



PUSHBACKZOOMER 1.0

USER MANUAL

Author: Martin Rüedi

1 FEATURES

PushbackZoomer has the following features:

- Freeware (also no-adds) Windows 10 app
- Distributed via the Microsoft Store, as well as from flightsim.to
- Zero-controls GUI
- The only feedback, that the GUI provides, is the connection state to the sim via SimConnect (GUI in grey = idle / GUI colored = connected)
- Minimal resources footprint. By intelligent context sensing, the app only activates voice recognition during phases, when a pushback could occur. For the rest of the flight, the app will be running idle.
- Speech commands in English only.
- Speech commands follow the real world protocol as much as possible.
- Freely spoken commands are not understood, but the phrases need to match a set of predefined commands (see list of minimal commands below).
- A healthy number of variations is supported both for spoken phrases as well as for the responses.
- Depending on the mood of the virtual pushback driver, the conversation will have a different character. For every pushback any of the moods TightLipped, Formal or Expressive will randomly be applied to our virtual pushback driver.
- Literally hundreds of millions of different conversations with the pushback driver are possible.
- Robust speech recognition, even as non-native English speaker I easily get confidence values of 0.90 or more
- Any action can be canceled by saying “no, no, no”, if something was not understood correctly by the speech recognizer.

2 PROCEDURE

A typical pushback procedure is shown in this video:

<https://www.youtube.com/watch?v=j8qo4O00sIA>

Follow these steps to perform a pushback:

1. Start the PushbackZoomer app
2. Launch the sim and load or create a flight. Select a gate as starting point of the flight.
3. Enter the cold&dark flightdeck.

4. From the moment when the yellow **READY TO FLY**-button appears, at latest 30 seconds later, the pushback tug automatically moves to a position in front of the aircraft. If this step fails, request it by the voice command "*Request pushback*".
5. Perform all the necessary preparation steps at the gate. When ready for pushback, say "*ground from flightdeck*".
6. From that point on engage in the conversation with the pushback driver, follow his requests (releasing and later setting the parking brake, starting the engines) and provide commands as described in the next chapter.

3 SET OF VOICE COMMANDS

This is the minimal set of commands, which you should memorize:

- *Request pushback* -> Requests the pushback tug to move in front of the aircraft. Normally this step is performed automatically, however if for whatever reason the pushback tug does not appear in front of the aircraft automatically, you can request it with this command)
- *Ground from flightdeck* -> Start the conversation
- *Brakes released* -> Confirmation that the parking brakes have been released
- *Turn <right | left>* -> Requests a 90° turn to the right or left
- *Turn <right | left> by degrees* -> Requests a turn by the specified angle to the right or left
- *Turn facing <north | east | south | west>* -> Requests a 90° turn, the direction is derived from the cardinal direction
- *Stop turning* -> Stop turning and continue with current heading
- *No, no, no* -> Cancel started action
- *Stop pushback* -> Terminate the pushback
- *Brakes set* -> Confirmation that the parking brakes have been set

Additionally, the following command is supported too:

- *Test pushback zoomer* -> Receive a spoken status update of the app. This can be used to test the proper audio setup and to troubleshoot the app.

4 FULL SRGS GRAMMAR

The full set of supported voice commands is precisely defined by the implemented SRGS grammar. Here is a copy of the full grammar, so if it does not bother you to read a bit XML, here you can see all the supported spoken variations:

Version 1.0.10:

```
<?xml version="1.0" encoding="utf-8"?>
<grammar xml:lang="en-US" root="pushbackZoomerVoiceInput"
tag-format="semantics/1.0" version="1.0"
xmlns="http://www.w3.org/2001/06/grammar">

  <rule id="testPushbackZoomer">
    <item>Test pushback zoomer<tag> out.command="testPushbackZoomer"</tag></item>
  </rule>

  <rule id="pushbackRequested">
    <item>Request pushback<tag> out.command="pushbackRequested" </tag></item>
  </rule>

  <rule id="triggerActivities">
    <item>
      <one-of>
        <item>Ground from flightdeck</item>
        <item>Ground from cockpit</item>
        <item>Flightdeck to ground</item>
        <item>Cockpit to ground</item>
        <item>Hi ground, flightdeck here</item>
      </one-of>
      <item repeat="0-1">
        <one-of>
          <item>ready for pushback</item>
          <item>we are ready for pushback</item>
        </one-of>
      </item>
      <tag> out.command="triggerPushbackActivities" </tag>
    </item>
  </rule>

  <rule id="brakesReleased">
    <item>
      <one-of>
        <item>Brakes released</item>
        <item>We are good to go</item>
        <item>You can start the pushback</item>
      </one-of>
      <tag> out.command="brakesReleased" </tag>
    </item>
  </rule>
```

```

<rule id=""turnRequested"">
  <item repeat=""0-1"">
    <item>Please</item>
  </item>
  <item>
    <one-of>

      <item>
        <item>turn facing<tag>out.command=""turnRequested""</tag></item>
        <item>
          <ruleref uri=""#cardinalDirection""/>
          <tag> out.cardinalDirection=rules.latest(); </tag>
        </item>
      </item>

      <item>
        <item>turn <tag>out.command=""turnRequested""</tag></item>
        <item>
          <ruleref uri=""#direction""/>
          <tag> out.direction=rules.latest(); </tag>
        </item>
        <item repeat=""0-1"">
          <item>by</item>
          <item>
            <ruleref uri=""#digit""/>
            <tag> out.first_digit=rules.latest(); </tag>
          </item>
          <item repeat=""0-1"">
            <item>
              <ruleref uri=""#digit""/>
              <tag> out.second_digit=rules.latest(); </tag>
            </item>
            <item repeat=""0-1"">
              <item>
                <ruleref uri=""#digit""/>
                <tag> out.third_digit=rules.latest(); </tag>
              </item>
            </item>
            <item>degrees</item>
          </item>
        </one-of>
      </item>
    </one-of>
  </item>
  <item repeat=""0-1"">
    <item>now</item>
  </item>
</rule>

<rule id=""turnStopRequested"">
  <item repeat=""0-1"">
    <item>Please</item>
  </item>
  <item>
    <one-of>
      <item>stop turning</item>
      <item>stop the turn</item>
    </one-of>
    <tag>out.command=""turnStopRequested""</tag>
  </item>
  <item repeat=""0-1"">
    <item>now</item>
  </item>
</rule>

```

```

<rule id=""pushbackStopRequested"">
  <item repeat=""0-1"">
    <item>Alright,</item>
  </item>
  <item repeat=""0-1"">
    <item>that's it,</item>
  </item>
  <item>
    <one-of>
      <item>stop the pushback</item>
      <item>you can stop the pushback</item>
    </one-of>
    <tag>out.command=""pushbackStopRequested""</tag>
  </item>
  <item repeat=""0-1"">
    <item>now</item>
  </item>
</rule>

<rule id=""brakesSet"">
  <item repeat=""0-1"">
    <item>Parking</item>
  </item>
  <item>
    <one-of>
      <item>brakes set</item>
      <item>brakes are set</item>
    </one-of>
    <tag> out.command=""brakesSet"" </tag>
  </item>
</rule>

<rule id=""negativeReply"">
  <item>
    <item repeat=""0-1"">
      <item>No, </item>
    </item>
    <one-of>
      <item>no no no</item>
      <item>don't do that</item>
      <item>Negative</item>
      <item>stop doing that</item>
      <item>stop stop stop</item>
    </one-of>
    <tag> out.command=""negativeReply"" </tag>
  </item>
</rule>

```

```

<rule id=""digit"">
  <one-of>
    <item>
      zero<tag> out=""0"" </tag>
    </item>
    <item>
      one<tag> out=""1"" </tag>
    </item>
    <item>
      two<tag> out=""2"" </tag>
    </item>
    <item>
      three<tag> out=""3"" </tag>
    </item>
    <item>
      four<tag> out=""4"" </tag>
    </item>
    <item>
      five<tag> out=""5"" </tag>
    </item>
    <item>
      six<tag> out=""6"" </tag>
    </item>
    <item>
      seven<tag> out=""7"" </tag>
    </item>
    <item>
      eight<tag> out=""8"" </tag>
    </item>
    <item>
      nine<tag> out=""9"" </tag>
    </item>
  </one-of>
</rule>

<rule id=""direction"">
  <one-of>
    <item>
      left<tag> out=""left"" </tag>
    </item>
    <item>
      right<tag> out=""right"" </tag>
    </item>
  </one-of>
</rule>

<rule id=""cardinalDirection"">
  <one-of>
    <item>
      north<tag> out=""north"" </tag>
    </item>
    <item>
      east<tag> out=""east"" </tag>
    </item>
    <item>
      south<tag> out=""south"" </tag>
    </item>
    <item>
      west<tag> out=""west"" </tag>
    </item>
  </one-of>
</rule>
</grammar>

```