



Holy Spirit University of Kaslik (USEK)  
School of Engineering

# PEPPER IN A COMMENCEMENT CEREMONY GIN456

Prepared By: Maria El Murr  
Paul Chbat  
Anthony Massaad  
Elio El Khoury  
Georges Chamoun

Presented to: **Br. Elie Saad**





# TABLE OF CONTENTS

• Introduction	01
• Personas	02
• Use cases	03
• Challenges	04
• Future Improvements	05
• Conclusion	06

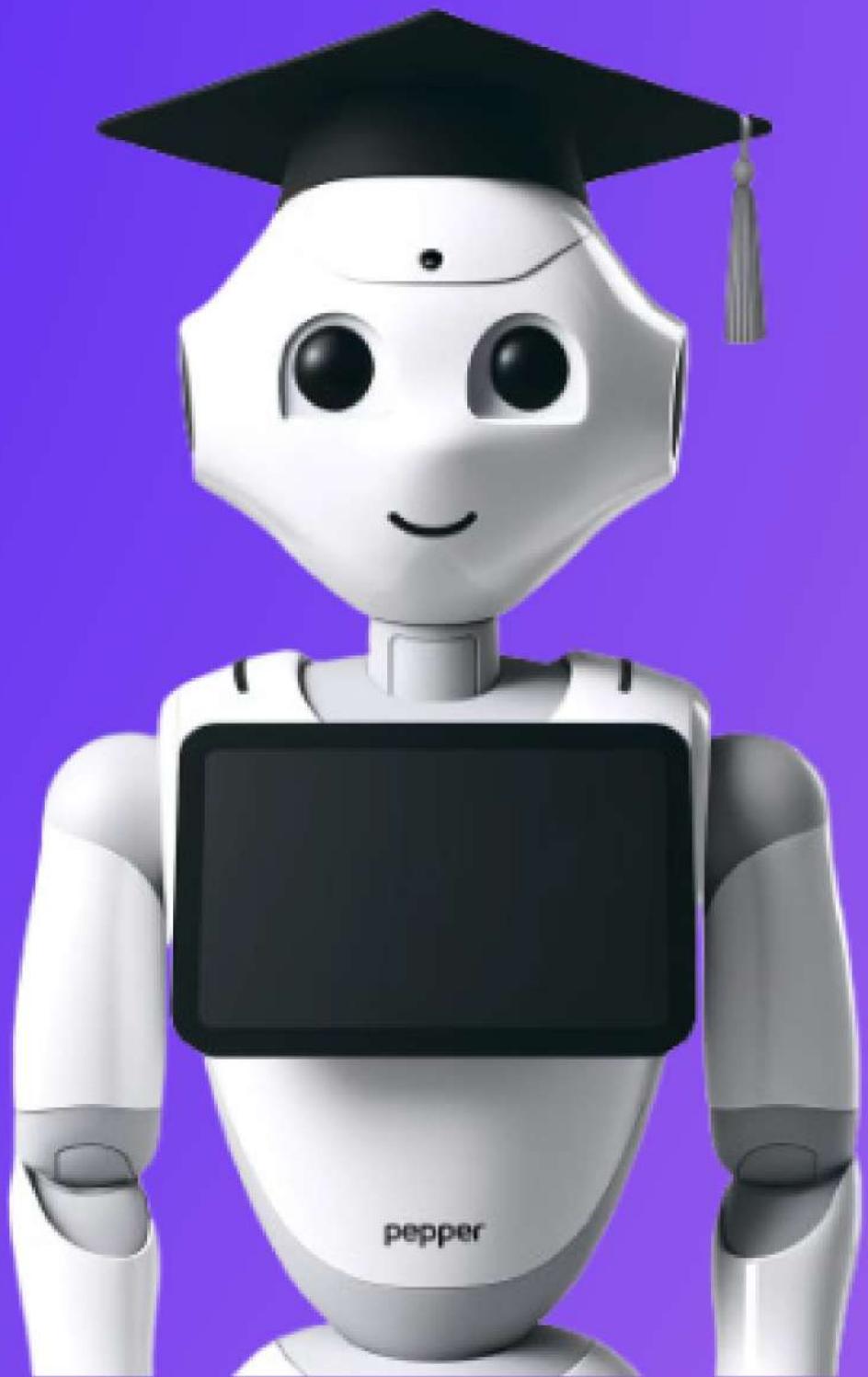


# INTRODUCTION

**Welcome to the future of graduation ceremonies!**

Pepper will enhance the guest experience at the graduation ceremony by providing seamless interaction and helpful services. Here's how, Pepper will:

- Allow guests to choose between english or french language.
- Explain the ceremony rules.
- Scan QR code provided by the guests and informing them about the zone name where they are seated.
- Direct the guests by highlighting the zone.
- And finally, take a selfie with the guests.



# PERSONAS



MARIE-JEANNE  
PSYCHOLOGIST



JEAN-PHILIPPE  
15 YO STUDENT



RAYMOND  
SALES MANAGER



CESAR  
SOFTWARE  
ENGINEER

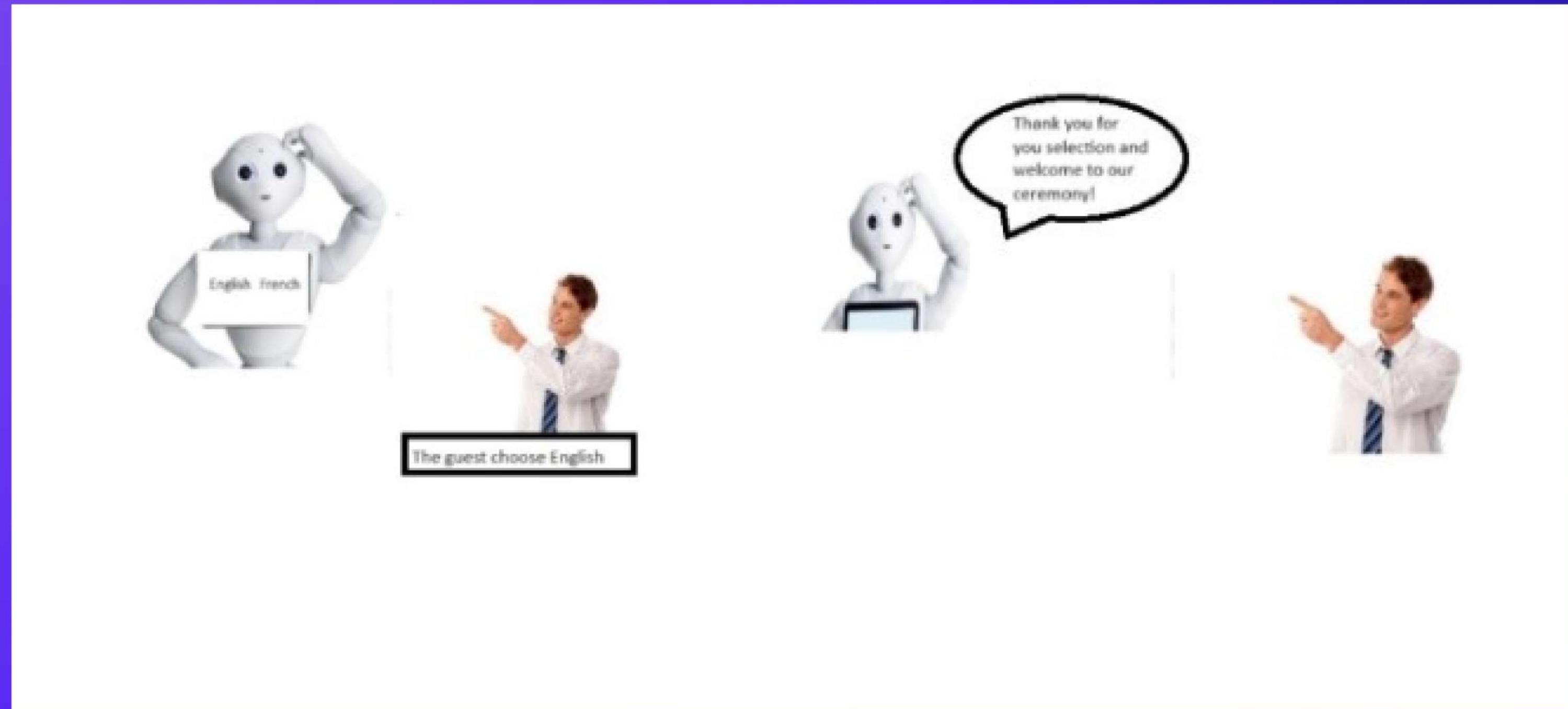


# SELECT THE LANGUAGE

Here's how Pepper will allow guests to select their preferred language for interaction:

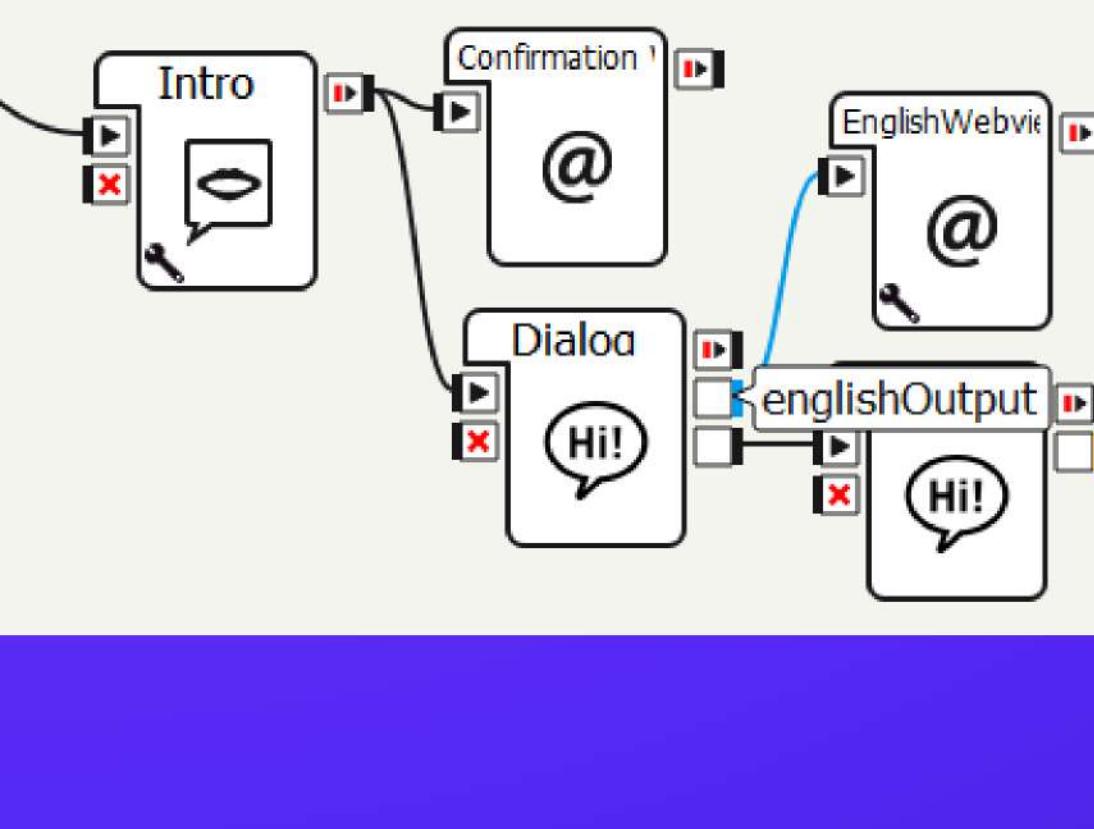
- Pepper greets the guest with a friendly wave and displays two language options (English and French) on the screen.
- If the guest selects French, Pepper will politely inform them that it does not have access to that language and will continue the interaction in English.
- The robot greets the guest and asks for some information about them.

# STORY BOARD OF SELECTING A LANGUAGE



# IMPLEMENTATION USING CHOREOGRAPHY

- The Intro state marks the start of the flow, during which an introduction will be presented.
- When transitioning from the Intro, Confirmation is set to activate the Dialog state, where a dialog interaction occurs, potentially interacting with the user when selecting the language.
- When the user confirms in the language ,englishOutput is set to value 1, transitioning to the English\_Dialog state where an English dialog interaction takes place.



```
# Set page details
pageImage = "pepper-question.png"
pageHeading = "You now have to decide"
pageText = "Please select an option"

# Set the URL
url = "http://{ip}/{appName}/pages/confirmation.html?pageHeading={pageHeading}&pageText={pageText}&pageImage={pageImage}".format(ip=robotIP, appName=appName, pageHeading=pageHeading, pageText=pageText, pageImage=pageImage)

self.logger.info(url)

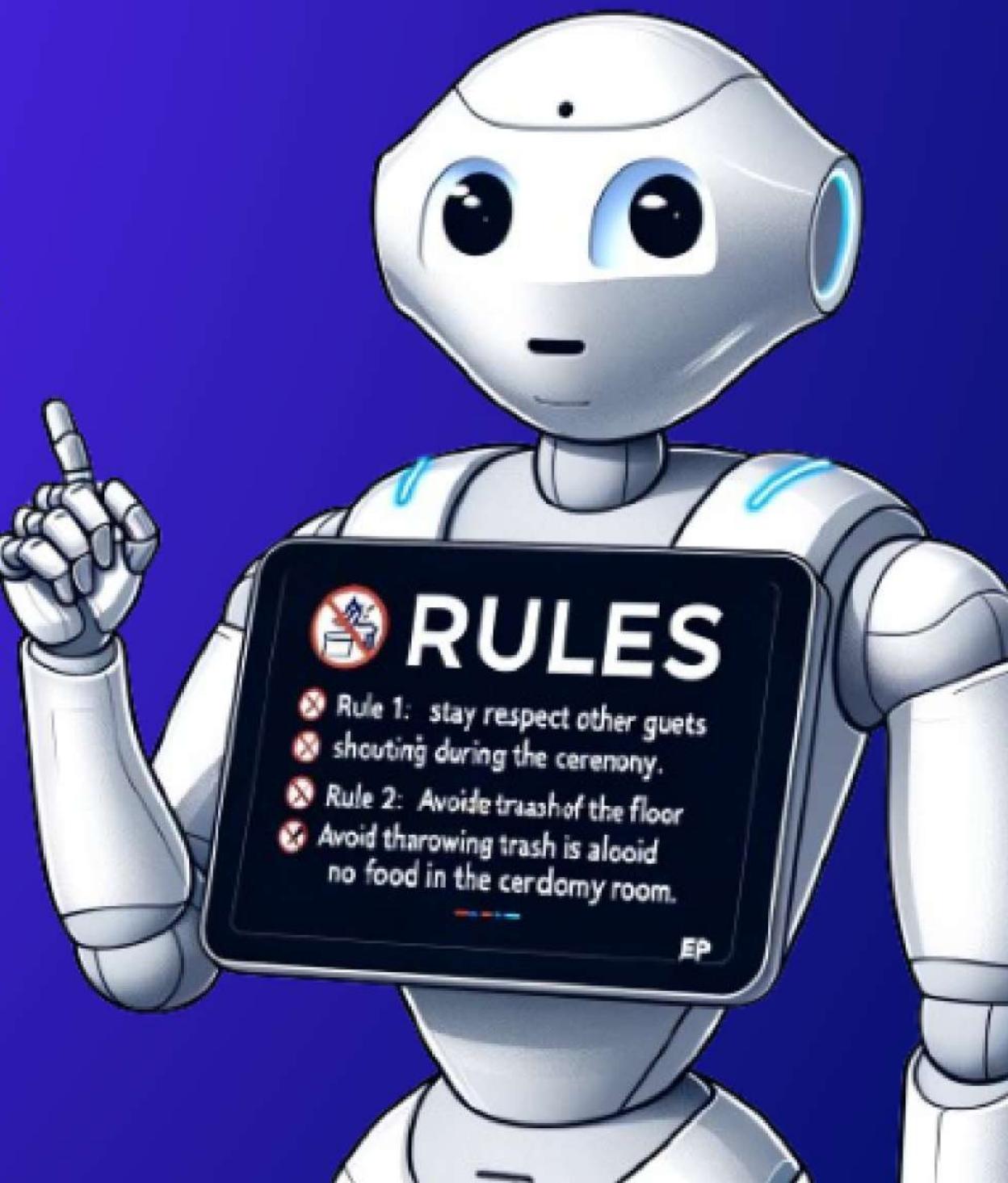
# Show the web view on the tablet
tabletService.showWebview(url)
else:
    self.logger.warning("Couldn't find tablet service, so can't set application: %s" % appName)

# self.onStopped()
```

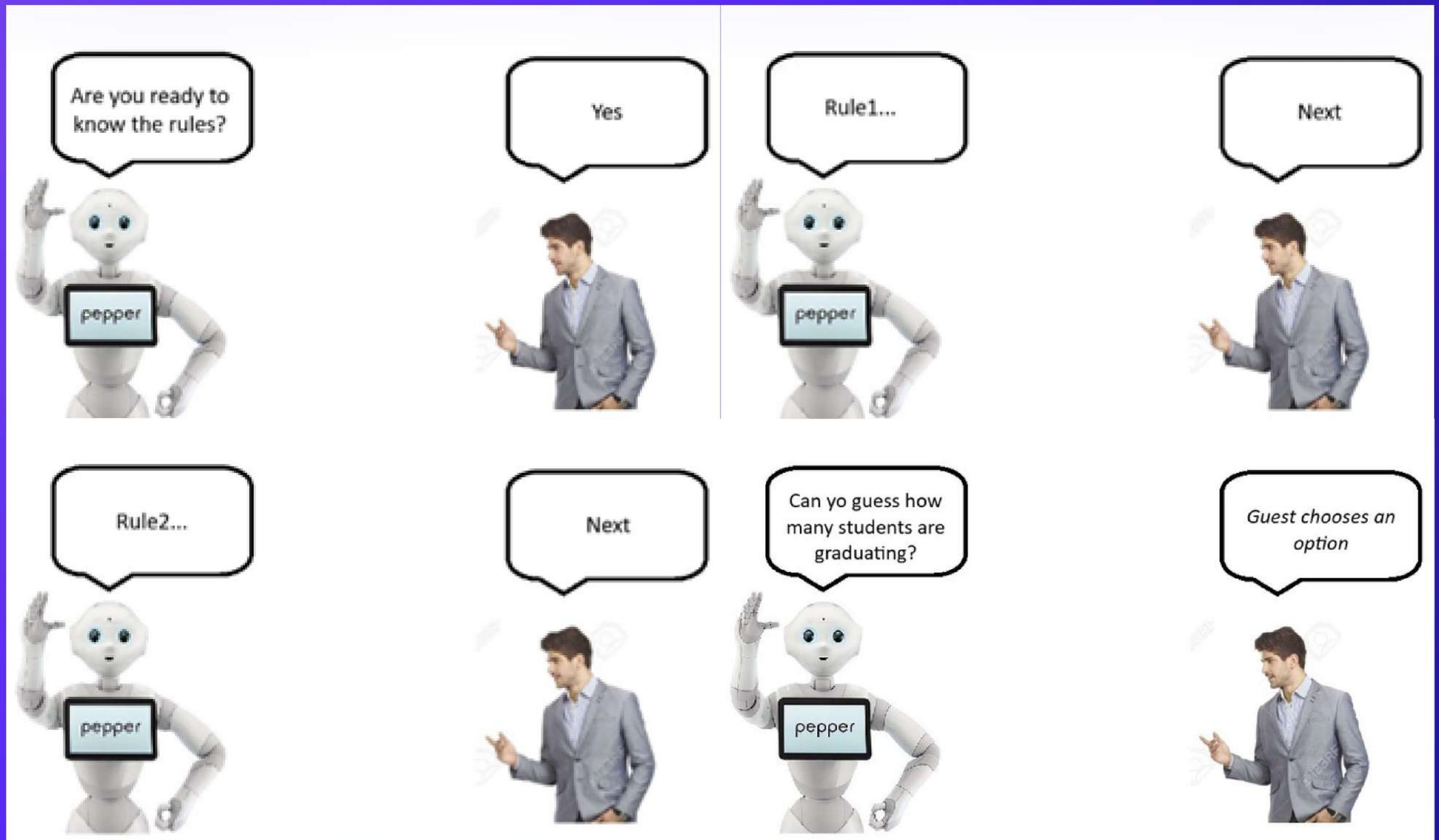
# STATE RULES OF THE CEREMONY

Here's how Pepper will engage with guests to ensure everyone is informed about the ceremony rules:

- Pepper asks the guest if they are ready to learn about the rules.
- Upon the guest's confirmation, Pepper starts by stating the first rule.
- The guest can prompt Pepper to continue by saying "next" or pressing the 'next' button.
- After the rules, Pepper shares a quick, interesting fact about the graduation ceremony to enhance the guest's experience.

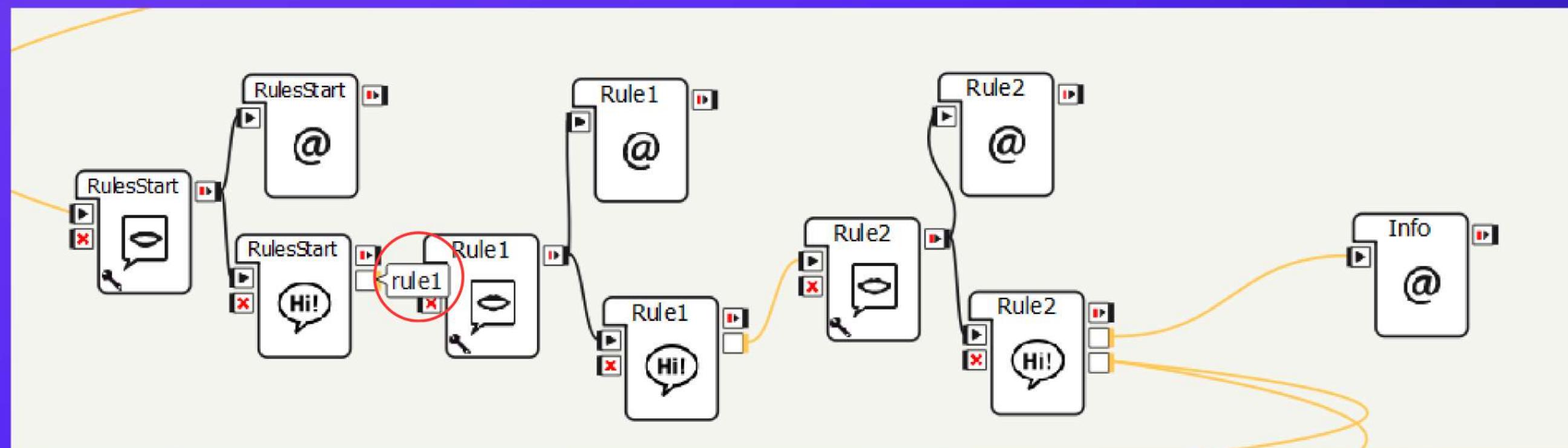


# STORY BOARD OF STATING THE RULES



# IMPLEMENTATION USING CHOREOGRAPHY

- The flow begins with RulesStart, where Pepper asks the guests if he wants to know about the ceremony rules, while displaying an image on the tablet.
- When the guest responds with yes, rule1 is set 1 to activate rule 1 dialog where Pepper will say rule 1 and also display on his table the rule with an image.
- When the guest presses the button “Next” a confirmation event will be raised setting rule 2 to value 1... and so on.
- At the end, will ask the guest a trivia question before finishing.



# IMPLEMENTATION USING CHOREOGRAPHY

This is a closer look for example for the rule 1 dialog:

```
topic: ~rule1()
language: enu

proposal: %rule1prop Stay respectful to other guests attending this ceremony, and avoid shouting while the ceremony is underway.\pau=1000\
    u1:(next) $rule2=1 $onStopped=1

u:(e:onStart) Rule 1 \pau=250\ ^gotoReactivate(rule1prop)
u:(e:rule1next) $rule2=1 $onStopped=1
```

On start, we reactivate the proposal named “rule1prop”.

If the guest decides to say next it will be caught by : **u1:(next) \$rule2=1 \$onStopped=1**

If the guest decides to press the button the event will be caught by: **u:(e:rule1next) \$rule2=1 \$onStopped=1**

The rest of the dialog boxes are implemented in a similar way.

# IMPLEMENTATION USING CHOREOGRAPHY

This is a closer look for displaying text and image on the tablet:

```
if appName:  
    if tabletService:  
        # use default robot IP address from the tablet  
        robotIP = "198.18.0.1" # tabletService.getRobotIp()  
  
        # Set page details  
        pageHeading = "Rule 1"  
        pageText = "Stay respectful to other guests attending this ceremony, and avoid shouting while the ceremony is underway."  
        pageImage = "rule1.png"  
        # Set the URL  
        url = "http://{ip}/apps/{appName}/pages/rule1.html?pageHeading={pageHeading}&pageText={pageText}  
&pageImage={pageImage}".format(ip=robotIP, appName=appName, pageHeading=pageHeading, pageText=pageText, pageImage=pageImage)  
  
        self.logger.info(url)
```

We are passing pageHeading, pageText and pageImage as parameters to the url containing the path to rule1.html page

The rest of the webview boxes are implemented in a similar way.

# IMPLEMENTATION USING CHOREOGRAPHY

In the trivia question page we are using only pageText and pageImage so to be able to pass them dynamically just like before we modify the JavaScript function in this way:

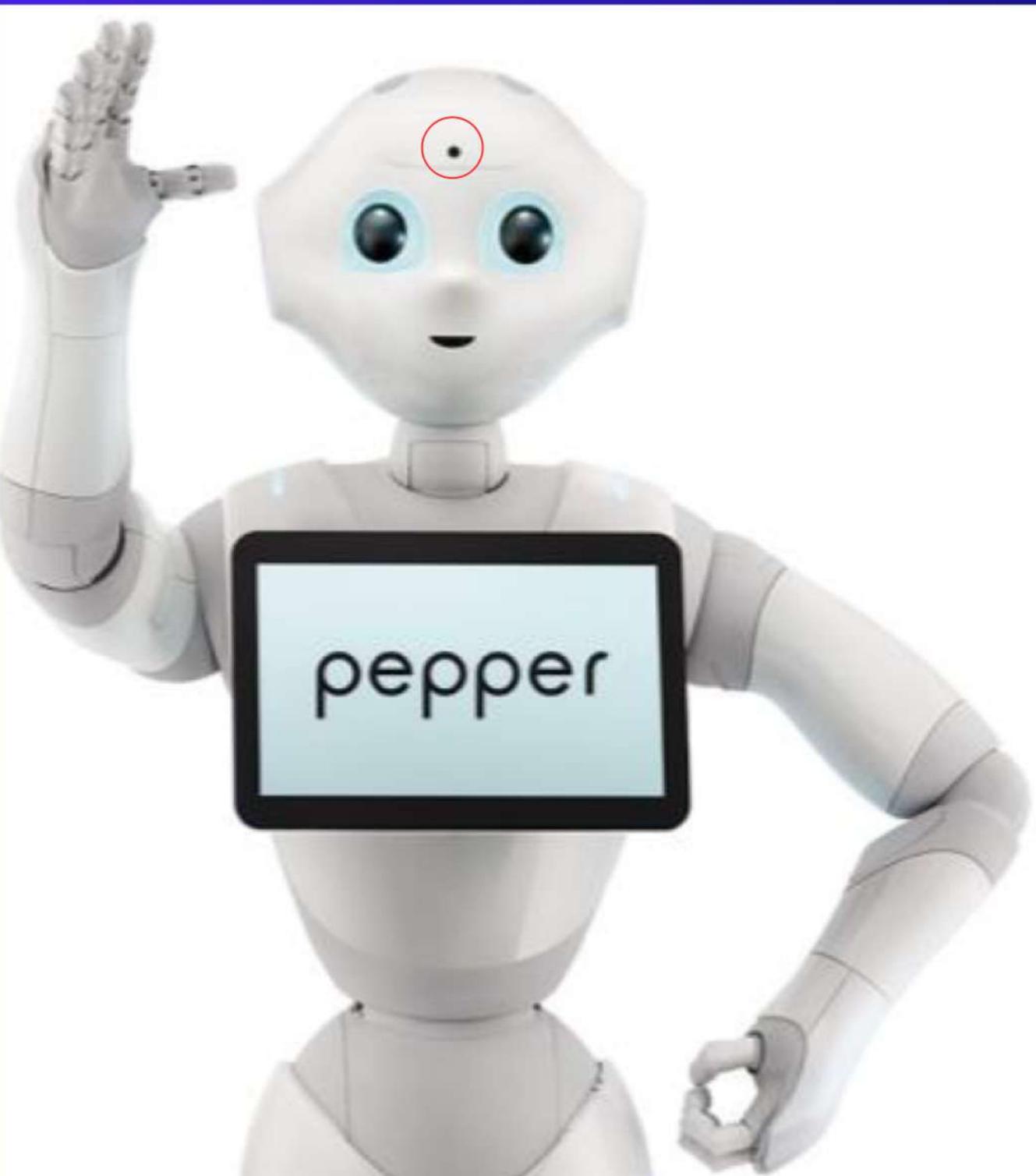
```
function displayTextImageInformation() {
    document.getElementById("pageText").innerHTML = decodeURI(getUrlParam("pageText", ""));
    imageName = getUrlParam("pageImage", "");
    if (imageName) {
        document.getElementById("pageImage").src = "../pics/" + imageName;
    }
}
```

This function fetches the parameters passed by URL and forwards them to the html elements in the webpage.

# SCAN QR CODE:

In this part, the robot will scan the QR code of the guest to enter the ceremony:

- First, the Pepper robot will guide the guest on where to place their QR code.
- Then, the guest will show the robot their QR code using the camera located on its head.
- After displaying the QR code, the guest will present the invitation card corresponding to their entry and provide some information about themselves.

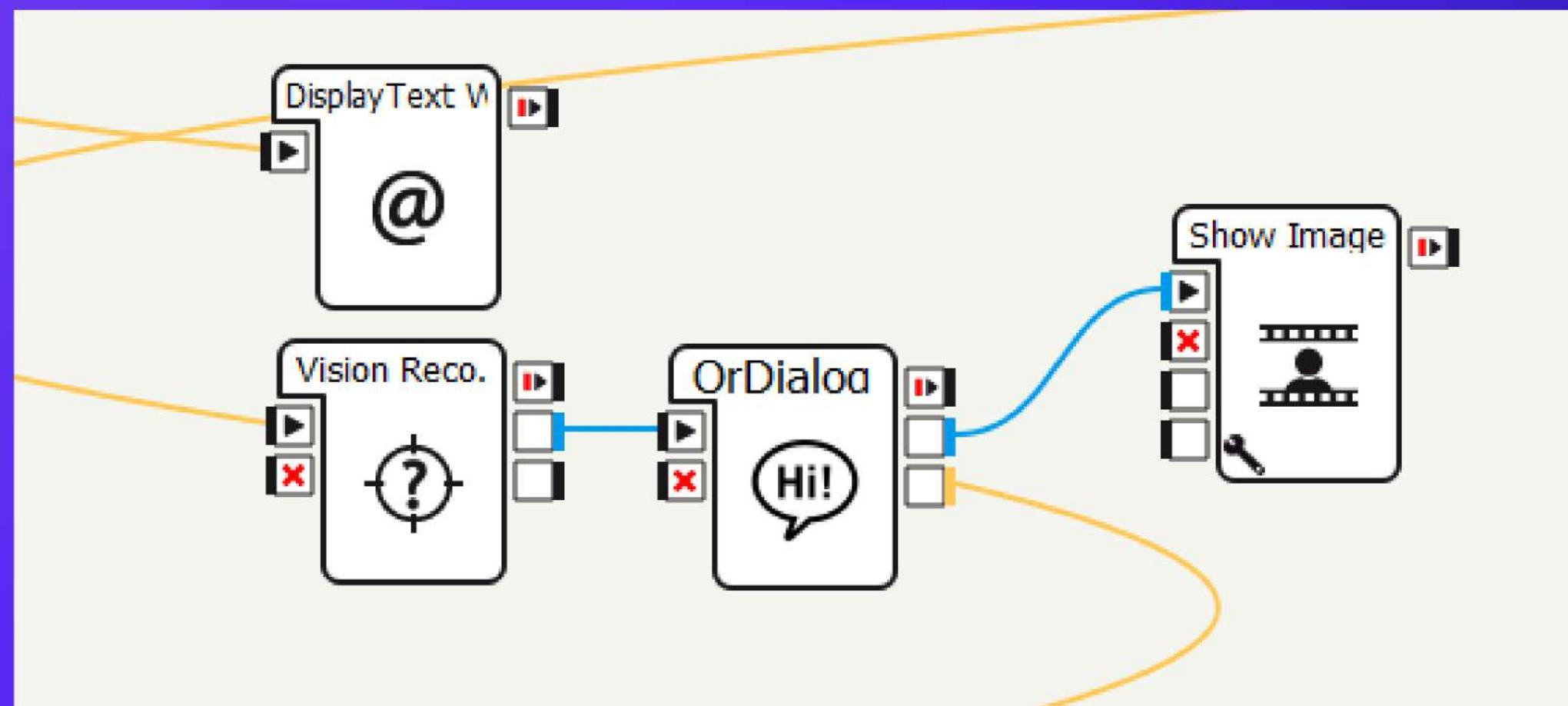


# STORY BOARD OF SCANING QR CODE



# IMPLEMENTATION USING CHOREOGRAPHY

- The flow begins with a webpage displaying instructions on how to scan the QR code.
- After the guest displays the QR code, the robot takes it as input and welcomes the guest in the "QRDialog", then displays the invitation card using the "show Image" box.



# IMPLEMENTATION USING CHOREOGRAPHY

This is the vision recognition code:

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self, False)

    def onLoad(self):
        self.nPicturesDetected = -1

    def onUnload(self):
        #puts code for box cleanup here
        pass

    def onInput_onStart(self, p):
        self.memory_proxy = self.session().service("ALMemory") #create a session
        if(len(p) > 1):
            self.nPicturesDetected = len(p[1])
            labels = []
            for s in p[1]:
                labels.append(s[0])
            self.onPictureLabel(labels)
            self.memory_proxy.raiseEvent("PictureLabel", s[0][0]) #raise for the event
            self.logger.info("event picture label raised :" + s[0][0])# show event in logs
        else:
            if(self.nPicturesDetected != 0):
                self.nPicturesDetected = 0
                self.onNoPicture()
```

```
topic: ~QrDialog()
language: enu

proposal: %Scan1 $showImage=maria.jpg welcome to the graduation \pau=2500\ $direct=1
proposal: %Scan2 $showImage=elio.jpg welcome to the graduation \pau=2500\ $direct=1
proposal: %Scan3 $showImage=anthony.jpg welcome to the graduation \pau=2500\ $direct=1
proposal: %Scan4 $showImage=paul.jpg welcome to the graduation \pau=2500\ $direct=1
proposal: %Scan5 $showImage=georges.jpg welcome to the graduation \pau=2500\ $direct=1

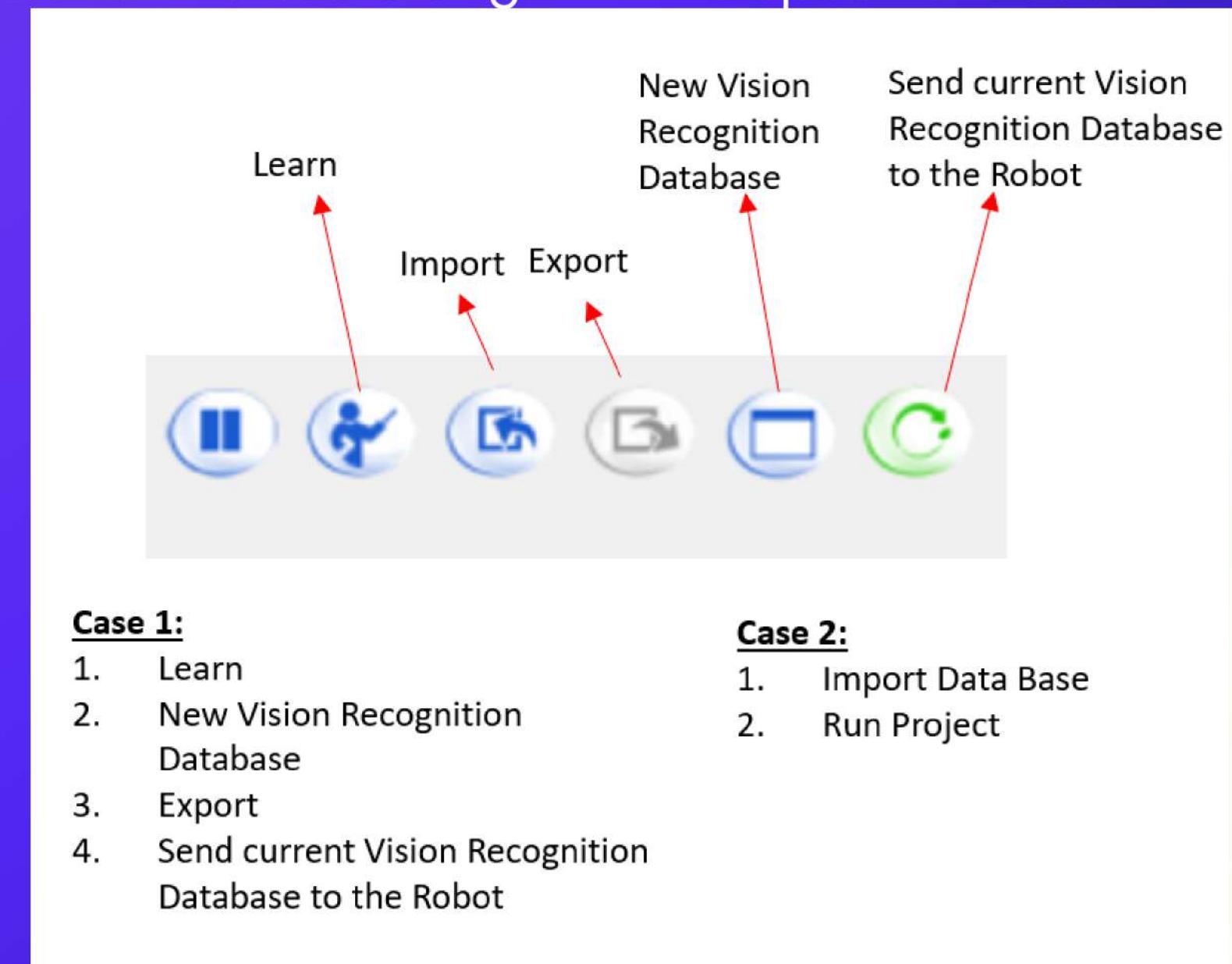
u: (e:PictureLabel) ^gotoReactivate($PictureLabel)

def onInput_showImage(self, url):
    # We create TabletService here in order to avoid
    # problems with connections and disconnections of the tablet during the life of the application
    tabletService = self._getTabletService()
    if tabletService:
        try:
            url = self.getParameter("ImageUrl")
            if url == '':
                self.logger.error("URL of the image is empty")
            if not url.startswith('http'):
                url = self._getAbsoluteUrl(url)
            tabletService.showImage(url)
        except Exception as err:
            self.logger.error("Error during ShowImage : %s" % err)
            self.onStopped()
    else:
        self.logger.warning("No ALTabletService, can't display the image.")
        self.onStopped()
```

# IMPLEMENTATION USING CHOREOGRAPHY

## Creation of Databases:

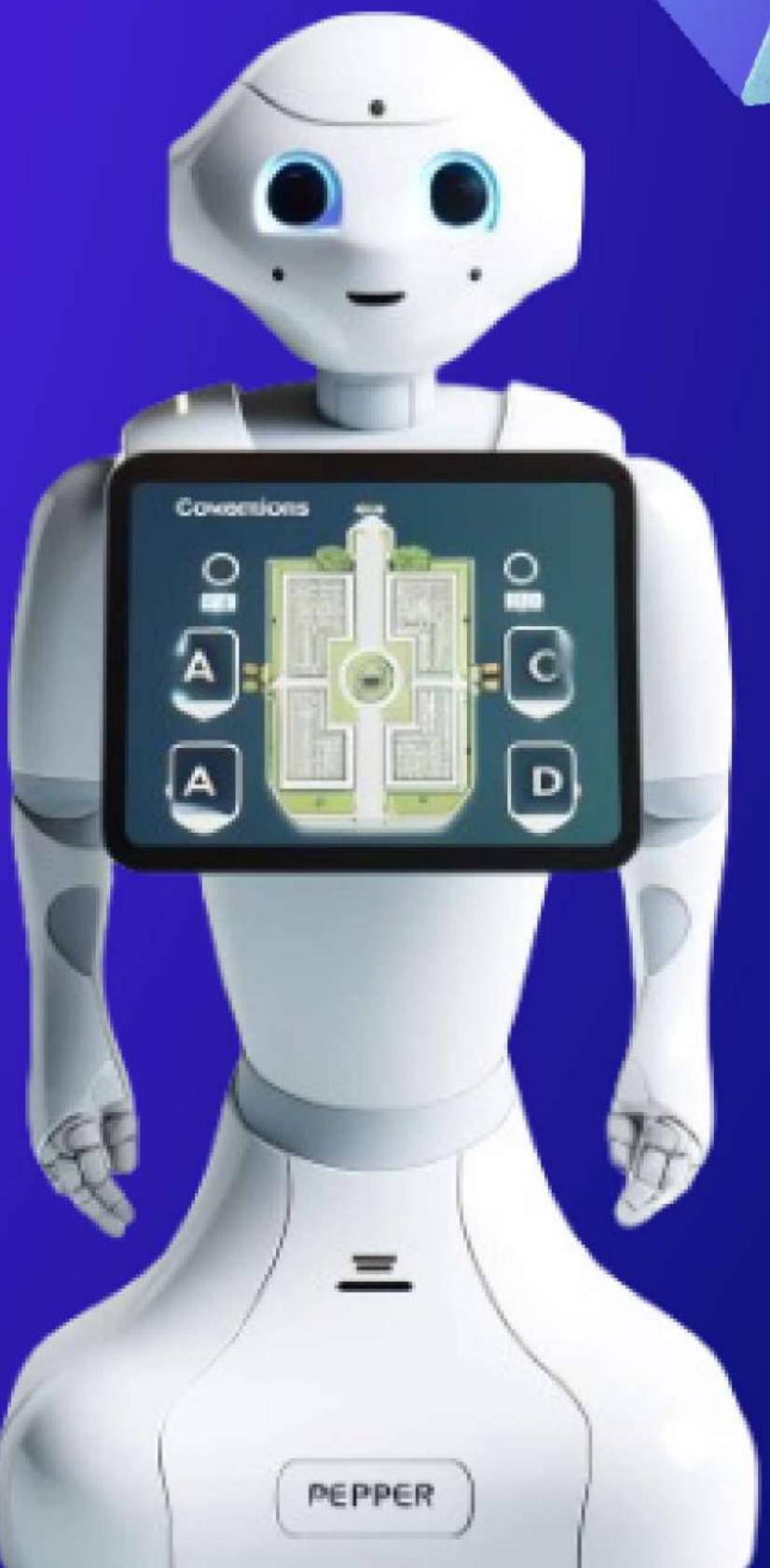
In this class, we process input to extract picture labels. Assuming the input contains a list of picture labels, we directly access the first label. We then raise an event named "PictureLabel" with this label. Finally, we log a message indicating the event raised along with the specific label.



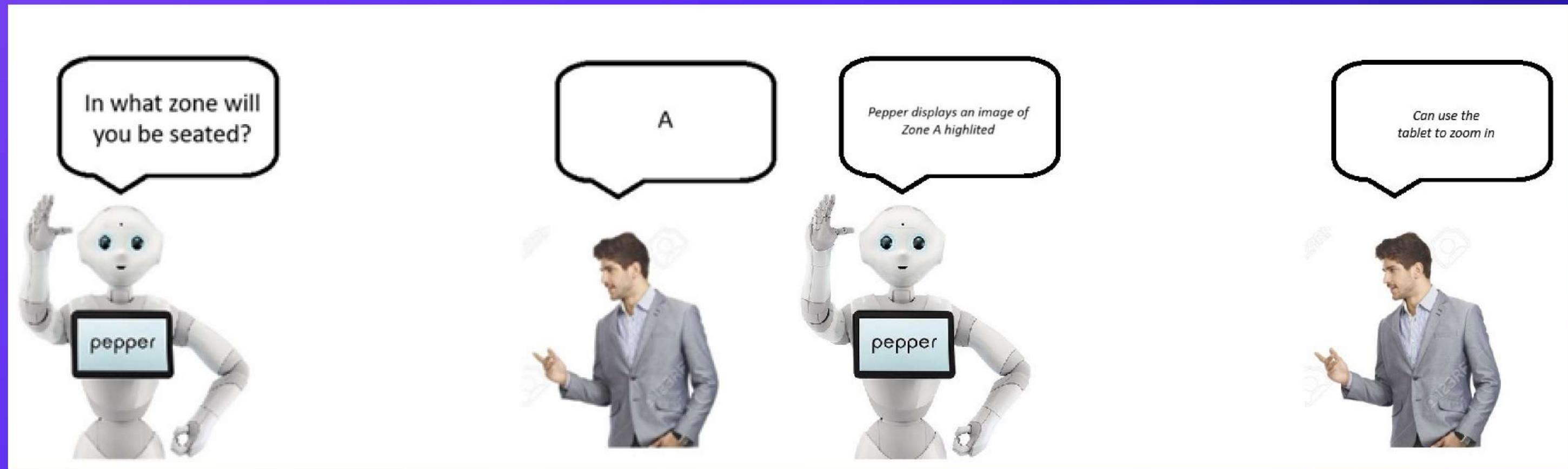
# DIRECT GUESTS

Here's how Pepper directs guests to their designated zones:

- Pepper shows an image of the ceremony hall on its tablet, equipped with four buttons labeled A, B, C, and D, corresponding to the different zones in the hall.
- Pepper asks the guest, "In which zone are you seated?" This invites the guest to interact either verbally or by pressing one of the labeled buttons on the display.
- Depending on the guest's response or button pressed, Pepper will then display the ceremony hall image again, this time highlighting the selected zone.

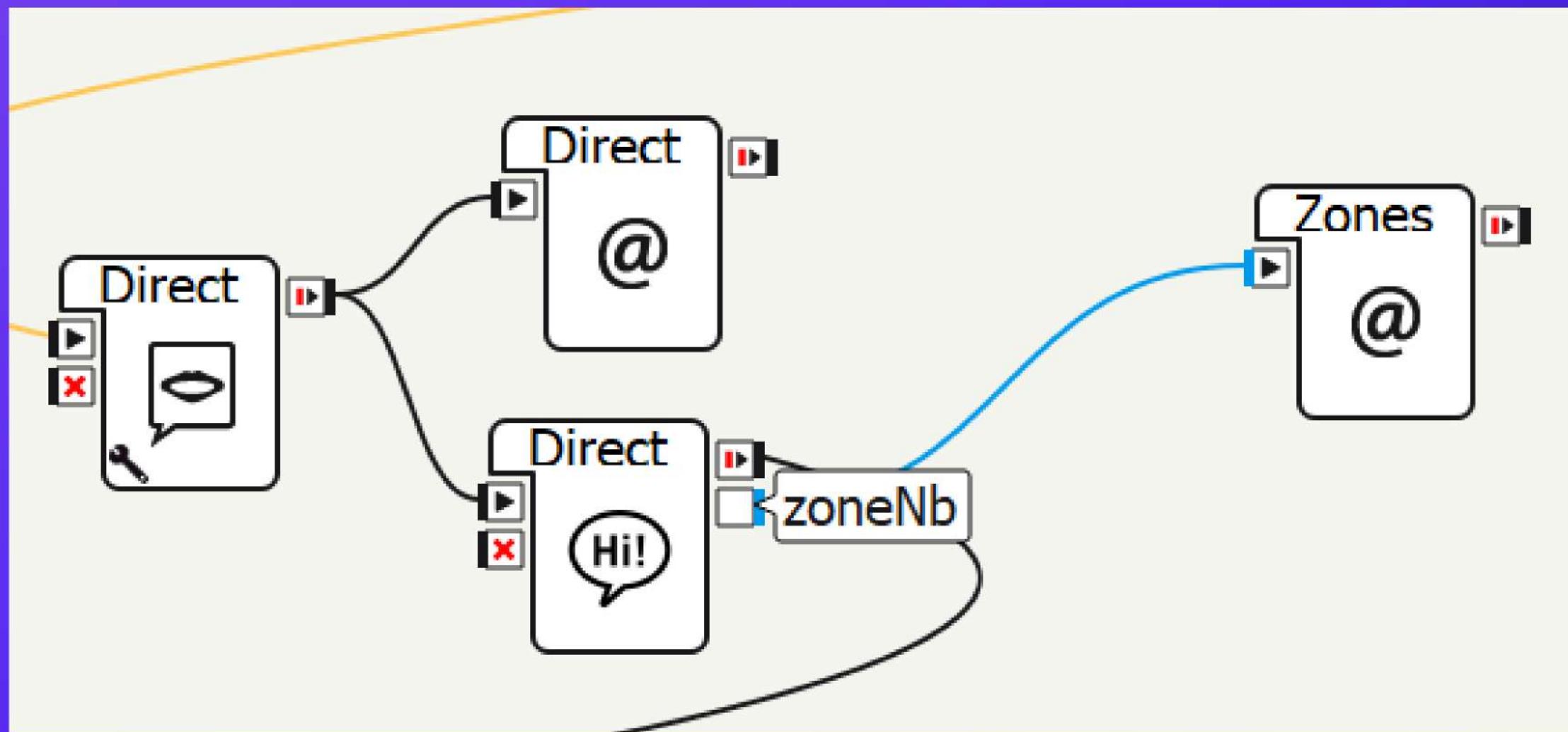


# STORY BOARD OF DIRECTING GUESTS



# IMPLEMENTATION USING CHOREOGRAPHY

- The flow begins with Direct Dialog, where Pepper asks the guests in which zone they are seated.
- Using the answer from the guest, zoneNB will be given one of 4 predefined values and will be passed to the Zone webview.



# IMPLEMENTATION USING CHOREOGRAPHY

```
def onInput_onStart(self, zoneNb):
    # We create TabletService here in order to avoid
    # problems with connections and disconnections of the tablet during the life of the application
    tabletService = self._getTabletService()
    self.logger.info("Zone Number: %s" % zoneNb)
    # Get the app name
    appName = self.packageUid()

    if appName:
        if tabletService:
            # use default robot IP address from the tablet
            robotIP = "198.18.0.1" # tabletService.getRobotIp()

            # Set page details
            # Assign pageHeading and pageImage based on zoneNb
            if zoneNb == "1":
                pageHeading = "Zone A!"
                pageImage = "zoneA.jpg"
            elif zoneNb == "2":
                pageHeading = "Zone B!"
                pageImage = "zoneB.jpg"
            elif zoneNb == "3":
                pageHeading = "Zone C!"
                pageImage = "zoneC.jpg"
            elif zoneNb == "4":
                pageHeading = "Zone D!"
                pageImage = "zoneD.jpg"
            else:
                pageHeading = "Unknown Zone!"
                pageImage = "unknownZone.jpg"

            pageText = "Choose an empty seat!"
            # Set the URL
            url = "http://{}{}/apps/{}/pages/zone.html?pageHeading={}&pageText={}{}&pageImage={}" .format(ip=robotIP, appName=appName, pageHeading=pageHeading, pageText=pageText, pageImage=pageImage)

            self.logger.info(url)

            # Show the web view on the tablet
            tabletService.showWebview(url)
        else:
            self.logger.warning("Couldn't find tablet service, so can't set application: %s" % appName)

    # self.onStopped()
```

- zoneNb will be received in the Zone webview as an argument of the onInput\_onStart function.
- Depending on the value of zoneNb, the pageHeading and pageImage fields will be defined and then passed into the url of the html page

# IMPLEMENTATION USING CHOREOGRAPHY

This is a closer look for the “directGuest” dialog:

```
topic: ~directGuest()
language: enu

proposal: %question In which region will you be seated?
    u1: (A) $zoneNb="1" ^gotoReactivate(speak)
    u1: (B) $zoneNb="2" ^gotoReactivate(speak)
    u1: (C) $zoneNb="3" ^gotoReactivate(speak)
    u1: (D) $zoneNb="4" ^gotoReactivate(speak)

proposal: %speak Now you can zoom in using the tablet to choose any seat, say or press continue to move forward.
    u1(continue): $onStopped=1

u: (e:onStart) Now let me direct you to your seat. \pau=250\ ^gotoReactivate(question)
u: (e:continueButton) $onStopped=1
u: (e:choose) $zoneNb=$choose ^gotoReactivate(speak)
```

On Start, Pepper will ask the user for the zone he/she was assigned.

The guest can choose to interact verbally or by pressing on the tablet, in both cases according to the specified zone, zoneNb will be given a predefined value.

When the guest finishes interacting with the map they can either say or press continue to move on.

# IMPLEMENTATION USING CHOREOGRAPHY

Using this JavaScript function:

It raises an event with a name <name> and assigns it a value <value>

```
function raiseEvent(name, value) {
    QiSession(function (session) {
        session.service("ALMemory").then(function (mem) {
            mem.raiseEvent(name, value);
        }, function (error) {
            console.log("An error occurred: ", error);
        });
    });
}
```

# IMPLEMENTATION USING CHOREOGRAPHY

```
<body>
  <div class="..." style="max-width:1100px">
    <div class="...">
      <div class="...">
        <div id="informationSection" class="...">
          <hr />
          <h1 id="pageHeading" style="font-size: 45px;padding-top:50px"></h1>
          <hr />
          <h3 id="pageText"></h3>
        </div>
        <div class="...">
          <div class="...">
            <button type="button" id="next" onclick="raiseEvent('choose', '1')" class="..." style="font-size:40px;">
              <span class="...">A</span><br>
            </button>
            <button type="button" id="next" onclick="raiseEvent('choose', '2')" class="..." style="font-size:40px;">
              <span class="...">B</span><br>
            </button>
            <button type="button" id="next" onclick="raiseEvent('choose', '3')" class="..." style="font-size:40px;">
              <span class="...">C</span><br>
            </button>
            <button type="button" id="next" onclick="raiseEvent('choose', '4')" class="..." style="font-size:40px;">
              <span class="...">D</span><br>
            </button>
          </div>
        </div>
        <div class="...">
          <img id="pageImage" class="..." alt="Pepper Pic" width="600" height="650">
        </div>
      </div>
      <hr />
    </div>
  </body>
```

- Button A -> raises event choose with a value of '1'
- Button B -> raises event choose with a value of '2'
- Button C -> raises event choose with a value of '3'
- Button D -> raises event choose with a value of '4'

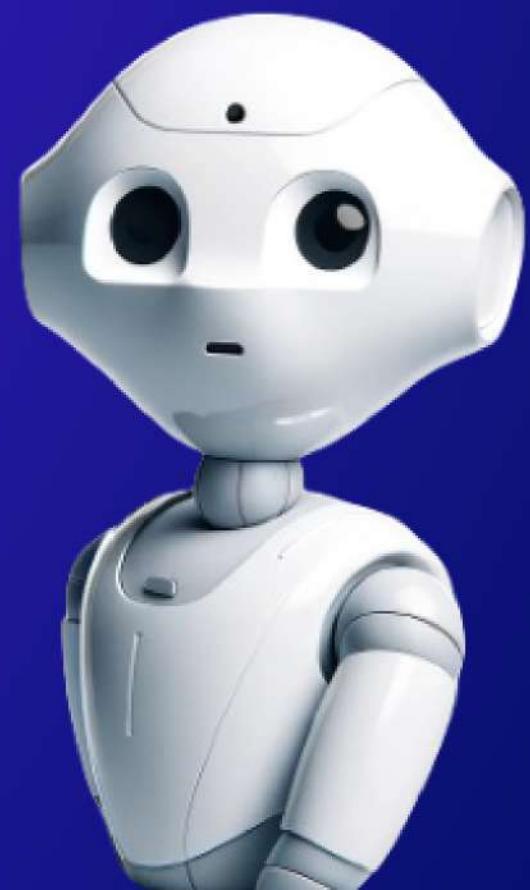
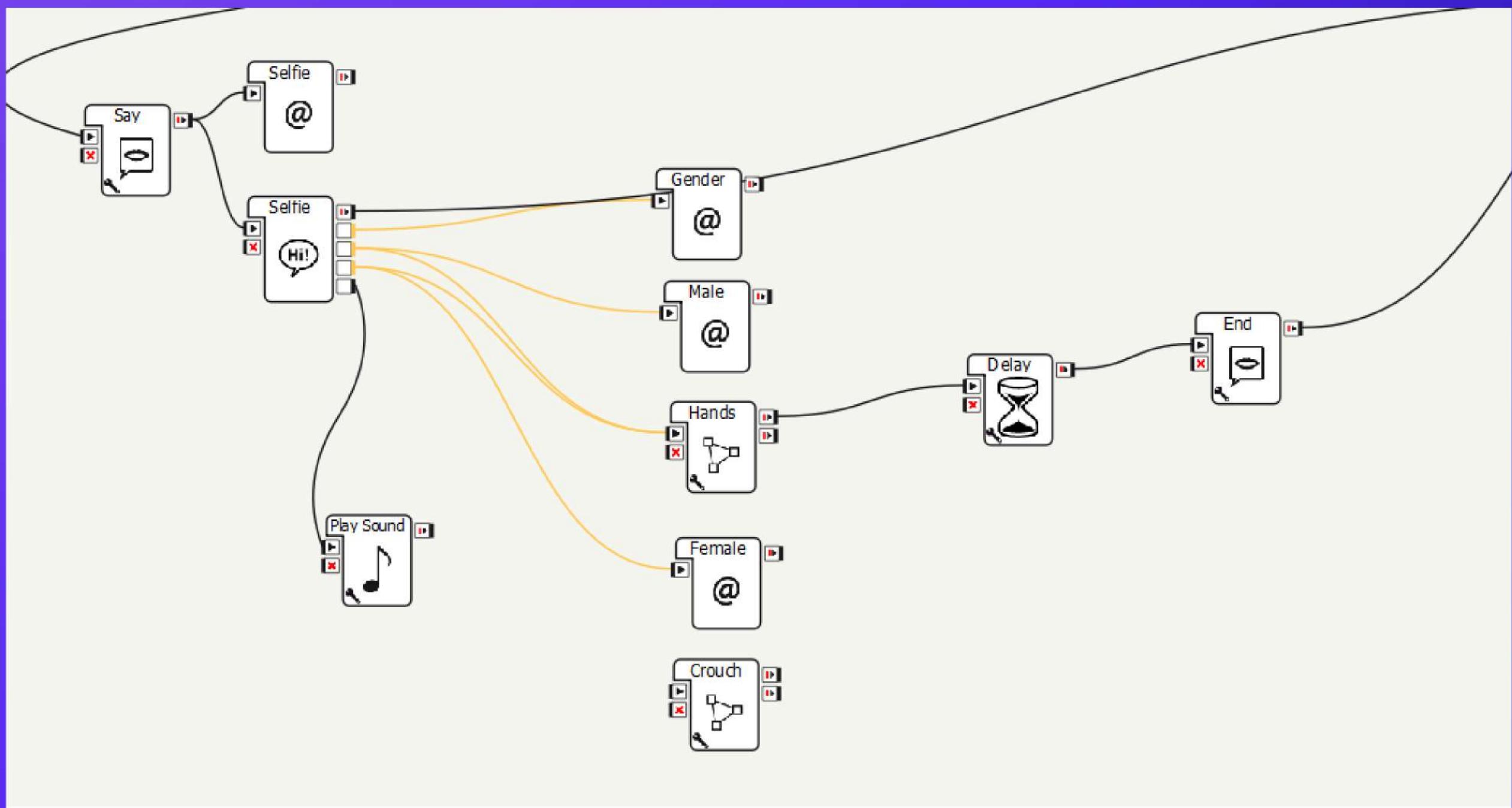
# TAKE SELFIE

In this use case pepper will interact with the parents and it will ask him if they love to take selfie with her

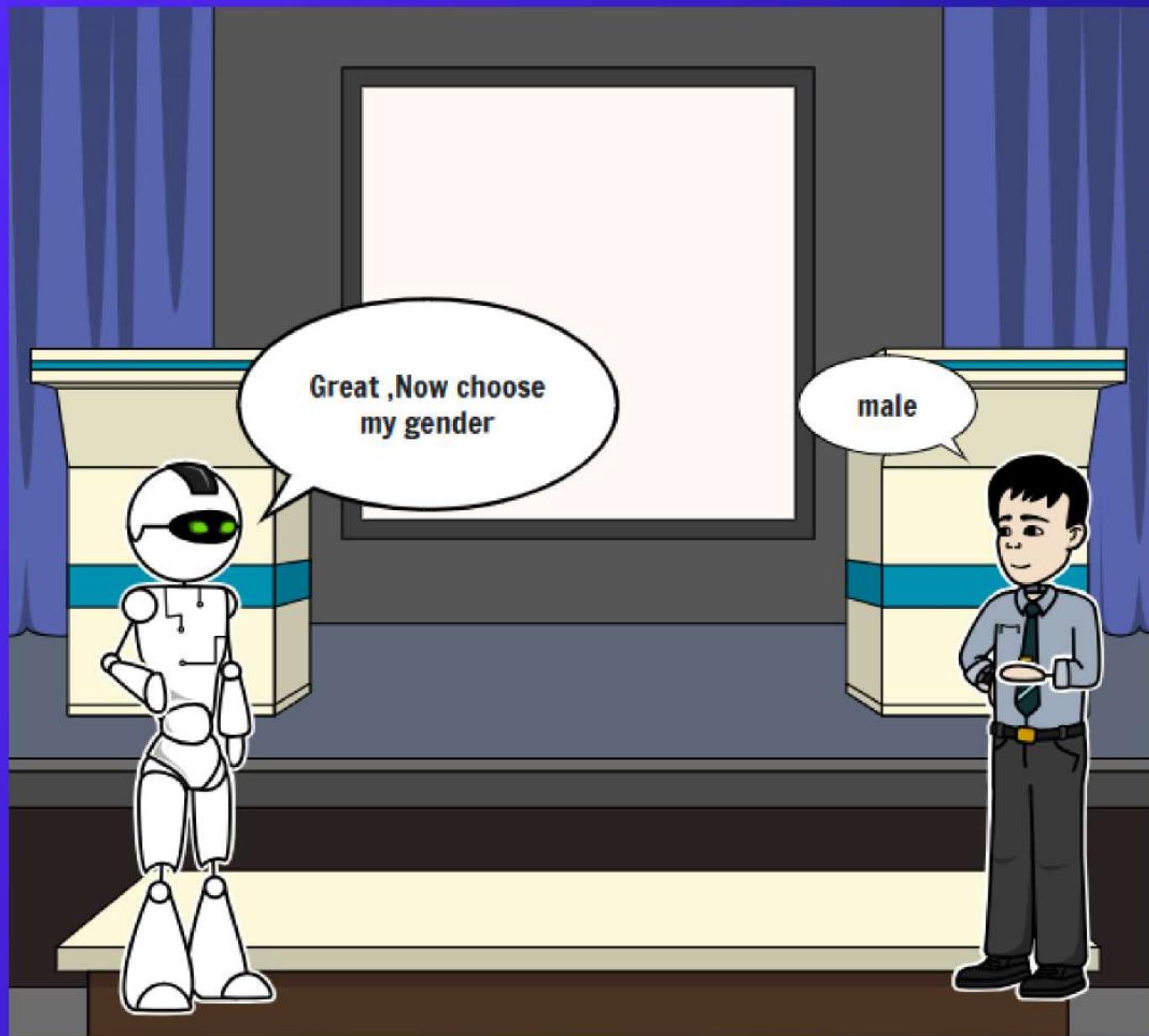
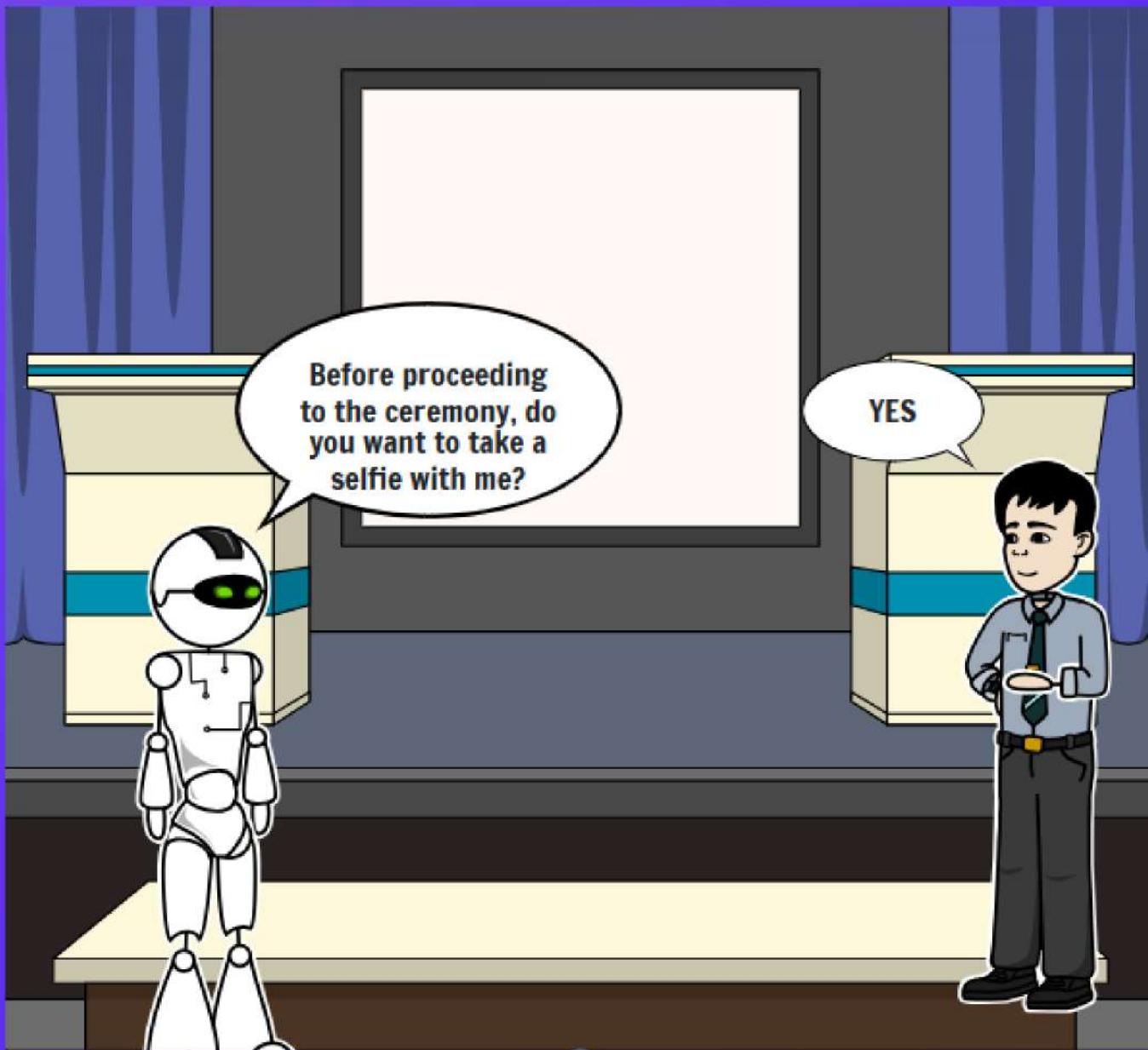
- It will ask the user if he like to take selfie or no
- If he said yes it will tell him to choose her gender
- After that it will make a small count down and give a camera sound

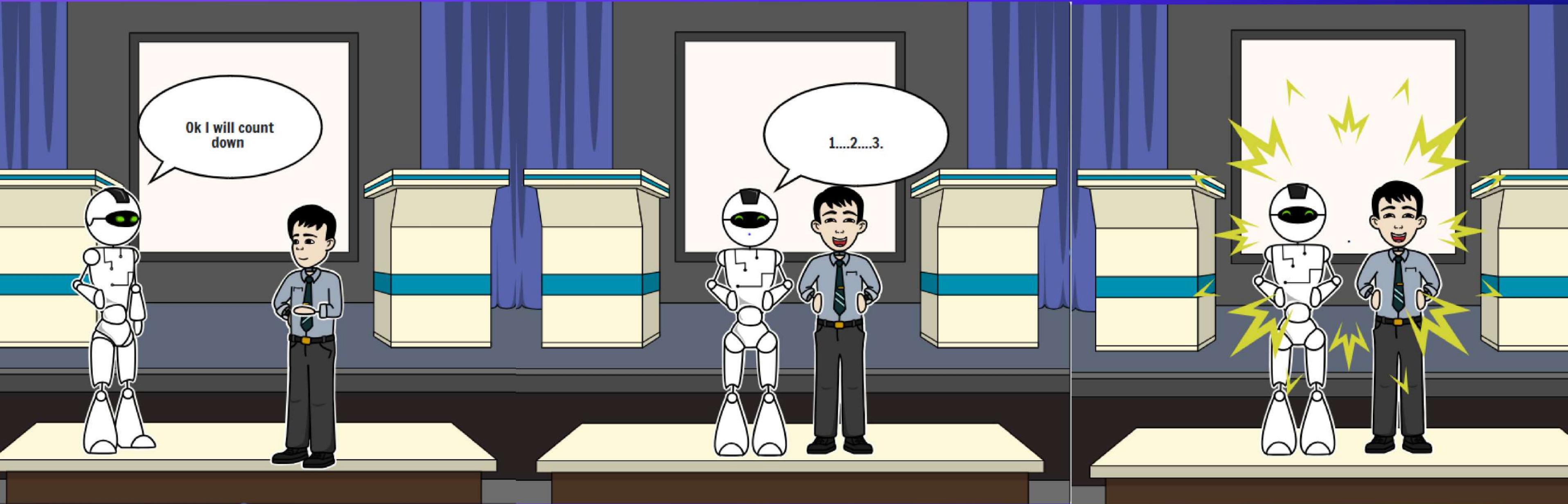


# TAKE SELFIE



# TAKE SELFIE STORY BOARD





# DIALOG IMPLEMENTATION

```
topic: ~selfie()
language: enu

concept: (approval) [yes sure "why not" absolutely]
concept: (denial) [no "another time"]

#selfie
proposal: %noselfie Have a great ceremony \pau=500\ $onStopped=1

u:(e:onStart) Before proceeding to the ceremony, do you want to take a selfie with me?
  u1: (~approval) Great $choosegender=1 ^gotoReactivate(gender)
  u1: (~denial) Too bad ^gotoReactivate(noselfie)
u:(e:yselfie) Great $choosegender=1 ^gotoReactivate(gender)
u:(e:nselfie) Too bad ^gotoReactivate(noselfie)

#gender
proposal: %gender \pau=200\ Now choose my gender
  u1: (male) ^gotoReactivate(male)
  u1: (female) ^gotoReactivate(female)

proposal: %male $maleoutput=1 Ok i will count down, 3 \pau=500\ 2 \pau=500\ 1 \pau=500\ $picsound=1
proposal: %female $femaleoutput=1 Ok i will count down, 3 \pau=500\ 2 \pau=500\ 1 \pau=500\ $picsound=1

#button inputs
u:(e:maleB) ^gotoReactivate(male)
u:(e:femaleB) ^gotoReactivate(female)
```

# WEB VIEW DYNAMIC IMPLEMENTATION

```
<body>
    <div class="..." style="max-width:1100px">
        <div class="...">
            <div class="...">
                <img id="pageImage" class="..." alt="Pepper Pic" width="600" height="650">
            </div>
            <div class="...">
                <div id="informationSection" class="...">
                    <hr />
                    <h1 id="pageHeading" style="font-size: 45px;padding-top:20px"></h1>
                    <hr />
                    <h3 id="pageText"></h3>
                </div>
                <hr />
                <div class="...">
                    <!-- Content goes here -->
                </div>
            </div>
        </div>
    </div>
</body>

# Set page details
pageImage = "male.jpg"
# Set the URL
url = "http://{}{}/apps/{}{}/pages/displayimage.html?pageImage={pageImage}".format(ip=robotIP, appName=appName,
pageImage=pageImage)
```

# WEB VIEW IMPLEMENTATION

```
</div>
<div class="w3-col m6 w3-padding-large">
    <div id="informationSection" class="w3-center">
        <hr />
        <h1 id="pageHeading" style="font-size: 45px; padding-top:20px"></h1>
        <hr />
        <h3 id="pageText"></h3>
    </div>
    <hr />
    <div class="w3-center">
        <button type="button" id="yesButton" onclick="raiseConfirmationEvent('yselfie')" class="fa fa-thumbs-up">
            <span class="default-font">Yes</span><br>
        </button>
        <button type="button" id="noButton" onclick="raiseConfirmationEvent('nselfie')" class="fa fa-thumbs-down">
            <span class="default-font">No</span><br>
        </button>
    </div>
</div>
<hr />
</div>
</body>
<script type="text/javascript">
    displayPageInformation();
</script>
</html>
```

ul: (~denial) Too bad ^gotoReactivate(noselfie)  
e:yselfie) Great \$choosegender=1 ^gotoReactivate(gender)  
e:nselfie) Too bad ^gotoReactivate(noselfie)

```
<div class="w3-col m6 w3-padding-large">
  <div id="informationSection" class="w3-center">
    <hr />
    <h1 id="pageHeading" style="font-size: 45px;padding-top:20px"></h1>
    <hr />
    <h3 id="pageText"></h3>
  </div>
  <hr />
  <div class="w3-center">
    <button type="button" id="maleButton" onclick="raiseConfirmationEvent('maleB')" class="fa rat
      <span class="default-font">Male</span><br>
    </button>
    <button type="button" id="femaleButton" onclick="raiseConfirmationEvent('femaleB')" class="fa
      <span class="default-font">Female</span><br>
    </button>
  </div>
</div>
<hr />
</div>
</body>
<script type="text/javascript">
  displayPageInformation();
</script>
</html>
```

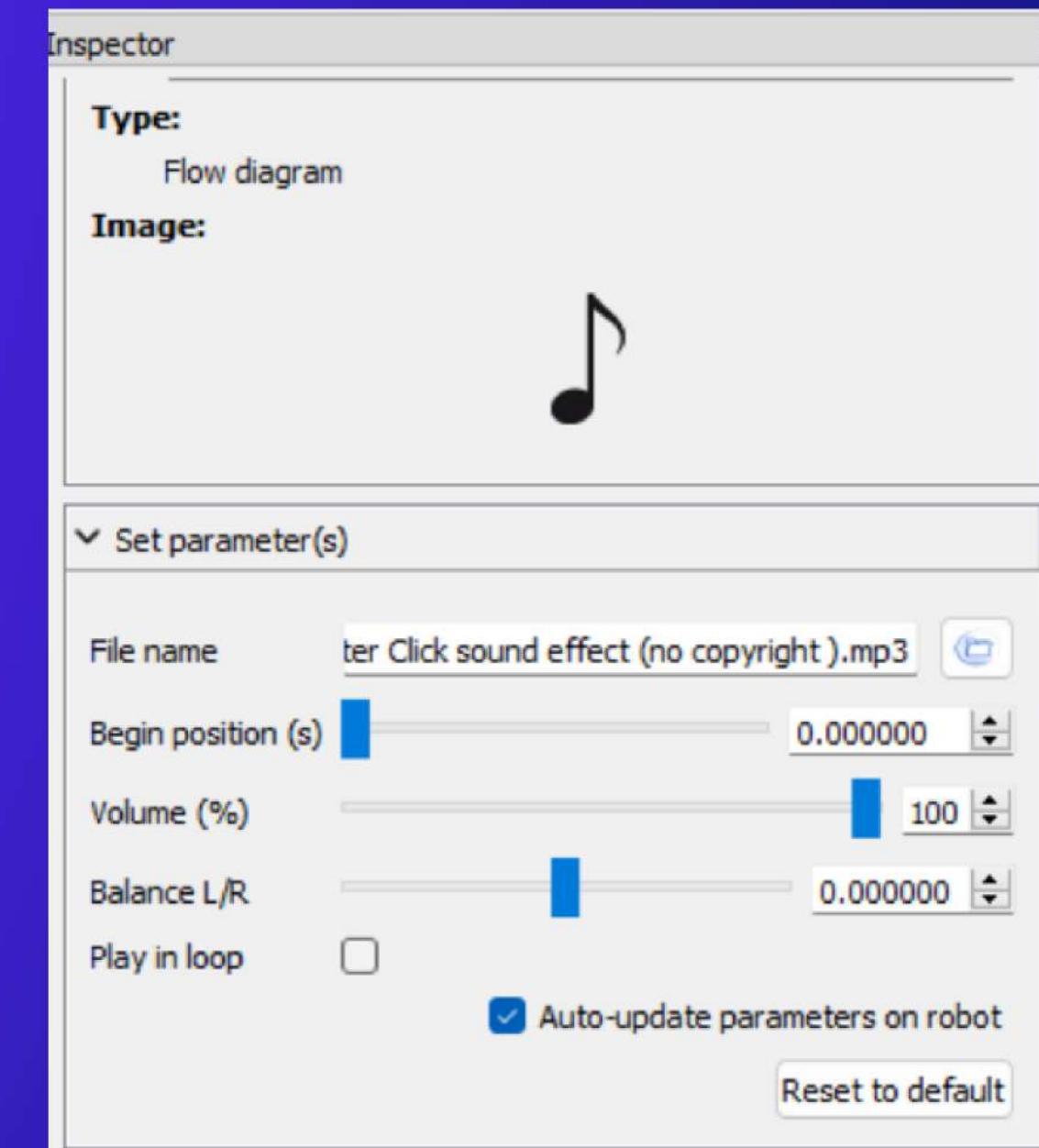
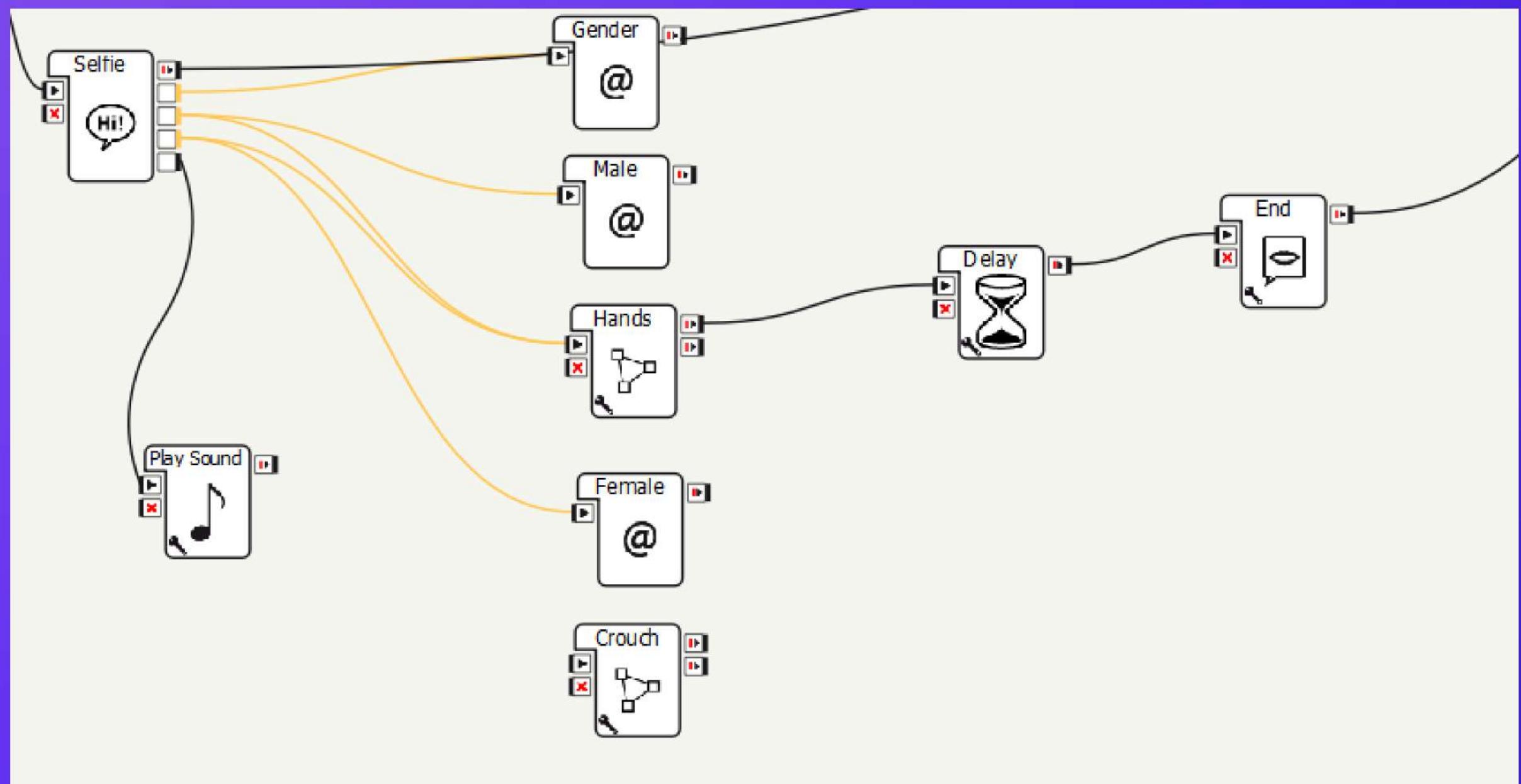
#button inputs

u: (e:maleB) ^gotoReactivate(male)

u: (e:femaleB) ^gotoReactivate(female)

\*

# SOUND AND DELAY IMPLEMENTATION



```
proposal: %male $maleoutput=1 Ok i will count down, 3 \pau=500\ 2 \pau=500\ 1 \pau=500\ $picsound=1
proposal: %female $femaleoutput=1 Ok i will count down, 3 \pau=500\ 2 \pau=500\ 1 \pau=500\ $picsound=1
```

# CHALLENGES FACED



- The error messages in Choregraphe weren't always clear.
- When designing the Scan QR Code use case, we found that the robot couldn't differentiate between two very similar photos.
- Sometimes, no error message appears, but nothing would work, and the only solution was to reboot Pepper.
- The available documentation for Qichat and for Pepper wasn't always easy to browse through, for example for the take a selfie part, we only found 2 postures in the documentation, but later on found out that animations can be used as postures by pausing them.



# FUTURE IMPROVEMENTS

- **Advanced Interaction Modes:** Implement voice recognition improvements and Introduce gesture-based interactions
- **Feedback Collection:** Equip Pepper with the capability to collect feedback from guests about their experience at the ceremony when they are leaving
- **Integration with Other Systems:** Enable synchronization with mobile devices, allowing guests to interact with Pepper or receive guidance directly through their smartphones.



# CONCLUSION



- Pepper has enhanced the guest experience by making the ceremony more accessible, informative, and enjoyable.
- This project showed how technology can be seamlessly incorporated into educational and formal settings to benefit all participants.

THANK YOU!

