

Documentation concernant l'Interface Homme-Robot (HRI)

Liste des vues existantes :

- AskAge (ask_age.py)
- AskDrink (ask_drink.py)
- AskName (ask_name.py)
- AskSpeciality (ask_drink.py)
- AskOpenDoor (askOpenDoor.py)
- AskToFollow (ask_to_follow.py)
- AskRoomToClean (ask_room_to_clean.py)
- Confirm (confirm.py)
- ConfirmObjectAction (confirm_object_action.py)
- DisplayInfo (displayInfo.py)
- Find (find.py)
- FindObject (find_object.py)
- FoundAnyone (found_anyone.py)
- FoundGuest (found_guest.py)
- FoundNoObject (found_no_object.py)
- GoTo (goTo.py)
- LookForGuest (look_for_guest.py)
- LookForKnownGuest (look_for_known_guest.py)
- MainMenuPalbator (main_menu_palbator.py)
- ObjectAction (object_action.py)
- OpenDoor (open_door.py)
- PointTo (point_to.py)
- PresentPerson (presentPerson.py)
- SeatGuest (seatGuest.py)
- ShowVideo (showVideo.py)
- Wait (wait.py)

Paramètres génériques pour crée une étape ou sous-étape :

- **Order:** Détermine le numéro de la tâche pour pouvoir les trier
- **Name:** Le texte à afficher sur le panneau des étapes-Action: La vue à lancer(vide si c'est une étape)
- **Eta:** Le temps estimé pour que le robot accomplisse l'étape
- **Id:** Un identifiant de la sous-étape/étape

Paramètres spécifiques pour chaque vue du fichier JSON d'un scénario :

askAge : *Demande d'une saisie vocale ou manuelle de l'âge de l'utilisateur*

- Arguments
 - Who : identifiant de l'utilisateur à qui on demande son âge (ex : « Guest_1 » pour le premier invité)
- Speech
 - Said:message énoncé par le robot à l'utilisateur
 - Title:message affiché sur la vue afin de demander à l'utilisateur de saisir son âge

```
{
  "name": "Ask age",
  "order": 13,
  "eta": 0,
  "speech": {
    "said": "By the way, how old are you?",
    "title": "How old are you?"
  },
  "arguments": {
    "who": "Guest_1"
  },
  "action": "askAge",
  "id": "askinfogl_ask-age"
},
```

askDrink : *Demande d'une saisie vocale ou manuelle de la boisson préférée de l'utilisateur*

- Arguments
 - Who: identifiant de l'utilisateur à qui on demande sa boisson préférée (ex : « Guest_1 » pour le premier invité)
- Speech
 - Said: message énoncé par le robot à l'utilisateur
 - Title: message affiché sur la vue afin de demander à l'utilisateur quelle est sa boisson favorite

```
{
  "name": "Ask drink",
  "order": 11,
  "eta": 0,
  "speech": {
    "said": "Alright {name}, what is your favorite drink?",
    "title": "What's your favorite drink?"
  },
  "arguments": {
    "who": "Guest_1"
  },
  "action": "askDrink",
  "id": "askinfogl_ask-drink"
},
```

askName : Demande d'une saisie vocale ou manuelle de l'âge de l'utilisateur

- Arguments
 - Who: identifiant de l'utilisateur à qui on demande son nom (ex : « Guest_1 » pour le premier invité)
- Speech
 - Said: message énoncé par le robot à l'utilisateur pour lui demander son nom
 - Title: message affiché sur la vue afin de demander à l'utilisateur quelle est son nom

```
{
  "name": "Ask name",
  "order": 9,
  "eta": 0,
  "speech": {
    "said": "Hello, what is your name?",
    "title": "Hello, what's your name?"
  },
  "arguments": {
    "who": "Guest_1"
  },
  "action": "askName",
  "id": "askinfogl_ask-name"
},
```

askOpenDoor : Demande d'ouverture de la porte par l'utilisateur

- Speech
 - said: message énoncé par le robot à l'utilisateur pour lui demander d'ouvrir la porte
 - title: message affiché sur la vue afin de demander à l'utilisateur d'ouvrir la porte
 - description: message indiquant à l'utilisateur ce qu'il doit faire pour continuer le scénario et passer à l'étape suivante

```
{
  "name": "Ask referee to open the door",
  "order": 4,
  "eta": 0,
  "speech": {
    "said": "Hey referee, could you please open the door, and then say Next, or click on the button?",
    "description": "Say Next or click on the button",
    "title": "Please open the door"
  },
  "arguments": {},
  "action": "askOpenDoor",
  "id": "findgl_ask-referee-to-open-the-door"
},
```

askRoom : *Demande de saisie vocale ou manuelle de la pièce à inspecter*

- Arguments
 - What : type de la saisie à réaliser (ex : location)
 - Rooms : liste des pièces disponibles (laisser une chaîne de caractères vide comme sur la photo ci-dessous)
- Speech
 - Said : texte prononcé par le robot
 - Title : texte affiché par le robot

```
{
  "name": "Choose room to clean",
  "order": 4,
  "eta": 0,
  "speech": {
    "said": "In which room do I have to go ?",
    "title": "In which room do I have to go ?"
  },
  "arguments": {
    "what": "location",
    "rooms": ""
  },
  "action": "askRoom",
  "id": "chooseLoc1"
},
```

askToFollow : *Bilan des données apprises concernant l'utilisateur et demande de suivre le robot*

- arguments
 - key : identifiant de l'utilisateur auquel le robot s'adresse (ex : « Guest_1 »)
 - who:
 - name : key + « _name » (ex : « Guest_1_name »)
 - drinkObj :
 - name : key + « _drink » (ex : « Guest_1_drink »)
 - pathOnTablet : key + « _drink_path » (ex : « Guest_1_drink_path »)
 - guestPhotoPath : key + « _face_path » (ex : « Guest_1_face_path »)
 - age : key + « age » (ex : « Guest_1_age »)
 - location: lieu vers lequel on guide l'utilisateur
 - pathOnTablet : chemin de l'image du lieu sur la tablette
 - name : nom du lieu
- Speech
 - said: message énoncé par le robot à l'utilisateur pour lui demander à l'utilisateur de le suivre
 - title: message affiché sur la vue afin de demander à l'utilisateur de le suivre

```
{
  "name": "Ask to follow",
  "order": 15,
  "eta": 0,
  "speech": {
    "said": "Great Guest_1_name, you like Guest_1_drink and you are Guest_1_age years old. Now follow me to the living room please",
    "title": "Follow me to the living room"
  },
  "arguments": {
    "key": "Guest_1",
    "who": {
      "name": "Guest_1_name",
      "drinkObj": {
        "name": "Guest_1_drink",
        "pathOnTablet": "Guest_1_drink_path"
      },
      "guestPhotoPath": "Guest_1_face_path",
      "age": "Guest_1_age"
    },
    "location": {
      "pathOnTablet": "img/locations/livingRoom.png",
      "name": "livingRoom"
    }
  },
  "action": "askToFollow",
  "id": "gotoLr1_ask-to-follow"
},
```

confirm : *Demande de confirmation vocale ou manuelle de la saisie précédemment réalisée*

- Arguments
 - Who: L'utilisateur à qui on demande de confirmer
 - What: Le type de donnée que l'utilisateur doit confirmer (ex : nom, boisson, pièce à inspecter etc..)
- Speech
 - said: message énoncé par le robot à l'utilisateur pour lui demander de confirmer l'information précédemment saisie
 - title: message affiché sur la vue afin de demander de confirmer l'information précédemment saisie

```
{
  "name": "Confirm drink",
  "order": 12,
  "eta": 0,
  "speech": {
    "said": "Did you say your favorite drink was {drink}?",
    "title": "Did you say your favorite drink was {drink}?"
  },
  "arguments": {
    "what": "{drinkId}",
    "who": "Guest_1",
    "timeout": 5
  },
  "action": "confirm",
  "id": "askinfoL1_confirm-drink"
},
```

confirmObjectAction : *Demande de confirmation vocale ou manuelle d'une action réalisée impliquant un objet (ex : demande du robot à l'utilisateur pour ranger un objet à sa place)*

- Arguments
 - What: Identifiant de l'objet (ex : « Object_1 »)

- Object :
 - pathOnTablet : id + « _pathOnTablet » (ex : « Object_1_pathOnTablet »)
 - name : id + « _name » (ex : « Object_1_name »)
- Speech
 - said: message énoncé par le robot à l'utilisateur pour lui demander de confirmer l'information précédemment saisie
 - title: message affiché sur la vue afin de demander de confirmer l'information précédemment saisie

```
{
  "name": "Confirm object taken",
  "order": 12,
  "eta": 0,
  "speech": {
    "said": "Can you confirm that you gave me the Object_1_name ?",
    "title": "Can you confirm that you gave me the Object_1_name ?"
  },
  "arguments": {
    "what": "Object_1",
    "object": {
      "pathOnTablet": "Object_1_pathOnTablet",
      "name": "Object_1_name"
    }
  },
  "action": "confirmObjectAction",
  "id": "chooseLoc1_confirm"
},
```

Find : Recherche d'une chaise libre

- Arguments
 - What : type d'objet à trouver (ex : chair)
 - pathOnTablet : chemin de l'image associée à l'objet recherché
- Speech
 - Said : texte prononcé par le robot
 - Title : texte affiché par le robot

```
{
  "name": "Find empty seat",
  "order": 24,
  "eta": 0,
  "speech": {
    "said": "I am finding an empty chair",
    "title": "I am finding an empty chair"
  },
  "arguments": {
    "what": "chair",
    "pathOnTablet": "img/misc/finding.jpg"
  },
  "action": "find",
  "id": "seatgl_find-empty-seat"
},
```

findObject : Recherche d'un objet dans une pièce spécifique

- Arguments
 - What : identifiant de la pièce à inspecter (ex : ChoosenRoom_1)
 - Location :
 - Name : id pièce + « _name » (ex : ChoosenRoom_1_name)
 - pathOnTablet : id pièce + « _pathOnTablet » (ex : ChoosenRoom_1_pathOnTablet)
 - objectKey : identifiant de l'objet à chercher (ex : Object_1)
- Speech
 - Said : texte prononcé par le robot
 - Title : texte affiché par le robot

```
{
  "name": "Find the first object",
  "order": 8,
  "eta": 0,
  "speech": {
    "said": "I am looking for objects in the ChoosenRoom_1_name",
    "title": "I am looking for objects in the ChoosenRoom_1_name"
  },
  "arguments": {
    "what": "ChoosenRoom_1",
    "location": {
      "pathOnTablet": "ChoosenRoom_1_pathOnTablet",
      "name": "ChoosenRoom_1_name"
    },
    "objectKey": "Object_1"
  },
  "action": "findObject",
  "id": "findObj1"
},
```

foundAnyone : Information à l'utilisateur que personne n'a été détecté par le robot

- Speech
 - Said : texte prononcé par le robot pour expliquer qu'il n'a détecté personne
 - Title : texte affiché sur la tablette

```
{
  "name": "NotFoundG1",
  "order": 7,
  "eta": 0,
  "speech": {
    "said": "I didn't find anyone",
    "title": "I didn't find anyone"
  },
  "arguments": {
  },
  "action": "foundAnyone",
  "id": "findg1_notfound"
},
```


foundGuest : Information à l'utilisateur qu'une personne a été détectée et affichage de sa photo

- Arguments
 - Who : identifiant du Guest détecté (ex : Guest_1)
 - guestPhotoPath : id Guest + « _face_path » (ex : Guest_1_face_path »)
- Speech
 - Said : texte prononcé par le robot
 - Title : texte affiché par le robot

```
{
  "name": "Found guest",
  "order": 6,
  "eta": 0,
  "speech": {
    "said": "I have found a new guest",
    "title": "I have found a new guest"
  },
  "arguments": {
    "who": "Guest_1",
    "guestPhotoPath": "Guest_1_face_path"
  },
  "action": "foundGuest",
  "id": "foundG1"
},
```

goTo : Déplacement vers un lieu spécifique

- Arguments
 - where : identifiant du lieu à atteindre (ex : ChoosenRoom_1)
 - location:
 - pathOnTablet : where + « _pathOnTablet » (ex : "ChoosenRoom_1_pathOnTablet")
 - name : where + "_name" (ex: ChoosenRoom_1_name)
- Speech
 - said: message énoncé par le robot afin d'indiquer le lieu vers lequel celui-ci se déplace
 - title: message affiché sur la vue afin d'indiquer le lieu vers lequel le robot se déplace

```
{
  "name": "Travel to room",
  "order": 6,
  "eta": 0,
  "speech": {
    "said": "I am travelling to the ChoosenRoom_1_name",
    "title": "I am travelling to the ChoosenRoom_1_name"
  },
  "arguments": {
    "where": "ChoosenRoom_1",
    "location": {
      "pathOnTablet" : "ChoosenRoom_1_pathOnTablet",
      "name": "ChoosenRoom_1_name"
    }
  },
  "action": "goTo",
  "id": "goTo1"
},
```


lookForGuest : *Recherche d'une personne*

- Arguments
 - Who : Identifiant du guest à détecter (ex : Guest_1)
- Speech
 - said : texte prononcé par le robot pour avertir qu'il cherche un nouveau guest
 - title : texte affiché sur la tablette

```
{
  "name": "LookG1",
  "order": 5,
  "eta": 0,
  "speech": {
    "said": "I am looking for a new guest",
    "title": "I am looking for a new guest"
  },
  "arguments": {
    "who": "Guest_1"
  },
  "action": "lookForGuest",
  "id": "findg1_look"
},
```

lookForKnownGuest : *Recherche d'un invité déjà connu*

- Arguments
 - Key : identifiant du Guest (ex : Guest_1)
 - Who :
 - Name : id Guest + « _name » (ex: Guest_1_name)
 - guestPhotoPath : id Guest + « _face_path » (ex: Guest_1_face_path)
- Speech
 - Said : texte prononcé par le robot
 - Title : Texte affiché par le robot

```
{
  "name": "LookG1ForPresent",
  "order": 21,
  "eta": 0,
  "speech": {
    "said": "I am looking for Guest_1_name",
    "title": "I am looking for Guest_1_name"
  },
  "arguments": {
    "key": "Guest_1",
    "who": {
      "name": "Guest_1_name",
      "guestPhotoPath": "Guest_1_face_path"
    }
  },
  "action": "lookForKnownGuest",
  "id": "findG1_look_for_present"
},
```

objectAction : *Action concernant un objet spécifique (grasping, rangement, détection etc..)*

- Arguments
 - What : identifiant de l'objet (ex : Object_1)
 - Object :
 - pathOnTablet : id Object + « _pathOnTablet » (ex : Object_1_pathOnTablet)
 - name : id Object + « _name » (ex: Object_1_name)
 - help_mode : False si le robot a un comportement normal, True si le robot a un comportement dégradé (exemple : besoin d'aide pour attraper un objet)
- Speech
 - Said : texte prononcé par le robot
 - Title : texte affiché par le robot

```
{
  "name": "Found the first object",
  "order": 9,
  "eta": 0,
  "speech": {
    "said": "I have found the Object_1_name",
    "title": "I have found the Object_1_name"
  },
  "arguments": {
    "what": "Object_1",
    "object": {
      "pathOnTablet": "Object_1_pathOnTablet",
      "name": "Object_1_name"
    },
    "help_mode": false
  },
  "action": "objectAction",
  "id": "foundObj1"
},
```

openDoor : *Ouverture de la porte*

- Speech
 - Said : texte prononcé par le robot
 - Title : texte affiché par le robot

```
{
  "name": "Open door",
  "order": 2,
  "eta": 0,
  "speech": {
    "said": "I will open the door",
    "title": "I will open the door"
  },
  "action": "openDoor",
  "id": "openDoor1"
},
```

pointTo : *Pointage d'un objet par le robot*

- Arguments
 - What : type d'objet à pointer (ex : chair)
 - pathOnTablet : chemin de l'image associée à l'objet pointé
- Speech
 - Said : texte prononcé par le robot
 - Title : texte affiché par le robot

```
{
  "name": "Point to empty seat",
  "order": 25,
  "eta": 0,
  "speech": {
    "said": "I am pointing an empty chair",
    "title": "I am pointing an empty chair"
  },
  "arguments": {
    "what": "chair",
    "pathOnTablet": "img/misc/chair.png"
  },
  "action": "pointTo",
  "id": "seatgl_point-to-empty-seat"
},
```

presentPerson : *Présentation d'un nouvel invité aux autres convives*

- Arguments
 - Who: *Personne à présenter*
 - name : identifiant de la personne à présenter (ex : Guest_2)
 - drinkObj :
 - name : id Guest + « _drink » (ex: Guest_2_drink)
 - pathOnTablet : id Guest + "_drink_path" (ex: Guest_2_drink_path)
 - guestPhotoPath : id Guest + "_face_path" (ex: Guest_2_face_path)
 - age : id Guest + "_age" (ex: Guest_2_age)
 - To (array): *Liste des personnes à qui on présente le nouvel invité*
La structure ci-dessous est à répéter pour chaque personne
 - name : id Personne (ex : Guest_1)
 - drink :
 - name : id Personne + « _drink » (ex: Guest_1_drink)
 - pathOnTablet : id Personne + « _drink_path" (ex: Guest_1_drink_path)
 - guestPhotoPath : id Personne + "_face_path" (ex: Guest_1_face_path)
 - age : id Personne + "_age" (ex: Guest_1_age)
- Speech
 - said: message énoncé par le robot à l'utilisateur pour présenter les 2 personnes

```
{
  "name": "Say name and drink",
  "order": 46,
  "eta": 0,
  "speech": {
    "said": "Hey Guest_1_name and John_name, here is Guest_2_name, whose favorite drink is Guest_2_drink. Guest_2_name is Guest_2_age."
  },
  "arguments": {
    "to": [
      {
        "name": "Guest_1_name",
        "drink": {
          "name": "Guest_1_drink",
          "pathOnTablet": "Guest_1_drink_path"
        },
        "guestPhotoPath": "Guest_1_face_path",
        "age": "Guest_1_age"
      },
      {
        "name": "John_name",
        "drink": {
          "name": "John_drink",
          "pathOnTablet": "John_drink_path"
        },
        "guestPhotoPath": "John_face_path",
        "age": "John_age"
      }
    ],
    "who": {
      "name": "Guest_2_name",
      "drinkObj": {
        "name": "Guest_2_drink",
        "pathOnTablet": "Guest_2_drink_path"
      },
      "guestPhotoPath": "Guest_2_face_path",
      "age": "Guest_2_age"
    }
  },
  "action": "presentPerson",
  "id": "introduce2toothers_say-name-and-drink"
},
```

seatGuest.py : *Information à l'utilisateur qu'il peut s'asseoir dans le siège libre détecté*

- Arguments
 - Who:
 - Name : Id Guest + « _name » (ex: Guest_2_name)
 - guestPhotoPath : Id Guest + "_face_path" (ex: Guest_2_face_path)
- Speech
 - said: message énoncé par le robot à l'utilisateur pour lui proposer de s'asseoir
 - title: message affiché sur la vue afin de proposer à l'utilisateur de s'asseoir

```
{
  "name": "Tell first guest to seat",
  "order": 56,
  "eta": 0,
  "speech": {
    "said": "Please have a seat Guest_2_name.",
    "title": "Please have a seat Guest_2_name."
  },
  "arguments": {
    "who": {
      "name": "Guest_2_name",
      "guestPhotoPath": "Guest_2_face_path"
    }
  },
  "action": "seatGuest",
  "id": "seatg2_tell-first-guest-to-seat"
},
```

showVideo.py : *Affichage d'une vidéo sur la tablette*

- Arguments
 - videoId : Identifiant de la vidéo
 - videoPath : Laisser ce champ vide, il sera modifié grâce aux données fournies par le GeneralManager
 - videoFormat : Laisser ce champ vide, il sera modifié grâce aux données fournies par le GeneralManager
- Speech
 - said: message énoncé par le robot à l'utilisateur
 - title: message affiché en titre de la vue
 - description : message affiché dans le corps du texte de la vue

```
{
  "name": "Show video test",
  "order": 14,
  "eta": 0,
  "speech": {
    "said": "I will display a video for you",
    "title": "video testing",
    "description": "Video testing implementation for tablet"
  },
  "arguments": {
    "videoId": "takeBag",
    "videoPath": "",
    "videoFormat": ""
  },
  "action": "showVideo",
  "id": "showVideo1"
},
```

wait.py : *Attente d'un temps spécifique avec un compte à rebours*

- Arguments
 - time: temps pendant lequel la vue doit attendre
- Speech
 - said: message énoncé par le robot à l'utilisateur pour lui dire d'attendre
 - title: message affiché sur la vue

```
{
  "name": "Wait",
  "order": 3,
  "eta": 0,
  "speech": {
    "said": "I am waiting for the referee",
    "title": "I'm waiting for the referee"
  },
  "arguments": {
    "time": 3
  },
  "action": "wait",
  "id": "findgl_wait"
},
```