First use case: Learning a new object

The use case which should be implemented can be described as follows: First Hobbit should grab the turntable from its storing position. Then a message on its tablet should be shown to ask the user to put an object on the table. After the user confirmed the placement, Hobbit should look at the object on the turntable and tell the user "I'm learning a new object" via its tablet interface. The table should turn clockwise first, before the user should be asked to place the the object upside down on the table. Again, the robot should wait for confirmation, then telling "I'm learning a new object" wihle rotating the table counterclockwise. After that, the user should be asked to remove the object and confirm the action. Then Hobbit should look straight, store the table and ask for the name of the object. Finally Hobbit should show a happy emotion and tell "Thank you, now I know what X is", where X is the name of the object. The desired workflow is visualed in figure 1.

Please implement a solution using the provided code editor, which lets Hobbit show the desired behaviour, with respect to the following conditions:

- Do exclusively use the provided editor for implementing your solution
- Start working by clicking the "Start" button of the interface
- Do not close the graphical editor during your work
- Click the "Stop" button when you finished implementation
- Click "Submit" to submit your solution

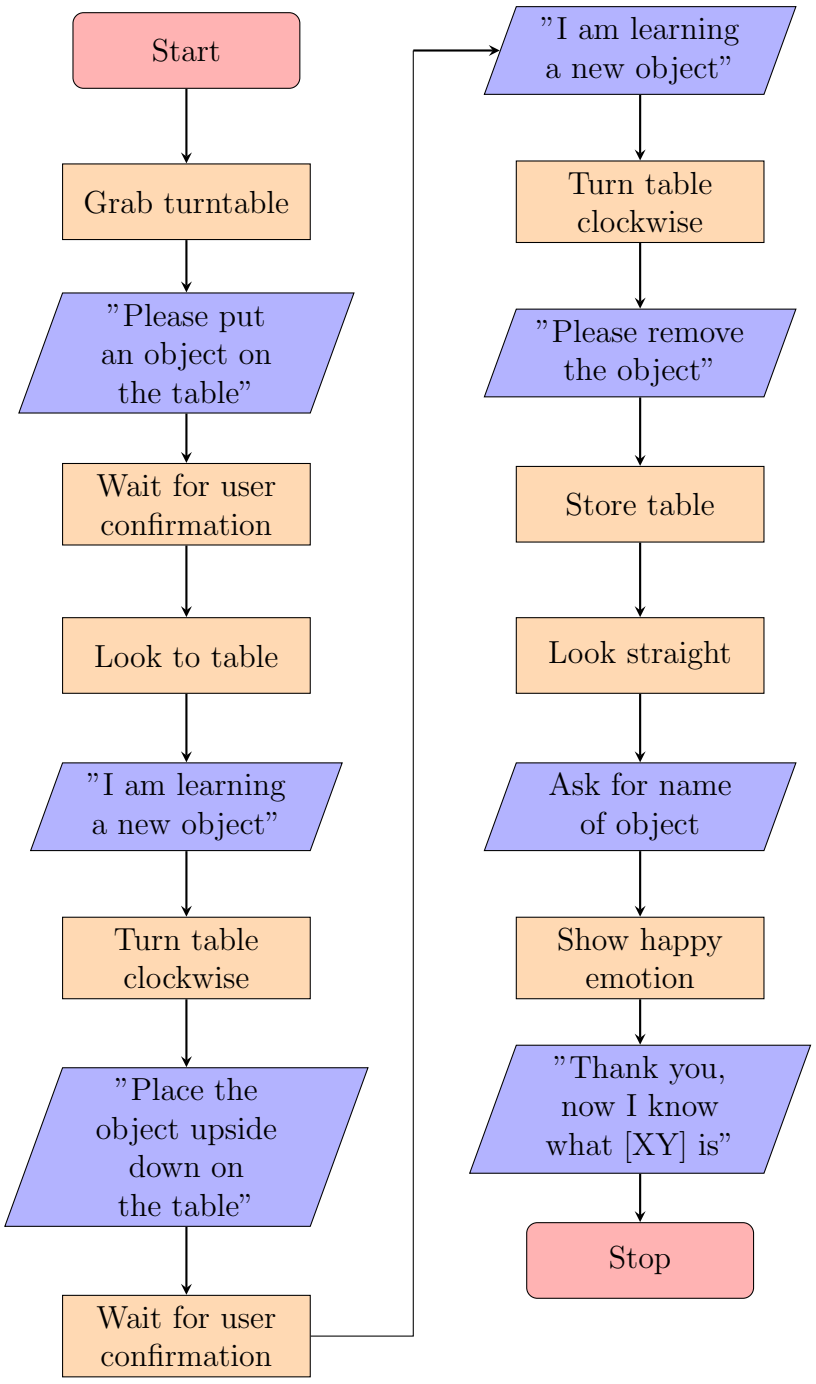

Figure 1: Flowchart of first use case

First use case: Learning a new object

The use case which should be implemented can be described as follows: First Hobbit should grab the turntable from its storing position. Then a message on its tablet should be shown to ask the user to put an object on the table. After the user confirmed the placement, Hobbit should look at the object on the turntable and tell the user "I'm learning a new object" via its tablet interface. The table should turn clockwise first, before the user should be asked to place the the object upside down on the table. Again, the robot should wait for confirmation, then telling "I'm learning a new object" wihle rotating the table counterclockwise. After that, the user should be asked to remove the object and confirm the action. Then Hobbit should look straight, store the table and ask for the name of the object. Finally Hobbit should show a happy emotion and tell "Thank you, now I know what X is", where X is the name of the object. The desired workflow is visualed in figure 1.

Please implement a solution using the the Blockly editor, which lets Hobbit show the desired behaviour, with respect to the following conditions:

- Start working by clicking the "Start" button of the interface
- Do not close the graphical editor during your work
- Click the "Stop" button when you finished implementation
- Click "Submit" to submit your solution
- Each block provides a help page - it is accessible via *right click* $\leftarrow$ *Help*

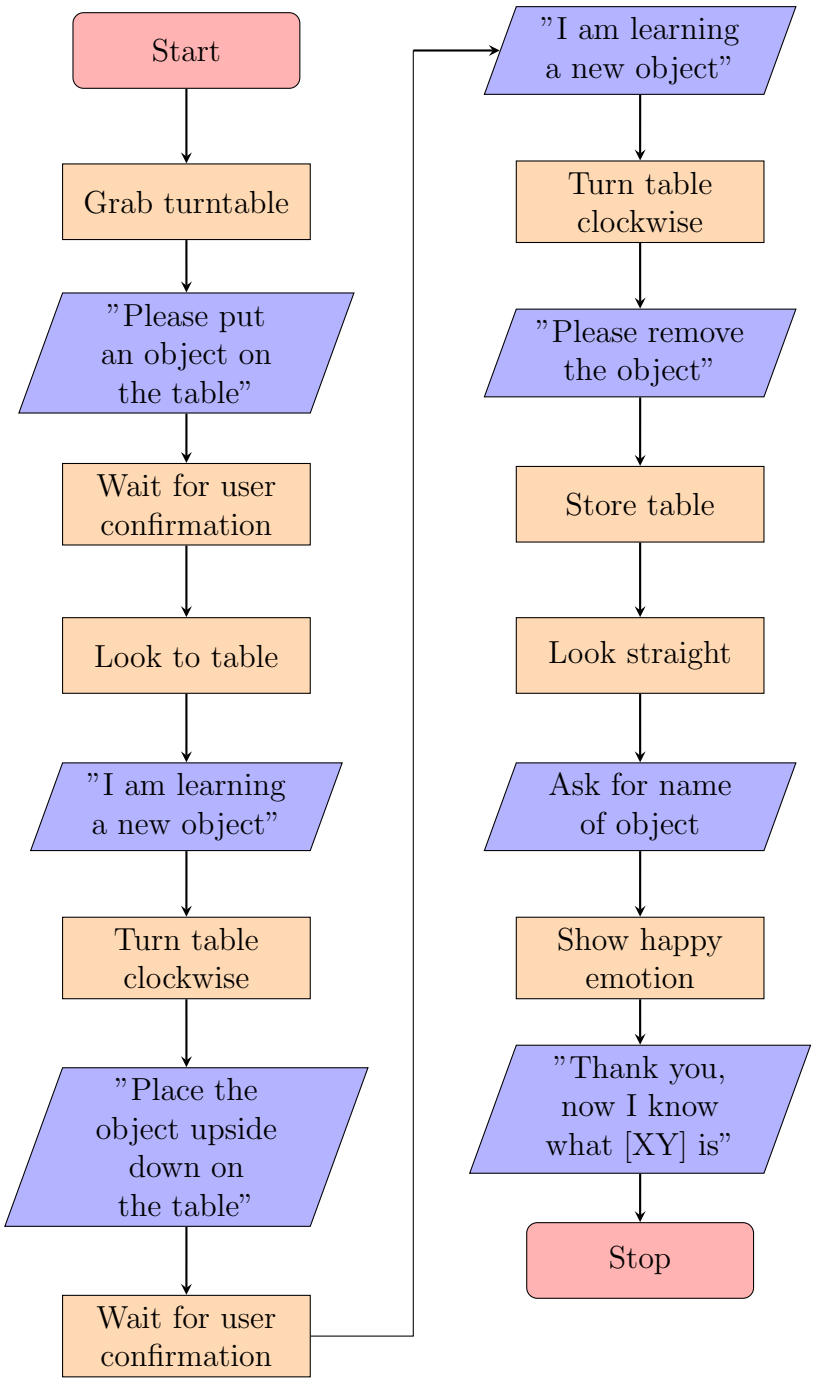

Figure 1: Flowchart of first use case

First use case: Bringing objects from another person

The use case which should be implemented can be described as follows: Hobbit should be repetitively asking User A if it should bring an object from User B, which is located at another place. First the user should be asked which objects should be picked up (e.g. "Which object do you want?"). User A then should use the robot's tablet to enter the name of the requested object. After that, Hobbit should navigate to the User B. User B then should be asked to handover the desired object. If it is answerd positively, the object should be placed on Hobbit's tray and the robot navigates back to its previous location telling User A "Here you are" and placing the object on the table. If the object has not been handed over, an appropriate message should be displayed on the tablet (e.g. "I'm sorry, your partner couldn't handover the object") after navigating back.

Afterwards User A should be asked, if Hobbit should bring another item. The whole procedure should be performed as long as User A does not request any other object. After the final decline Hobbit should show a happy emotion. For a better understanding figure 1 provides the flowchart of this use case. The locations of User A and B can be consired as the following poses:

- User A:
  - `position:{x:1.0,y:2.0,x:0.0}`
  - `orientation:{x:0.0,y:0.0,z:0.0,w:1.0}`
- User B:
  - `position:{x:14.0,y:-5.0,x:0.0}`
  - `orientation:{x:0.0,y:0.0,z:0.0,w:0.6}`

Please implement a solution using the provided code editor, which lets Hobbit show the desired behaviour, with respect to the following conditions:

- Do exclusively use the provided editor for implementing your solution
- Start working by clicking the "Start" button of the interface
- Do not close the graphical editor during your work
- Click the "Stop" button when you finished implementation
- Click "Submit" to submit your solution

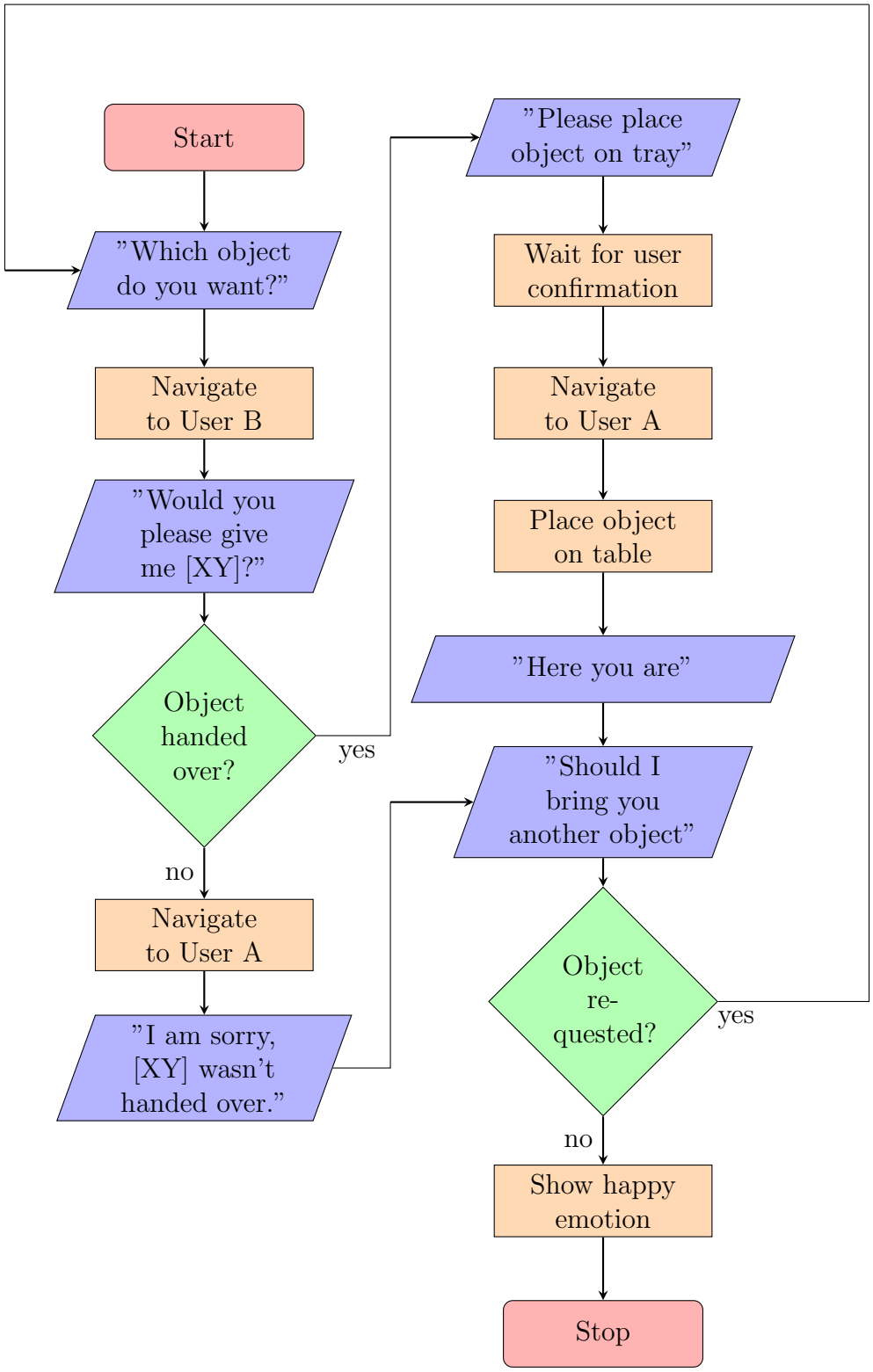

Figure 1: Flowchart of second use case

First use case: Bringing objects from another person

The use case which should be implemented can be described as follows: Hobbit should be repetitively asking User A if it should bring an object from User B, which is located at another place. First the user should be asked which objects should be picked up (e.g. "Which object do you want?"). User A then should use the robot's tablet to enter the name of the requested object. After that, Hobbit should navigate to the User B. User B then should be asked to handover the desired object. If it is answerd positively, the object should be placed on Hobbit's tray and the robot navigates back to its previous location telling User A "Here you are" and placing the object on the table. If the object has not been handed over, an appropriate message should be displayed on the tablet (e.g. "I'm sorry, your partner couldn't handover the object") after navigating back.

Afterwards User A should be asked, if Hobbit should bring another item. The whole procedure should be performed as long as User A does not request any other object. After the final decline Hobbit should show a happy emotion. For a better understanding figure 1 provides the flowchart of this use case. The locations of User A and B can be consired as the following poses:

- User A:
  - `position:{x:1.0,y:2.0,x:0.0}`
  - `orientation:{x:0.0,y:0.0,z:0.0,w:1.0}`
- User B:
  - `position:{x:14.0,y:-5.0,x:0.0}`
  - `orientation:{x:0.0,y:0.0,z:0.0,w:0.6}`

Please implement a solution using the Blockly editor, which lets Hobbit show the desired behaviour, with respect to the following conditions:

- Start working by clicking the "Start" button of the interface
- Do not close the graphical editor during your work
- Click the "Stop" button when you finished implementation
- Click "Submit" to submit your solution
- Each block provides a help page - it is accessible via *right click* → *Help*

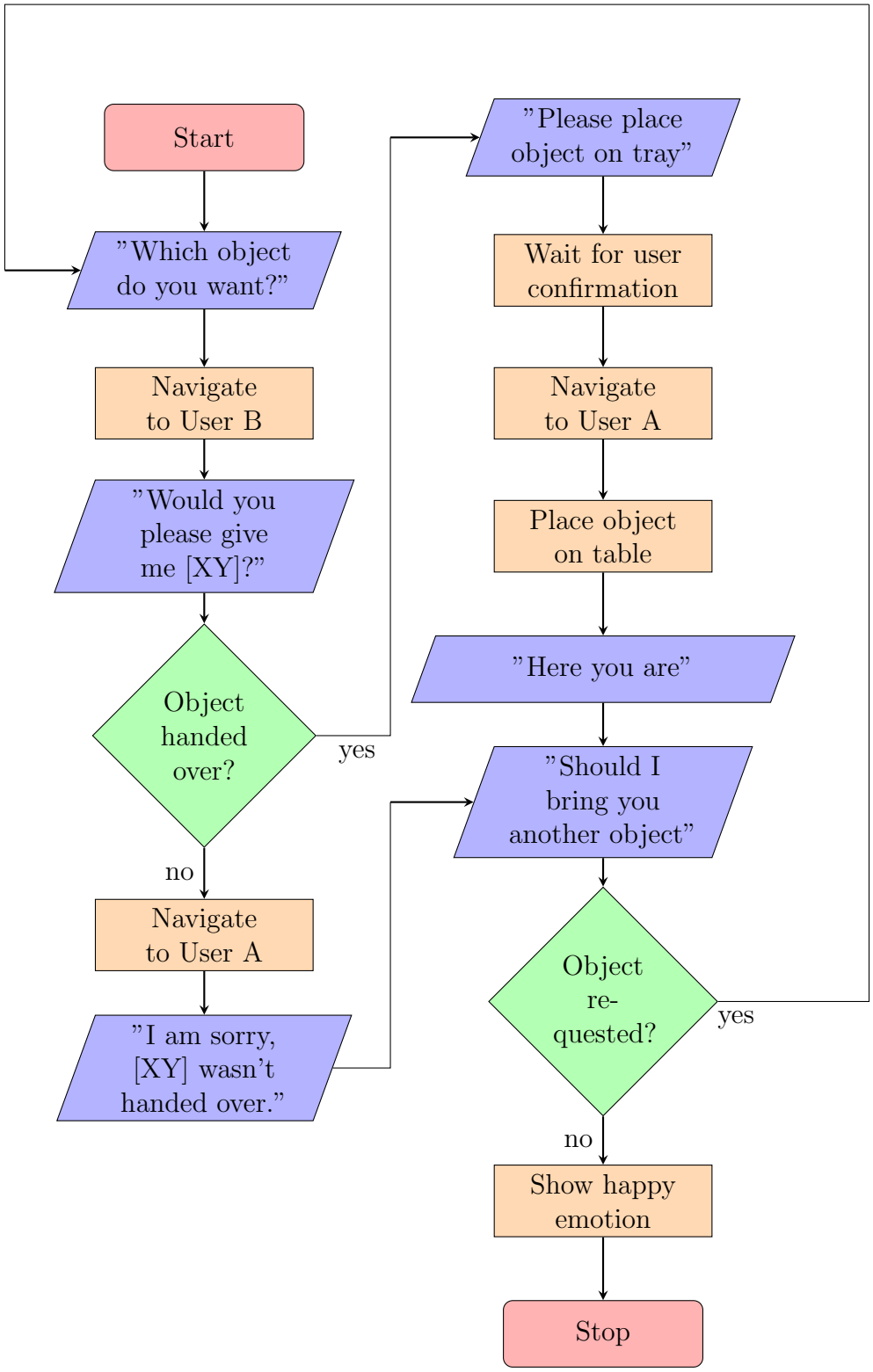

Figure 1: Flowchart of second use case