Contract

Description

This ”contract” contains everything that you can assume about the inputs to the various pieces of the ballot function, as well as what is required of you on output. Anything not specified are things that you cannot be ”sure” of, so you should not use them (for example one should remove the todos in the old single voting models). However, creating and using variables within sections is fine, as long as other sections do not modify them. In general, one can put utility declarations below the productions they refer to in the same problem (or really anywhere in the file, I think?).

Combine.lisp

This Lisp file contains functions to combine all of the following different categories into a single model and load the relevant ballot (experiment).

Contained within are the parameters, the chunk types, and the starting goal (these things are common to all models, or at least are for now).

Also contains direct folder locations that can be changed depending on the system.

Ballot

Contains the code necessary to create the ballot and run the model (i.e. an experiment). Can have multiple definitions of functions but at least one should be called vote (the main function to run) with parameters (realtime use-model).

Memory

Contains any number of declarative memory entries (lisp s-expressions). Can make use of chunk types in combine.lisp.

Macronavigation

Contains the productions for navigating from race to race.

Goal
Input:
State: Start-voting, Find-race
Left: Left of last race, -1 if start-voting
Right: Right of last race, -1 if start-voting

Output:
State: Storing-race-group, End
Left: Left of current race

Visual Location
Output: a race-header location in the visual location buffer

Visual
Input: attention somewhere in the last race (although it seems that it may be alright if this buffer is cleared as the model still works but best to be safe)

Imaginal
Input: the race, candidate, and party groups of the last race, in their respective fields (a full imaginal buffer from the race just voted on)
Output: an imaginal buffer with none in each of the group fields

Encoding

Contains the productions for encoding the race, party, and candidate groups for a specific race.

Goal
Input:
State: Storing-race-group
Left: Left of current race

Output:
State: Ready-to-make-choice
Left: Left of current race
Right: Right of current race

Visual Location
Input: a race-header location in the visual location buffer

Imaginal
Input: an imaginal buffer with none in each of the group fields
Output: an imaginal buffer with race, candidate, and party groups for the current race filled in

Micronavigation

Contains the productions for finding the candidate to vote for within this race. Either sends the model to look for a new race if abstaining (find-next-race in the goal module), or attends a candidate and prepares to vote for it (moved-to-candidate in the goal module).

Goal
Input:
State: Ready-To-Make-Choice
Left: Left of race
Right: Right of race

Output:
State: Moved-to-candidate, Find-next-race
Left: Left of race
Right: Right of race

Visual
Output: attention on the candidate or party one wants to vote for, or in the case of abstaining attention somewhere in the current race

Imaginal
Input: an imaginal buffer with race, candidate, and party groups for the current race filled in
Output: an imaginal buffer with race, candidate, and party groups for the current race filled in

Click

Contains the productions for clicking on the bubble corresponding to the candidate to vote on.

Goal
Input:
State: Moved-To-Candidate
Left: Left of race
Right: Right of race

Output:
State: Find-next-race
Left: Left of race
Right: Right of race

Visual
Input: attention on the candidate or party one wants to vote for
Output: attention somewhere in the race (it could be in the same position as the beggining)

Imaginal
Input: an imaginal buffer with race, candidate, and party groups for the current race filled in
Output: an imaginal buffer with race, candidate, and party groups for the current race filled in