AINT512 Project Proposal

# Project Concept:

The concept for the project is a text-adventure style game using speech instead of monitors and keyboards. Using the game “The Stanley Parable” as a guide, the computer will take the role as a narrator, and the player will direct the flow of the story when prompted to do so.

# Design Process:

## Task Domain:

The domain of the program will be to present and guide a user through a specific choose your own adventure story.

## Dialogue Specification / Observation:

A detailed dialogue specification, along with a dialogue flow-chart, will be available upon story completion.

## Personality Specification:

The narration will be polite and well spoken.

## Grammar Design:

Several use-cases will be used to identify the user’s intentions:

* Yes / No Responses:
  + Yes implies to perform the dialogue directed narrative
  + No implies to perform no action
* Direct Sentence-Based Responses:
  + The user explicitly responds with just the related action
  + Follow the users direction if it matches a pre-scripted path, else rephrase to direct the user again
  + The user may use additional descriptors, which should be referenced in the response narration
* Indirect Sentence-Based Responses:
  + The user responds with a full sentence to describe their desired action
  + Keywords will be scanned for
  + Should conflicting information arise, the user will be directed towards a statement that can be resolved by the code

## Dialogue Design:

Several building blocks will be utilised in the creation of the dialogue for the computer.

* Dialogue-directed texts to guide the user towards possible narrative options
* Scenes & sequels to keep the users attention
* User input cues, such that the user knows when to provide input in an intuitive, non-intrusive, manner
* The use of three options at most points where a user can direct the narrative:
  + Yes: The user performs the dialogue-suggested action
  + No: The user performs the exact opposite of the dialogue-suggested action
  + Different: The user performs some other action entirely, within reasonable relevance

## Error Catching & Handling:

Several errors can be expected:

* Speech to text mistranslation or incomplete input (e.g. user silent for prolonged period of time)
  + Results in garbage in, resulting in a high likelihood of garbage out
  + Resolve with the narrator apologising for their bad hearing in a variety of different manners
* Conflicting information detected in input
  + Results in an inability for the program to progress with the narrative
  + Resolve with the narrator apologising for their confusion and asking for clarification by guiding them towards one of the possible narrative options they were presented with
* Incorrect or invalid input
  + Results in an inability for the program to progress with the narrative
  + Resolve with witty remark regarding the poor writing skills of the author and attempt to direct the user towards a valid option

## Action / Response Generation:

Responses will for the most part be pre-scripted, with the possibility to include extra story-irrelevant details mentioned by the user. For example:

* Expected: Run to the light
* User responds: Walk to the light
* Narrative response: Character **walks** to the light

The exact responses will depend on the narration previously mentioned, and the decisions made by the user. This will be modelled as a state-machine.

## User Evaluations:

It is recognised that the speech to text process is likely to be the most error prone. As such, evaluation will be completed with two experiments. The first experiment will have a human translator performing the role of the speech to text translation. The second experiment will directly use the speech to text translation software with a different set of users. This allows for an independent analysis of the core system with and without the speech to text errors.

Assessments will record:

* The number of successful and failed dialogues
* The type of interaction (yes / no, or [in]direct sentence responses)
* The frequency of each ending occurrence