

**Assistance Bot Factory
Requirements Document
Comp Sci 4ZP6 - Capstone
Fall 2019 - Winter 2020**

Contact information/SRS team members

Tyler Phillips - philipst@mcmaster.ca
Hamid Yuksel - yukselh@mcmaster.ca
Jakub Pawlikowski - pawlikoj@mcmaster.ca
Anthony Mella - mellaa@mcmaster.ca

Table of Content

WHAT - Information on the system we are proposing.

USE - The intended use of the system.

USERS - The audience of the system.

ENVIRONMENT - The expected environment for the system to be used in.

FUNCTIONALITY - The behaviour of the system. Also includes non-functional requirements.

CONSTRAINTS - Limitations to be addressed while building the system.

Document conventions

McDSL	McMaster Decision Sciences Laboratory
VR	Virtual Reality

WHAT

The purpose of the project is to create a platform for the McMaster Decision Science Lab (McDSL) to easily build and configure assistance bots for their virtual reality experiments. The platform will have an interface to add the necessary configurations and view different bot configurations.

This platform will be a proof of concept interface used by experiment conductors to outfit assistance bots with necessary mechanisms to assist users while they are participating in experiments.

For example, in a VR game where a user must use a bucket to get water, sometimes the user may not realize there is a bucket available. During these cases, an assistance bot would be able to help the user realize where the bucket is, without the experimenter intervening or the experiment taking too long to complete.

As well, the assistance bot is expected to record data on the experiment itself and provide it to the experimenters for further analysis. The recorded data could represent anything from the number of times the experimentee did X, or how long it took for experimentee to perform Y.

A final component is version control. It was requested that a simplistic version control system be implemented so that the experimenters can keep track and reuse previous versions of bot configurations.

The architecture of the platform is aimed to be modular enough to allow for more ambitious components to be easily added in the future, such as a machine learning component for assistance bot, or more advanced bot configurations.

USE

The goal is to enhance the virtual reality experiments as well as provide the lab conductors with more advanced control options over VR experiments settings. Current VR experiments are not equipped with any advanced in game assistance mechanisms. This is a problem because it requires the conductors to interrupt the experiment to provide assistance.

The lab is looking for a more automated way of providing assistance without needing the experimenter to intervene. This is due to the nature of VR experiments; experimenters are often in another room observing and not directly in contact with the experimentee. It is worried the intervention can cause changes in the outcome of the experiment.

As well, the need for configuration capabilities stem from the fact that the VR experiments play more as VR games. They are not traditional games, but simple challenges such as moving water from one well to another using a bucket.

By having the bot configurable, that allows the lab to have more options to adjust the stimuli of the experiment. For example how would the user react if the bot had a different tone of voice or different appearance? These are questions that arise during experiments and by having a customizable bot interface it will allow for these questions to be answered.

The bot is expected to have a natural cadence and can interact to a limited degree with the experimentee. Should the experimentee need help or is stuck, the bot is there for them to ask for assistance or guidance. The experimenters will also be carefully observing how the experimentee interacts with the assistance bots.

Configurable options for the bot include currently:

- Scripted speech patterns
- Limited degree of understanding user
- Physical appearance in VR games
- Limited behaviour options (if X then do Y)

USERS

The intended users of the product are the McDSL lab conductors who will be using the platform to configure the VR assistance bots. McDSL is a brand new research facility at McMaster University with a special focus on empirical research into human decision making.

The actual users, however, are those who will be directly interacting with the end product (the assistance bots produced). These people are experimentees of McDSL's VR experiments. They will be in the experiments, and will be interacting with the assistance bot configured by McDSL.

The two intended users will interface with the product differently

1. McDSL lab conductor in charge of configuration of assistance VR bots
 - The initial interface the lab technician will be faced with will be an input form in which predefined configuration fields will be submitted
 - The expected details of the configuration form may vary and must be initially designed with modularity as a priority
 - Configurations can vary from bot's appearance, speech patterns and abilities
2. Lab experiment participant interacting with a bot within a virtual reality simulation

- Depending on the experiment the details of the interface with the assistance bot may vary however the lab experiments will be conducted in specific VR enabled rooms equipped with HTC Vive hardware
- The interaction with the assistance bots within the VR simulation is meant to facilitate and improve the lab experiments. The end goal would be interfacing via speech however for proof of concept purposes the aim will be text based interaction between the lab user and the bot.

To summarize the lab conductors will be interacting with the bot configuration interface while the lab experiment users will be interacting with the VR assistance bots themselves. The two audiences are equally vital in the overall effectiveness of the product.

ENVIRONMENT

The assistance bots will be part of VR lab experiments conducted by the McDSL. There are two identical lab rooms equipped with HTC Vive headsets including joystick controllers, microphones, speakers and dedicated computers for handling the simulations. The VR simulations will be running on unity and the computers using windows 10. The bots themselves will be part of the VR lab simulations which will vary, a basic one which will be used for initially development and testing purposes is a simple water game. As an example the water game is meant to simulate a basic work task and observe decisions the user is faced with, the environment of this game is a small room.

FUNCTIONALITY

1. Speech Detection

- Description and priority
 - The speech detection will be collected using the microphone on the HTC vive headset, the speech will only be recorded if the user presses the record button on the right controller, this will eliminate complexity by only recording speech the user is requesting to be recorded. This is one of our primary features and will have high implementation priority.
- Action/result
 - The user will speak into the microphone, and the system will detect and analyze the speech using a speech detection API, and then based on what the user says the bot will reply with advice accordingly.

- Functional requirements
 - The program will begin speech detection upon the user's command. Note that a speech to text api will be implemented to assist in the following step. The speech will be recorded and the data input will be run through our api. The data will be converted into text and we will process the information and have the assistance bot respond accordingly.

2. Progress Scoring

- Description and priority
 - The Progress scoring feature will allow the participants progress to be recorded and scored, ranging from a low to high numeric score. A high score indicates that the user is making progress with the experiment and the assistance bot will not be prompted to give advice, a low score indicates that the user is struggling with the tasks. If a low score is detected, based on the events happening during the gameplay the bot will give advice accordingly.
- Action/result
 - If the task is to transport water to a well using a bucket and the user is spilling water, the water spilling and other events (such as the angle of the bucket with respect to the ceiling, the speed of the user) will be recorded, and the bot will provide advice such as "make sure to hold the bucket straight and to walk at a moderate pace to avoid spills".
- Functional requirements
 - The progress scoring will be implemented by keeping track of task objects in the game to know if they have reached an objective or not. The task objects can consist of objects that the user needs to complete a task, and the objective is when a task is completed. If the user is struggling to complete a task then it will be recorded, this will be very situational depending on the session and the objectives in that session

3. Configuration Input

- Description and priority
 - The configuration feature will be available to the lab conductors, it will allow them to set configurations for how the bot will be presented to the participants. The immediate solution is to have the configuration as a file, that will act as input to a new session, and this file will contain properties that will be applied to the bot during a new session. Some of the properties can be appearance (i.e gender, clothes, etc), tone of voice, and mood. The end goal would be a user interface that the user can graphically set these configurations before each session.

- Action/result
 - The action would be the McDSL experiment conductor providing input to the simulation to set the bot configurations, such as tone of voice, default phrases, and appearance. The result would be the bot taking on these configurations during the simulation run.
- Functional requirements
 - The configuration platform must be modular (configuration options expected to change over time) and support all different settings depending on the session details (for example: not all configurations may apply to all potential sessions).

4. Bot Template

- Description and priority
 - A generic bot template will be required to perform the actions of the bot in game. This template will be required for positioning the bot in the game, and the physical motions the bot will perform during the game.
- Action/result
 - The bot template will be pre defined, when the user selects the configurations (position, appearance, voice, default phrases) then the bot in game will take on these features and be displayed to the user as a result.
- Functional requirements
 - There will be a generic bot model that is required to be customized with appearance, voice, and default phrase features at the start of each new game. Once the user selects they're customization options, the bot will be placed in the game and will speak the default phrases when appropriate.

5. Non-functional requirements

- Performance requirements
 - Smooth integration with the lab experiment simulation
 - Processing of information related to the assistant bot must not slow down the virtual reality simulation
 - interaction must be up to standards compared to the rest of the lab experiment simulation
- Safety requirements
 - There are no safety concerns regarding the assistance bot itself
 - Hardware safety: virtual reality equipment
 - Do not use if you have a history of epilepsy or seizures
 - Ensure the hardware is used in an open space free of any obstacles. The user will be completely unaware of their surroundings once the product

is in use. Hazards such as cables need to be moved out of the user's path to avoid injuries

- Virtual reality device may induce motion sickness, if this starts to occur please stop using immediately
 - Observers must stay out of the VR user's path as any physical movement the simulation requires may lead to accidental collisions
- Security requirements
 - Do not provide any private information to the navigator bot as all speech is processed and translated
 - Additional project considerations
 - Since this project is meant to be a proof of concept for assistance bots platform development it will be very important to keep track of progress and note the development process. McDSL will greatly benefit in future projects if there is strong documentation surrounding what has already been tried and how effective it was. Over the course of the project, the developers will keep note of things they research and attempt to implement and report this information to the McDSL.

CONSTRAINTS

- Limitations with natural language processing will be a limitation on how many voice configurations are possible (eg, tone, accent, mood)
- Limitations in time will result in designs to be focused on building an architecture for bot configurability rather than the bots themselves.
- Limitations in time will also constrain the amount of machine learning that is put into the bots.
- One limitation would be not knowing all of the simulation scenarios the bots will be in, so we will have to focus on the bots being more general