'objet pour déplacer le repère de référence



Utiliser l'édition de position pour des déplacements simples



Editer la position - Refer...	
Position relative	
X	0.00
Y	0.00
Z	0.00
Rx	180.00
Ry	0.00
Rz	-90.00

Figure 7 : Utilisation de l'éditeur de position pour déplacer le repère de référence

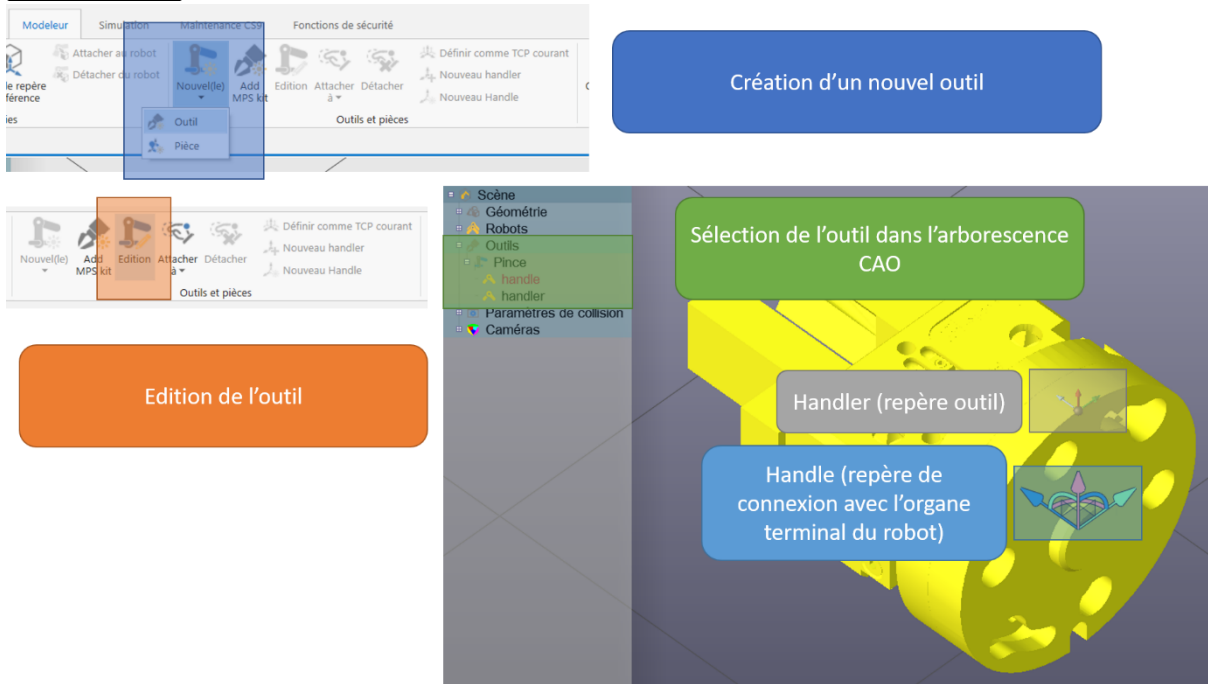


Figure 8 : Création et édition d'un outil

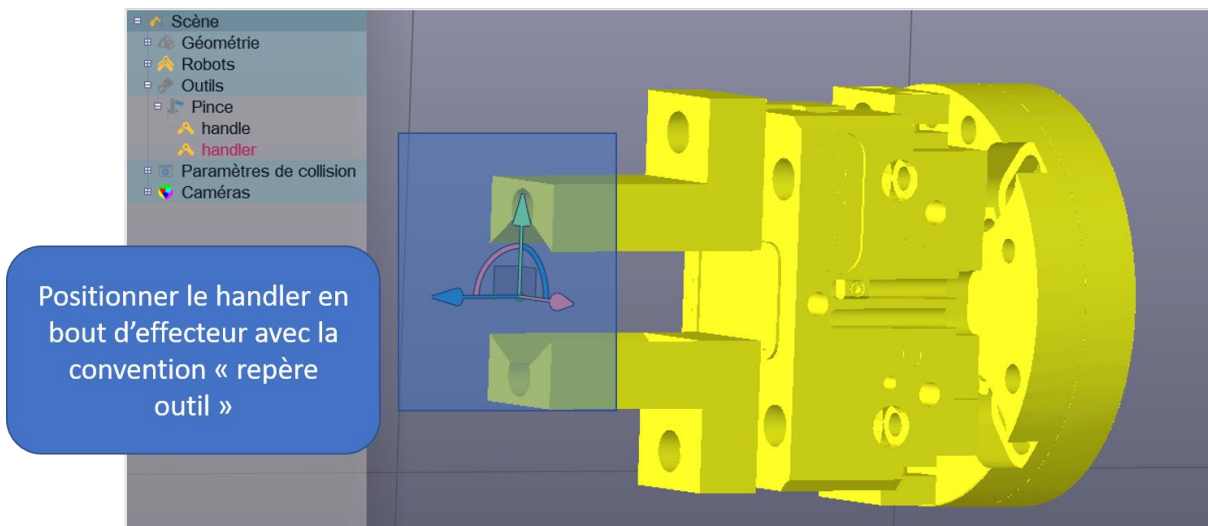
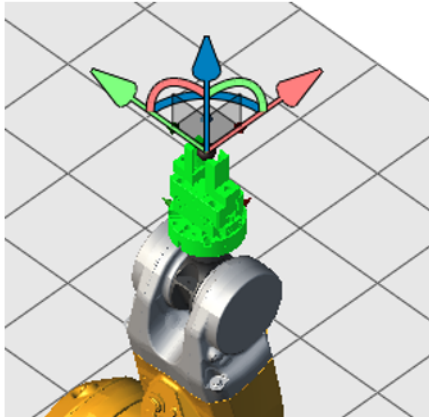


Figure 9 : Positionnement du handler en convention "repère outil"



Attacher l'outil au robot

Attention aux conventions de repère

Figure 10 : Montage de l'outil sur le robot

3- Ajout de l'environnement 3D

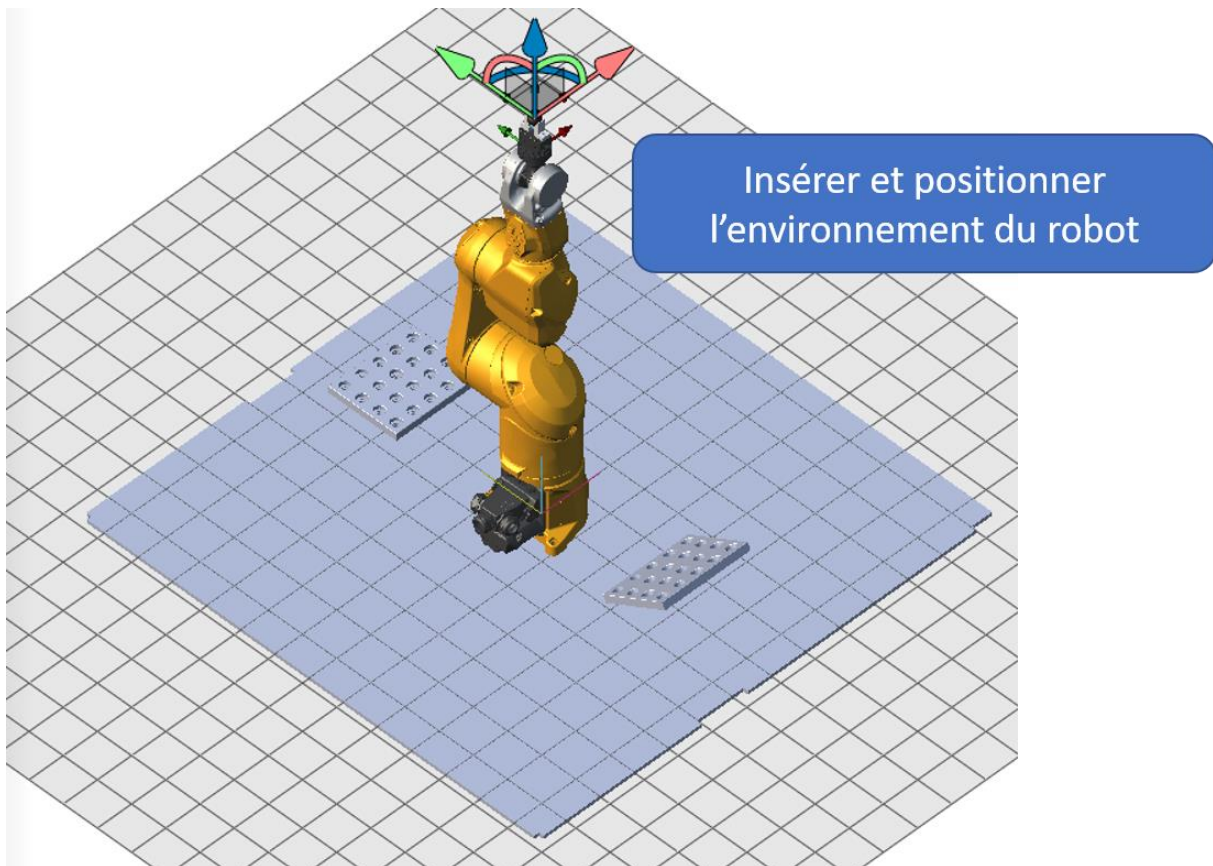


Figure 11 : Insertion de l'environnement 3D du robot

4- Création de l'application

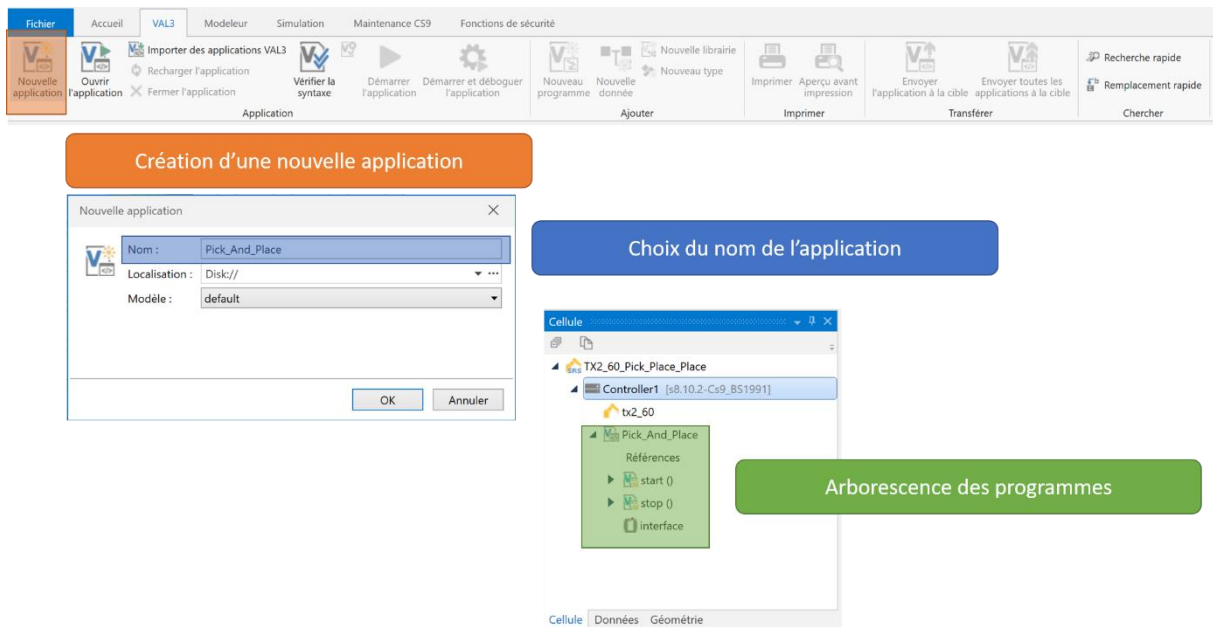


Figure 12 : Création d'une application VAL3

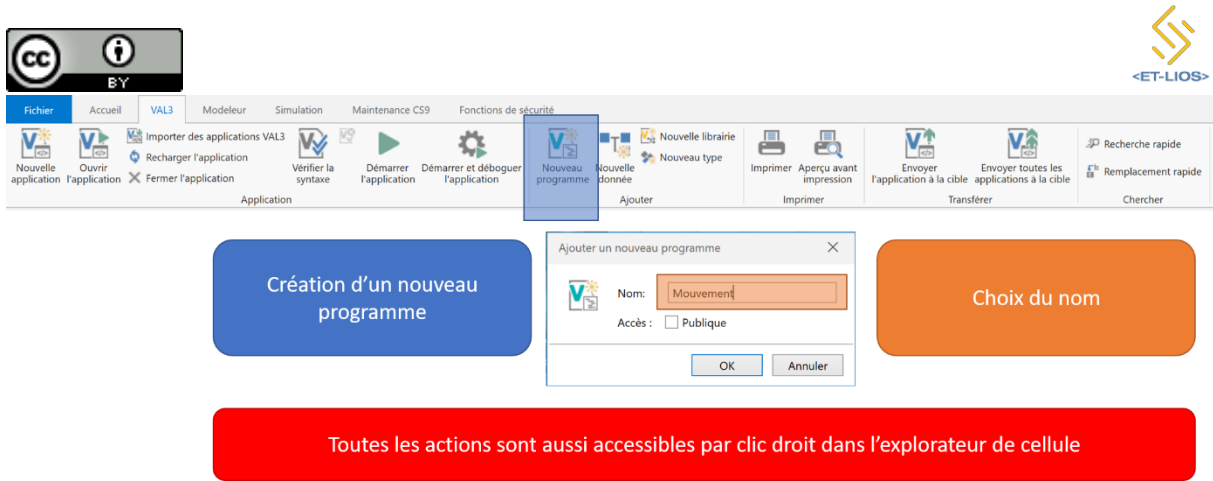
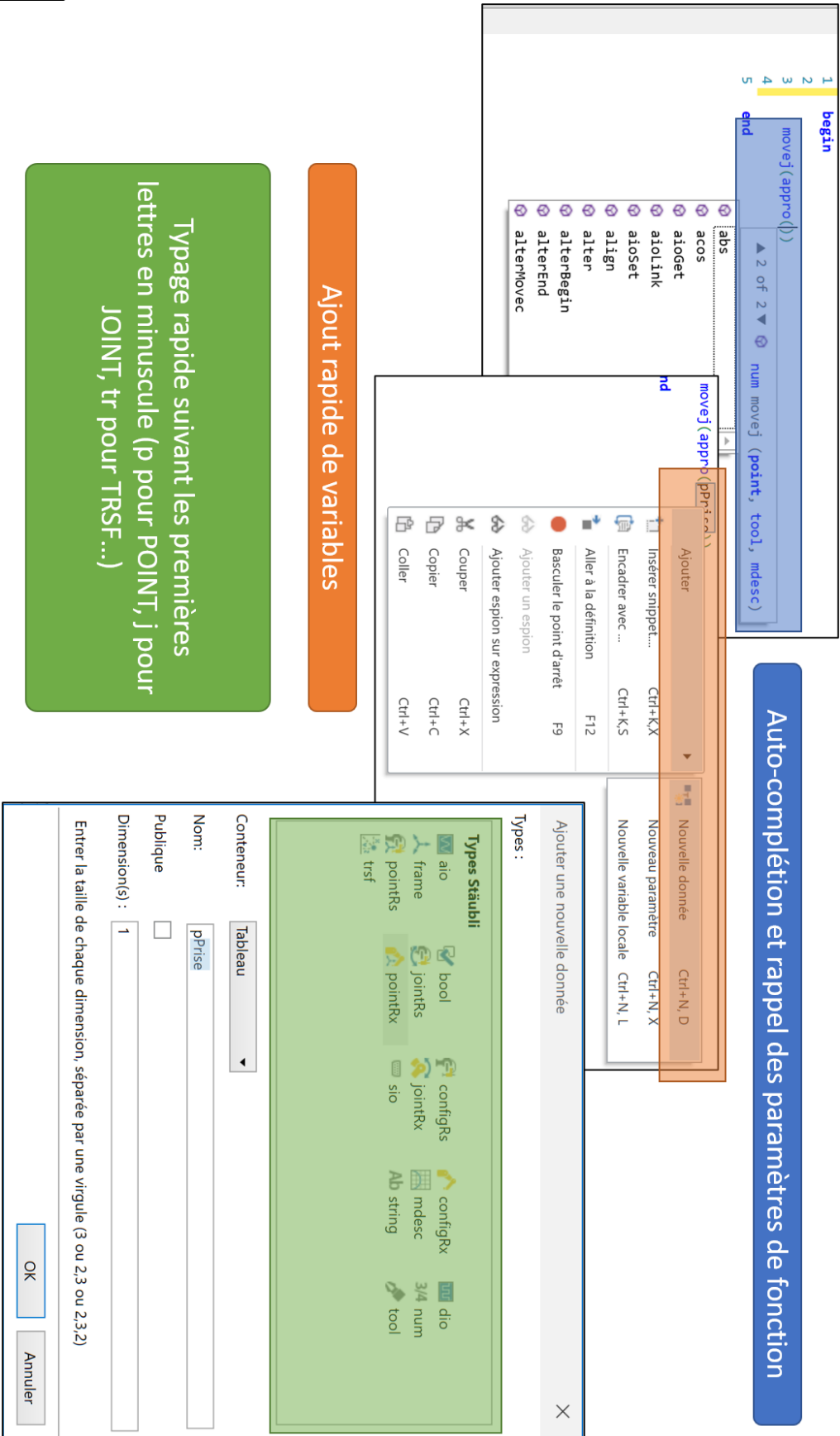


Figure 13 : Création d'un programme de Mouvement

The image shows a multi-step process in the VAL3 IDE:

- Code Editor:** A code window with five lines. Line 2 contains `movej (appro())`. A context menu is open over this line, listing functions like `abs`, `acos`, `aiogot`, `aiolink`, `aioset`, `align`, `alter`, `alterBegin`, `alterEnd`, and `alterMovec`.
- Context Menu:** A secondary menu is open, showing actions like `Ajouter` (Ctrl+KX), `Insérer snippet...` (Ctrl+KS), `Encadrer avec ...` (Ctrl+KS), `Aller à la définition` (F12), `Basculer le point d'arrêt` (F9), `Ajouter un espion`, and `Ajouter espion sur expression`.
- Dialog Box:** A dialog titled "Ajouter une nouvelle donnée" (Add new data) is open. It has a "Types" list with categories like "Types Stäubli" (aio, bool, configs, string, dio, num, tool, trsf) and "Ajouter une nouvelle donnée" (Nouvelle donnée, Nouvelle paramètre, Nouvelle variable locale). Below the list are fields for "Conteneur" (set to "Tableau"), "Nom:" (set to "pPrise"), "Publique" (checkbox), and "Dimension(s) : 1". A note at the bottom says "Entrer la taille de chaque dimension, séparée par une virgule (3 ou 2,3 ou 2,3,2)".

Auto-complétion et rappel des paramètres de fonction

Ajout rapide de variables

Typage rapide suivant les premières lettres en minuscule (p pour POINT, j pour JOINT, tr pour TRSF...)

Figure 14 : Edition d'un programme VAL3

5- Pick and place simple

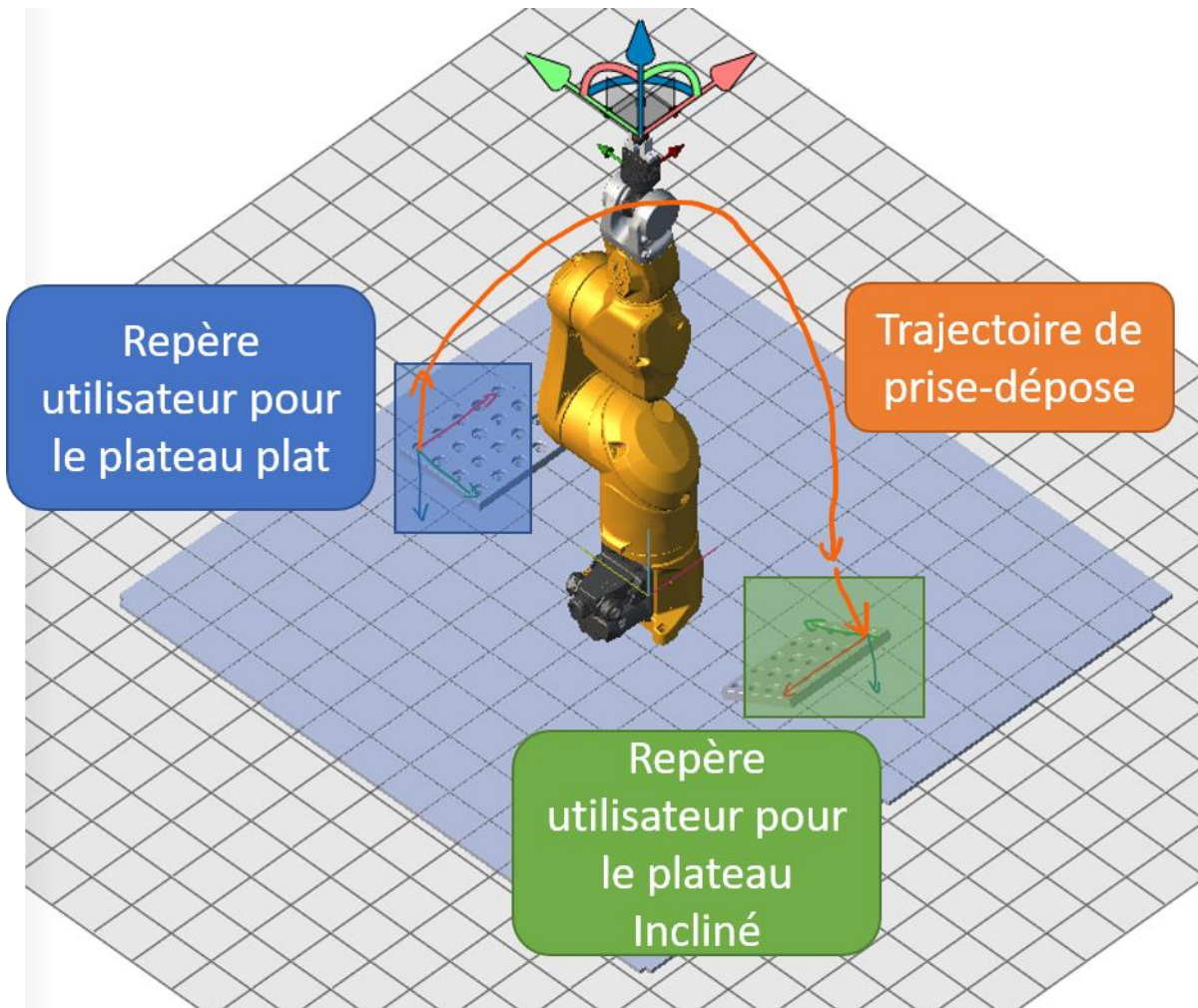


Figure 15 : Synoptique du fonctionnement du pick and place

```

1  begin
2
3  moveJ (appro (pPrise, trApproche), tPince, mNomSpeed)
4  moveL (pPrise, tPince, mNomSpeed)
5  close (tPince)
6  moveL (appro (pPrise, trApproche), tPince, mNomSpeed)
7
8  moveJ (appro (pDepose, trApproche), tPince, mNomSpeed)
9  moveL (pDepose, tPince, mNomSpeed)
10 open (tPince)
11 moveL (appro (pDepose, trApproche), tPince, mNomSpeed)
12
13 end

```

```

1  begin
2
3  call Mouvement ()
4
5  waitEndMove ()
6
7  end

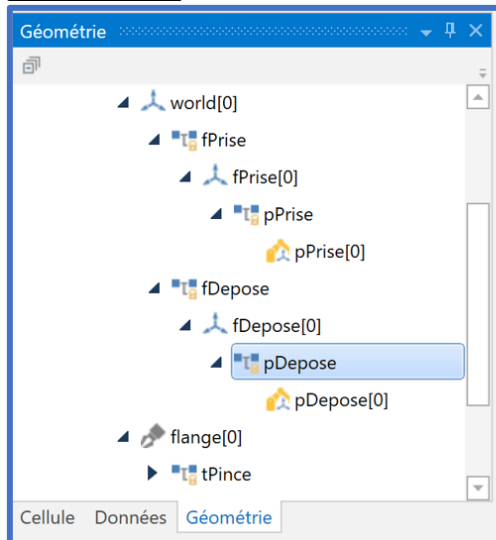
```

- Programme de base pour un pick and place :**
1. Approche pPrise en moveJ
 2. Descente prise en moveL
 3. Préhension
 4. Remontée en moveL
 5. Approche pDepose en moveJ
 6. Descente pDepose en moveL
 7. Dépose
 8. Remontée en moveL

- Programme principal (start()) :**
1. Appel du pick and place
 2. Attente de la fin de mouvement



Figure 16 : Programme de base pour un Pick and Place



Définition de repères utilisateurs et de point dans ces repères

Figure 17 : Définition des repères utilisateur

```

1 begin
2
3   for _l_nI=0 to 19
4
5     movej( appro(priise[_l_nI], trApproche), tPince, mNomSpeed)
6     moveL(priise[_l_nI], tPince, mNomSpeed)
7     close(tPince)
8     moveL( appro(priise[_l_nI], trApproche), tPince, mNomSpeed)
9
10    movej( appro(pdepose[_l_nI], trApproche), tPince, mNomSpeed)
11    open(tPince)
12    moveL(pdepose[_l_nI], tPince, mNomSpeed)
13    moveL( appro(pdepose[_l_nI], trApproche), tPince, mNomSpeed)
14
15  endfor
16
17 end

```

Ajouter Insérer snippet... Encadrer avec ...

Nouvelle donnée Nouvelle paramètre Nouvelle variable locale

Index	X	Y	Z	Rx	Ry	Rz	Shoulder	Elbow	Wrist
0	0	0	0	0	0	0	ssame	esame	wsame
1	0	0	0	0	0	0	ssame	esame	wsame
2	0	0	0	0	0	0	ssame	esame	wsame
3	0	0	0	0	0	0	ssame	esame	wsame
4	0	0	0	0	0	0	ssame	esame	wsame
5	0	0	0	0	0	0	ssame	esame	wsame
6	0	0	0	0	0	0	ssame	esame	wsame
7	0	0	0	0	0	0	ssame	esame	wsame
8	0	0	0	0	0	0	ssame	esame	wsame
9	0	0	0	0	0	0	ssame	esame	wsame
10	0	0	0	0	0	0	ssame	esame	wsame

Données 'Pick_And_Place-trApproche'

Index	X	Y	Z	Rx	Ry	Rz
0	0	0	-100	0	0	0

Réglage de l'approche

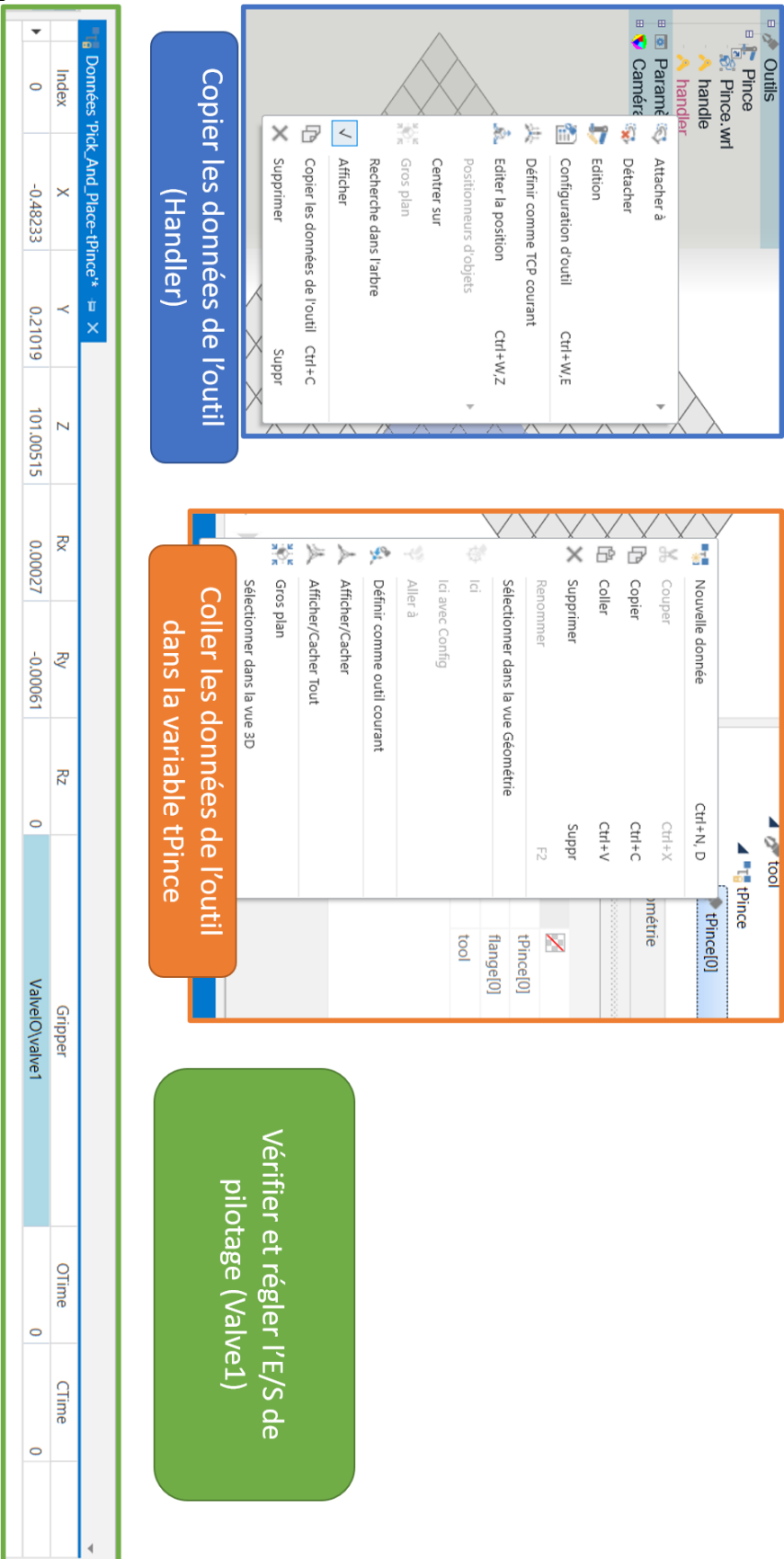
Création et saisie des points

Index	X	Y	Z	Rx	Ry	Rz	Shoulder	Elbow	Wrist
0	0	0	0	0	0	0	ssame	esame	wsame
1	0	0	0	0	0	0	ssame	esame	wsame
2	0	0	0	0	0	0	ssame	esame	wsame
3	0	0	0	0	0	0	ssame	esame	wsame
4	0	0	0	0	0	0	ssame	esame	wsame
5	0	0	0	0	0	0	ssame	esame	wsame
6	0	0	0	0	0	0	ssame	esame	wsame
7	0	0	0	0	0	0	ssame	esame	wsame
8	0	0	0	0	0	0	ssame	esame	wsame
9	0	0	0	0	0	0	ssame	esame	wsame
10	0	0	0	0	0	0	ssame	esame	wsame

Données

- TY2_60_Pick_Place_Place
- Controllert1 [s8;10.2;C9;_BS1991]
- tx2_60
- Pick_And_Place
 - mdesc
 - pointRx
 - pDepose [20]
 - pPrise [20]
 - tool

Figure 18 : Utilisation d'une boucle répétitive et paramétrage des variables

Copier les données de l'outil (Handler)

Coller les données de l'outil dans la variable tPince

Vérifier et régler l'E/S de pilotage (Valve1)

Données 'Pick_And_Place-tPince'	X	Y	Z	Rx	Ry	Rz	Gripper	OTime	CTime
Index	X	Y	Z	Rx	Ry	Rz	ValveQValve1	0	0
	0	-0,48233	0,21019	101,00515	0,00027	-0,00061			

Figure 19 : Apprentissage du repère outil en CAO

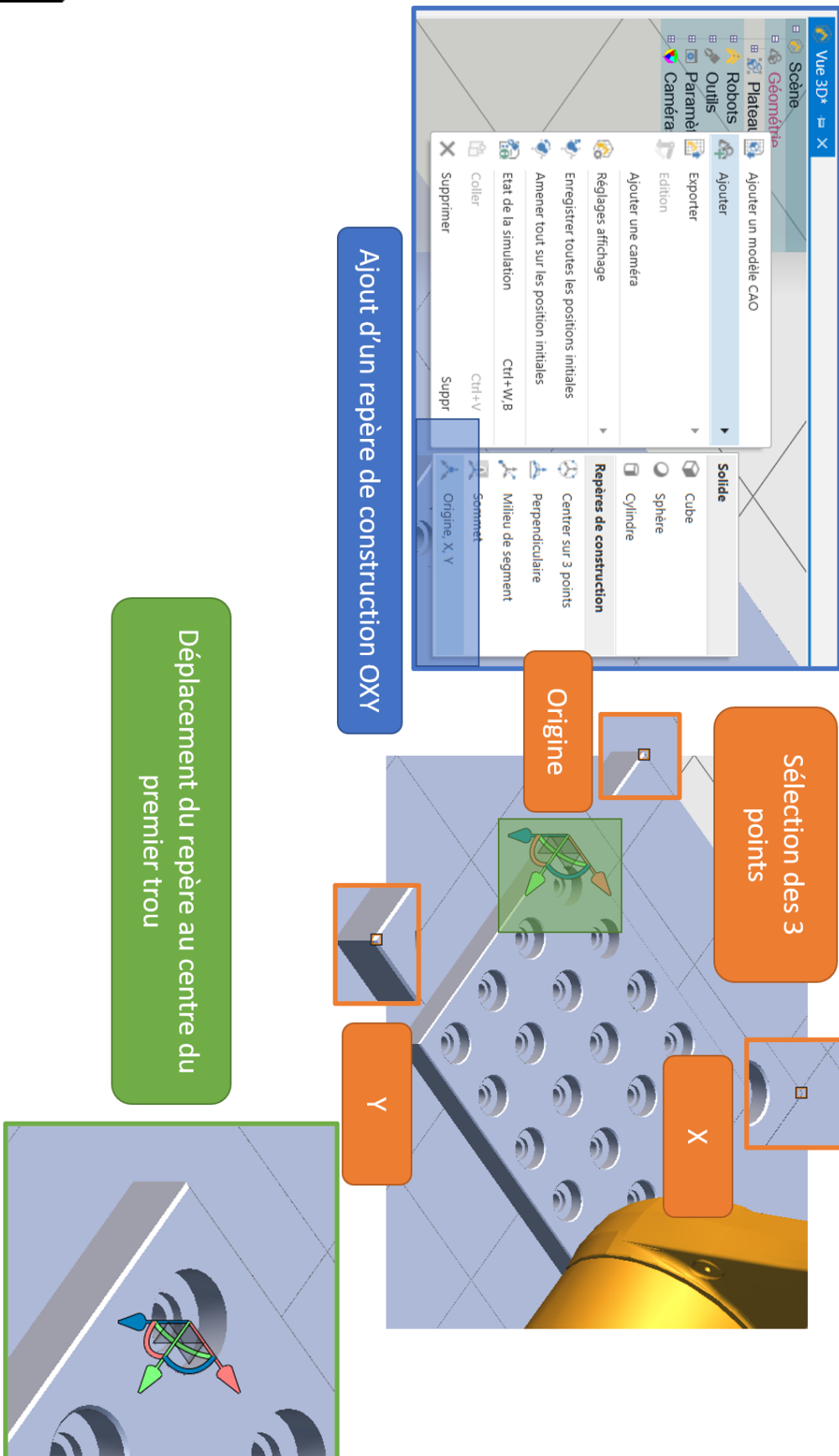
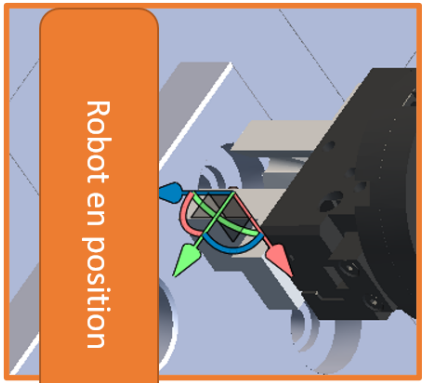


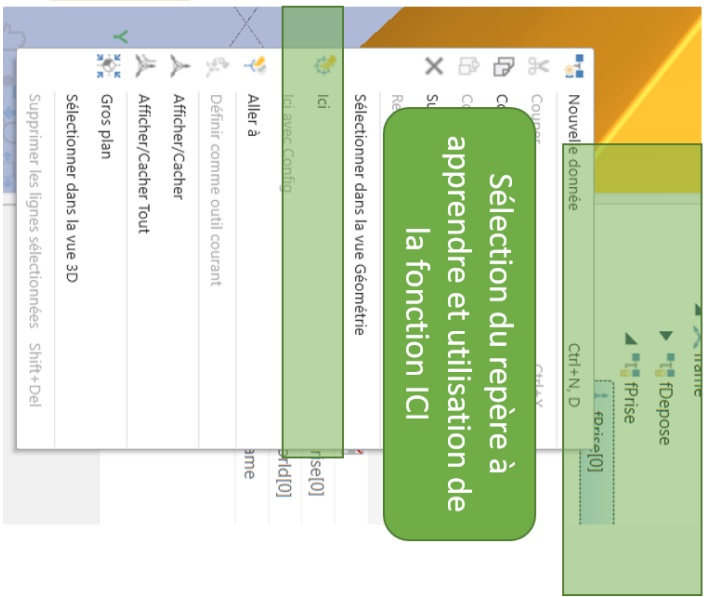
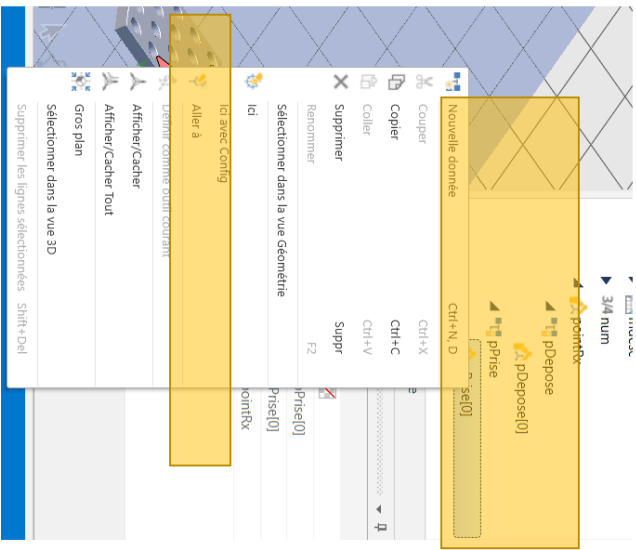
Figure 20 : Création d'un repère CAO



Déplacer le robot sur le repère CAO



Robot en position



Sélection du repère à apprendre et utilisation de la fonction ICI

Sélection d'un point pour vérification et utilisation de la fonction ALLER A



Figure 21 : Apprentissage d'un repère en CAO

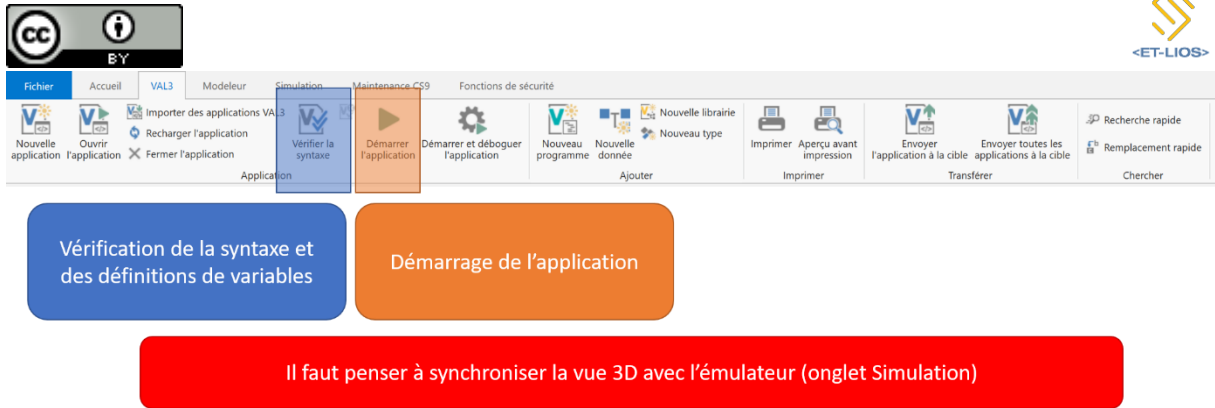


Figure 22 : Démarrage d'une application

6- Amélioration du code, ajout d'une IHM et d'un fonctionnement continu

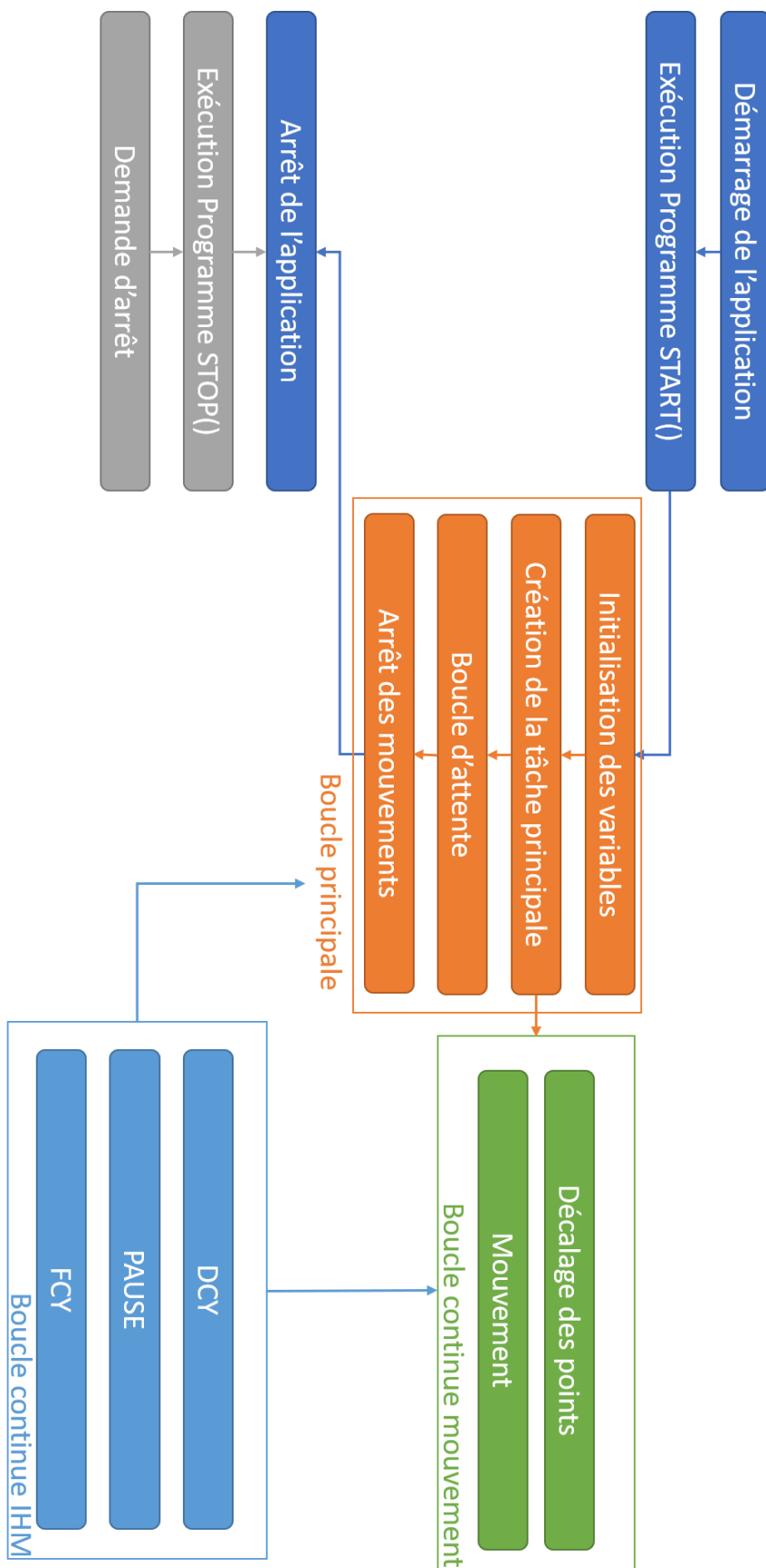
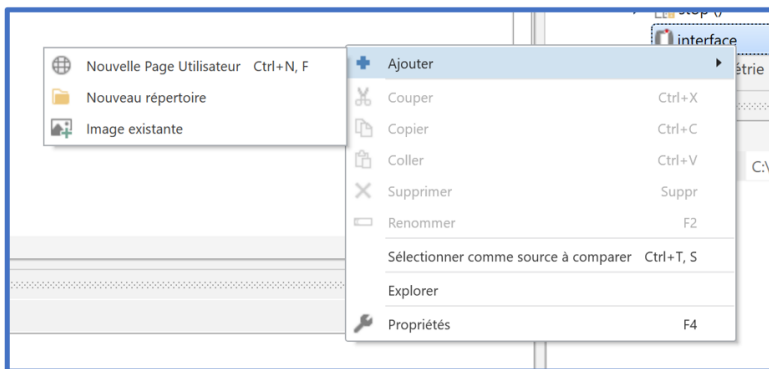


Figure 23 : Synoptique de fonctionnement de l'application



Ajout d'une page utilisateur



Figure 24 : Ajout d'une IHM de base



Appels
Click (enfoncement + relâchement) **Click_DCY**
MouseDown (enfoncement)
MouseDown (relâchement)

Courant
Id btn1

Classe CSS

Désactivé DCY

Texte

ID du contrôle
btn1

Propriété du contrôle
disabled

Variable VAL3
sbindcy[0]

Mode
Lecture

Lien entre le label du bouton et une variable VAL3

Lien entre l'état du bouton et une variable VAL3

Association d'un programme à une action sur le bouton

OK Annuler

Figure 25 : Paramétrage d'un bouton d'action

```
Pick_And_Place-start  ⇄ ×
1  begin
2
3  userPage("Principale")
4
5  // Initialisation
6  call Init()
7
8  // Création de la tache de pick and place
9  taskCreate "PickAndPlace",10,Pick_And_Place()
10
11 // Mise en attente
12 taskSuspend("PickAndPlace")
13
14 // Boucle principale
15 while(taskStatus("PickAndPlace")!=-1)
16   endwhile
17
18 taskKill("PickAndPlace")
19
20 end
```

Programme de démarrage

```
Pick_And_Place-stop  ⇄ ×
1  begin
2
3  bArret=true
4  stopMove()
5  resetMotion()
6  taskKill("PickAndPlace")
7
8  end
```

Programme si arrêt demandé hors IHM

Figure 26 : Programmes Start() et Stop()

```
Pick_And_Place-Init  ⇄ ×
1  begin
2
3  // Initialisation des points
4  pPrise.trsf=pDepose.trsf={0,0,0,0,0,0}
5
6  // Paramétrages des pas en X et Y
7  nPasX=50
8  nPasY=50
9
10 sBtnDCY="DCY"
11
12 bBtnFCY=true
13
14 bDCY=bArret=bBtnDCY=false
15
16 end
```

Initialisation

Figure 27 : Programme d'initialisation des données

```
Pick_And_Place-Pick_And_Place  X
1  begin
2
3  while(!bArret)
4
5      for nCompteurLigne = 1 to 4
6          for nCompteurColonne = 0 to 4
7
8              // Décalage en X
9              pPrise.trsf.x=pDepose.trsf.x+nCompteurColonne*nPasX
10
11             call Mouvement()
12
13         endFor
14
15         // Remise à zéro pour X
16         pPrise.trsf.x=pDepose.trsf.x=0
17
18         // Décalage en Y
19         pPrise.trsf.y=pDepose.trsf.y=nCompteurLigne*nPasY
20
21     endFor
22
23     // Remise à zéro
24     pPrise.trsf.x=pPrise.trsf.y=0
25     pDepose.trsf.x=pDepose.trsf.y=0
26
27     waitEndMove()
28
29 endwhile
30
31 end
```

Tache principale Pick And Place

Mouvement de Prise-Dépose

```
Pick_And_Place-Mouvement  X
1  begin
2
3      movej(appro(pPrise,trApproche),tPince,mNomSpeed)
4      movel(pPrise,tPince,mNomSpeed)
5      close(tPince)
6      movel(appro(pPrise,trApproche),tPince,mNomSpeed)
7
8      movej(appro(pDepose,trApproche),tPince,mNomSpeed)
9      movel(pDepose,tPince,mNomSpeed)
10     open(tPince)
11     movel(appro(pDepose,trApproche),tPince,mNomSpeed)
12
13 end
```

Figure 28 : Tâche principale

```
Pick_And_Place-Click_DCY
```

```
1 begin
2
3   bBtnFCY=false
4
5   bDCY=!bDCY
6
7   if bDCY
8     restartMove()
9     taskResume("PickAndPlace",1)
10    sBtnDCY="Pause"
11  else
12    taskSuspend("PickAndPlace")
13    stopMove()
14    sBtnDCY="Reprise"
15  endIf
16
17 end
```

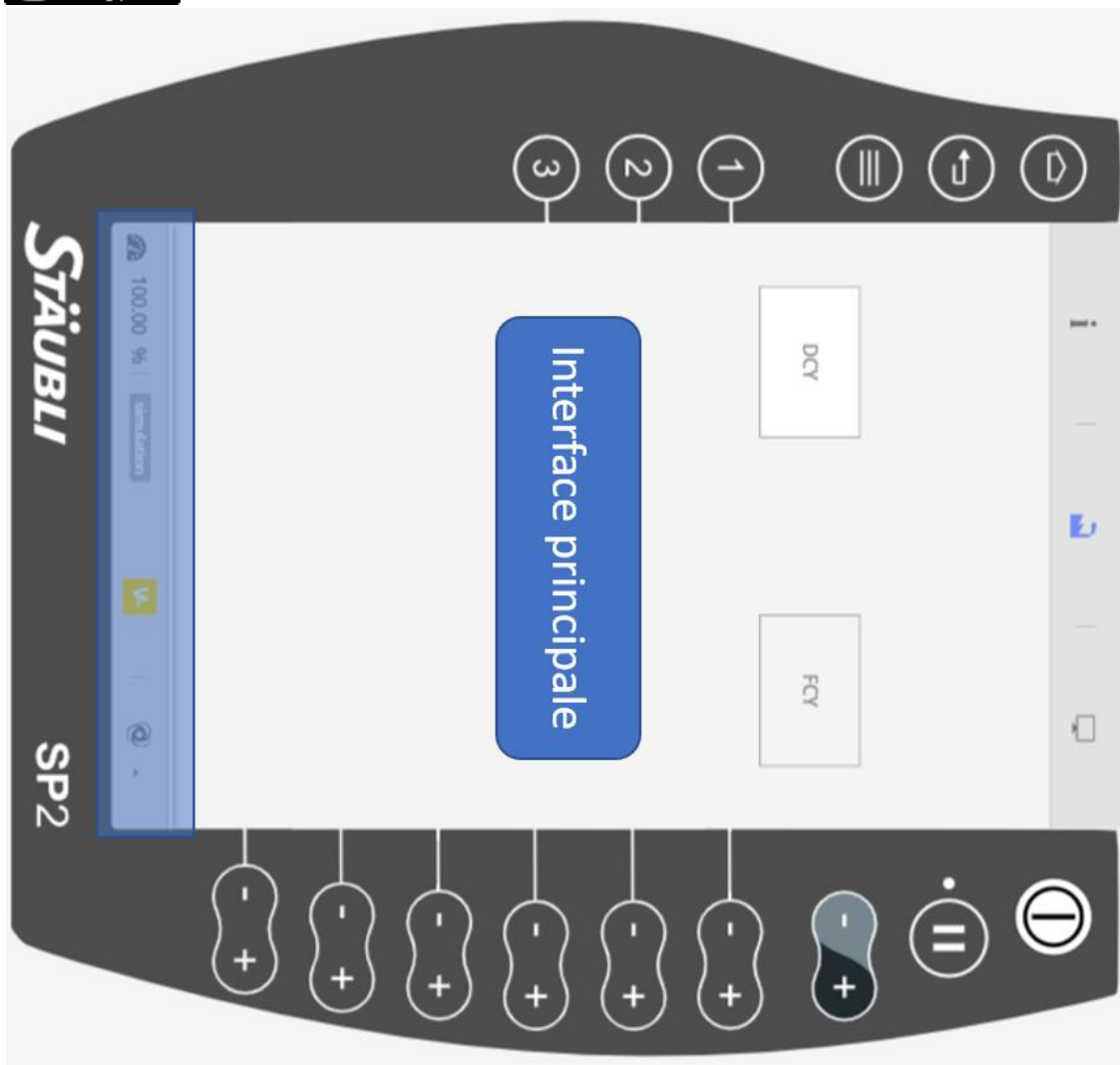
Action liée au bouton DCY

```
Pick_And_Place-Click_FCY
```

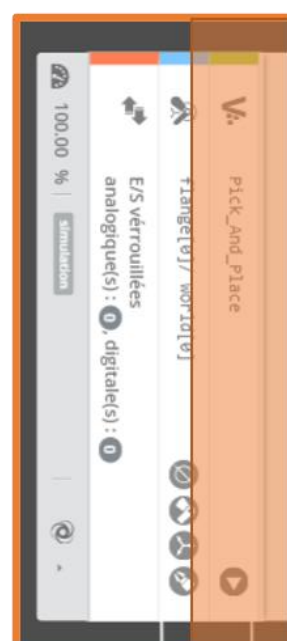
```
1 begin
2
3   bArret=!bArret
4
5   if bArret
6     bBtnFCY=true
7     bBtnDCY=true
8   endIf
9
10 end
```

Action liée au bouton FCY

Figure 29 : Programme liés aux actions des boutons



(re) Démarrage de l'application



Arrêt de l'application

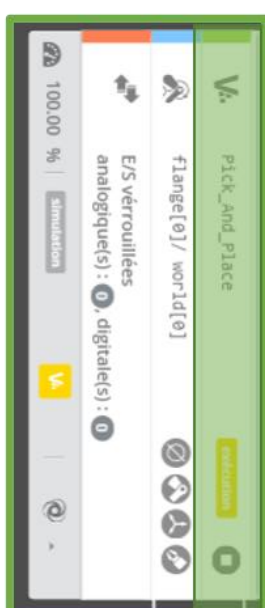


Figure 30 : Vue principale de l'application créée



« Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre du **Programme d'Investissements d'Avenir** portant la référence **ANR-20-NCUN-0009** ».