Blockly

The GitHub page for the accessible version of Blockly (see here: <https://github.com/google/blockly/tree/master/accessible>) gives some guidance about how to integrate it into the app. Included in the /JS/Blockly folder are all of the Blockly files needed to integrate accessible Blockly into the application. Note, the only files in the Blockly folder needed for the current implementation of Blockly to run are “blockly\_compressed.js", “javascript\_compressed.js" “blocks\_compressed.js", “BlockDefinitionsJSON.js", “BlockCodeGenerator.js”, and “en.js". The rest are purely for future use in integrating the accessible version of Blockly as well as building future versions of Blockly.

Canvas

There are a number of resources available that give some great guidance on making canvas elements accessible. Below is one of the better sources found.

The basis of what is said is to provide a text description of what is happening on the canvas at each stage of the change.

Within GUI.js there is gameArea variable object. This object has a function updateAria. Using this will allow for text description to be added to the canvas to allow for updates to occur when the robot moves.

Examples of text descriptions would be something along the following.

<h2>Map Size</h2>

<p>The map is 2 rows by 3 columns</p>

<h2>Game Play</h2>

<p>The Robot is in row one column 1 facing east. There are walls north, east and west.</p>

<https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Hit_regions_and_accessibility>

Web Page

All of the button on the page are already using role attributes and aria-label attributes. This will allow for navigation.

Additionally, the buttons on the main page offer tabindex attributes to allow tabbing through the buttons.

Images also are using aria standard Alt attributes to describe the images.