CAUSAL COGNITIVE ARCHITECTURE 3 (CCA3): A SOLUTION TO THE BINDING PROBLEM

Howard Schneider

Sheppard Clinic North, Ontario, Canada

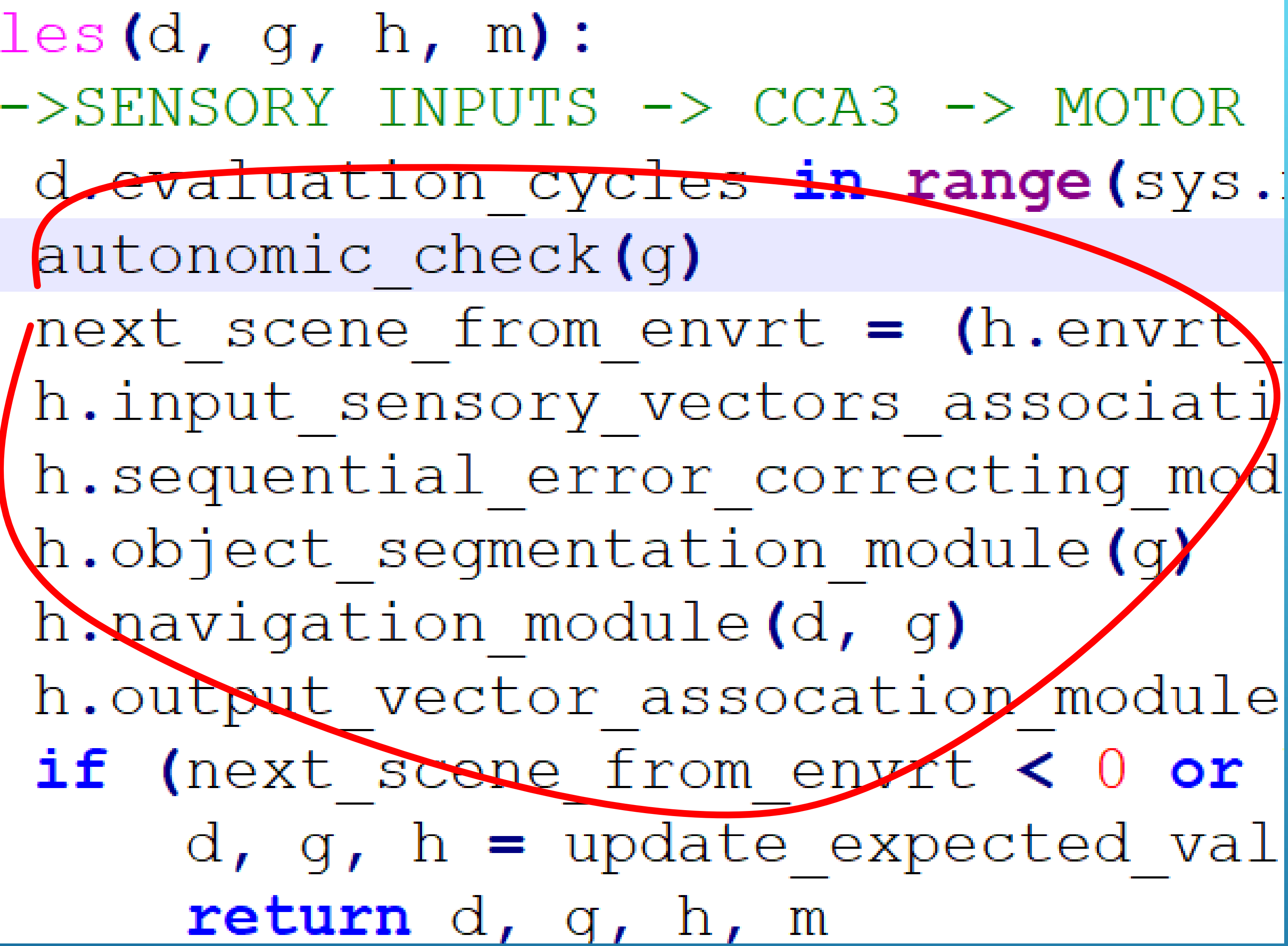
Cognitive Systems Research, in press
Supplementary Video File

GITHUB Username: "CausalCog"
https://github.com/CausalCog

VIDEO #3

- CCA3 Overview ✓
- Binding Problem Overview ✓
- Software Overview ✓
- Operations Overview ←
- Operations Causal
- Software in More Detail
- More videos, code on GitHub "CausalCog"

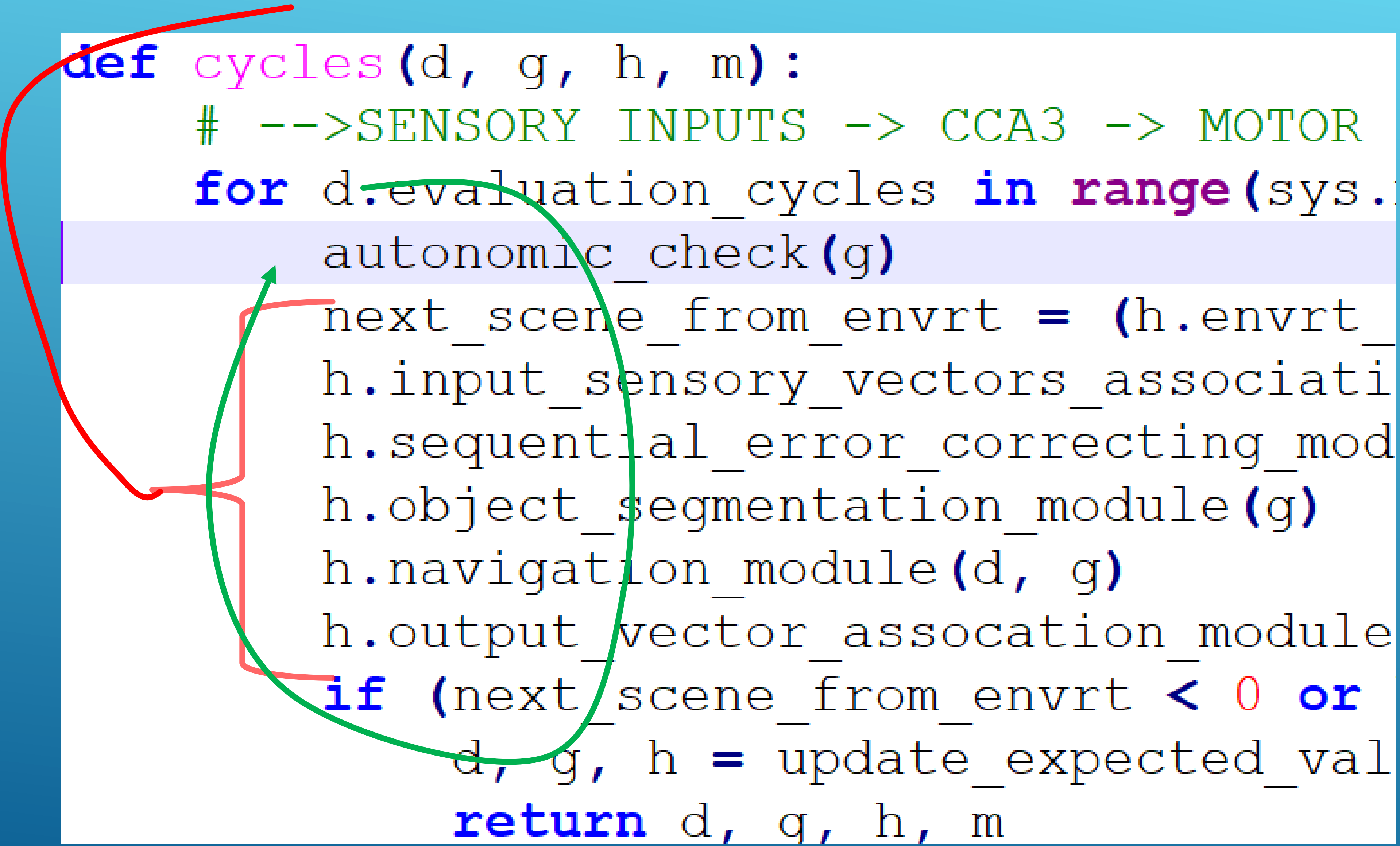
(If interest, continued updating on GitHub)



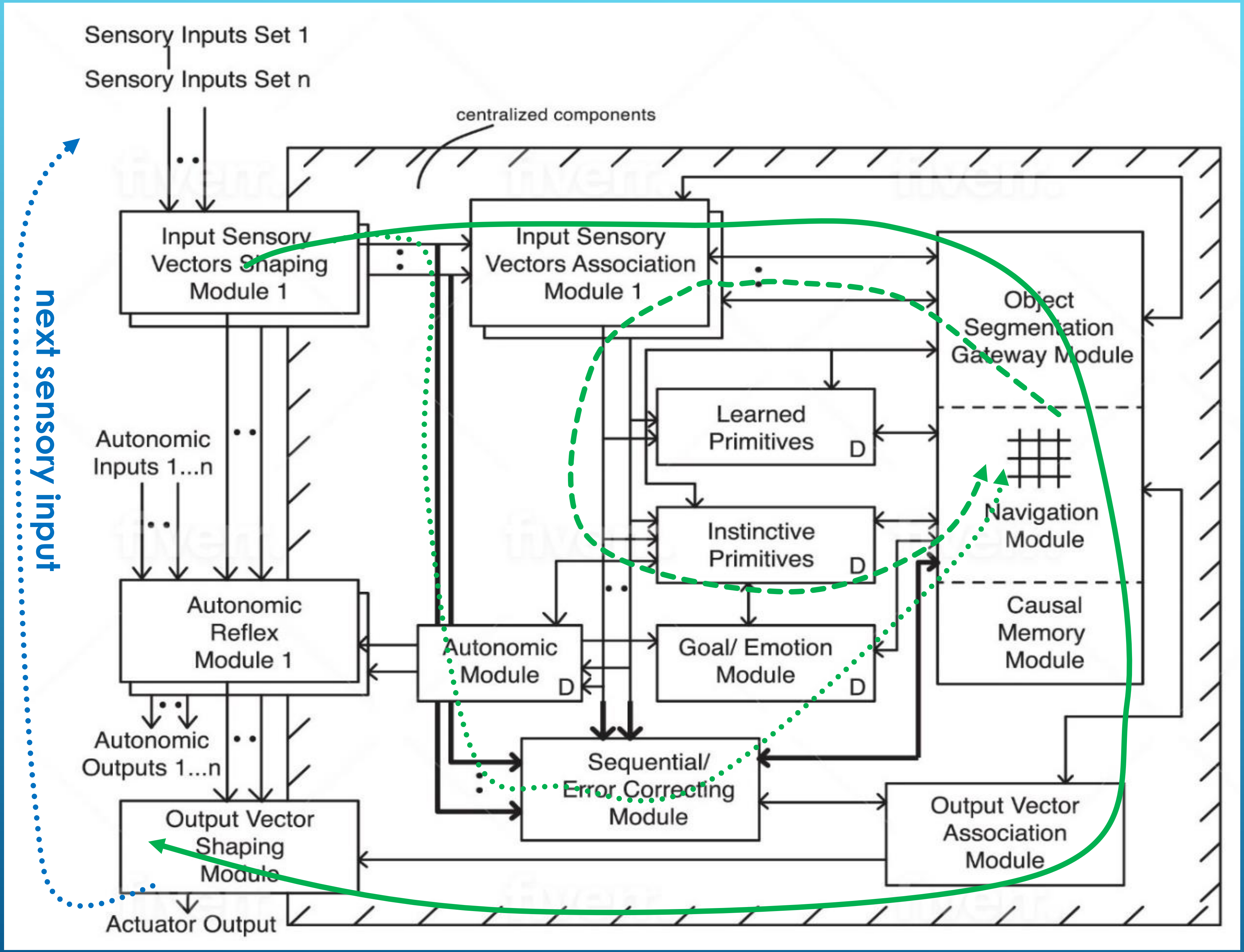
```
les(d, g, h, m):
->SENSORY INPUTS -> CCA3 -> MOTOR
 d.evaluation_cycles in range(sys.
 autonomic_check(g)
 next_scene_from_envrt = (h.envrt_
 h.input_sensory_vectors_associati
 h.sequential_error_correcting_mod
 h.object_segmentation_module(g)
 h.navigation_module(d, g)
 h.output_vector_assocation_module
 if (next_scene_from_envrt < 0 or
     d, g, h = update_expected_val
     return d, g, h, m
```

Now, let's look at each of these methods that occur each cycle, and how they relate to the equations of the paper....

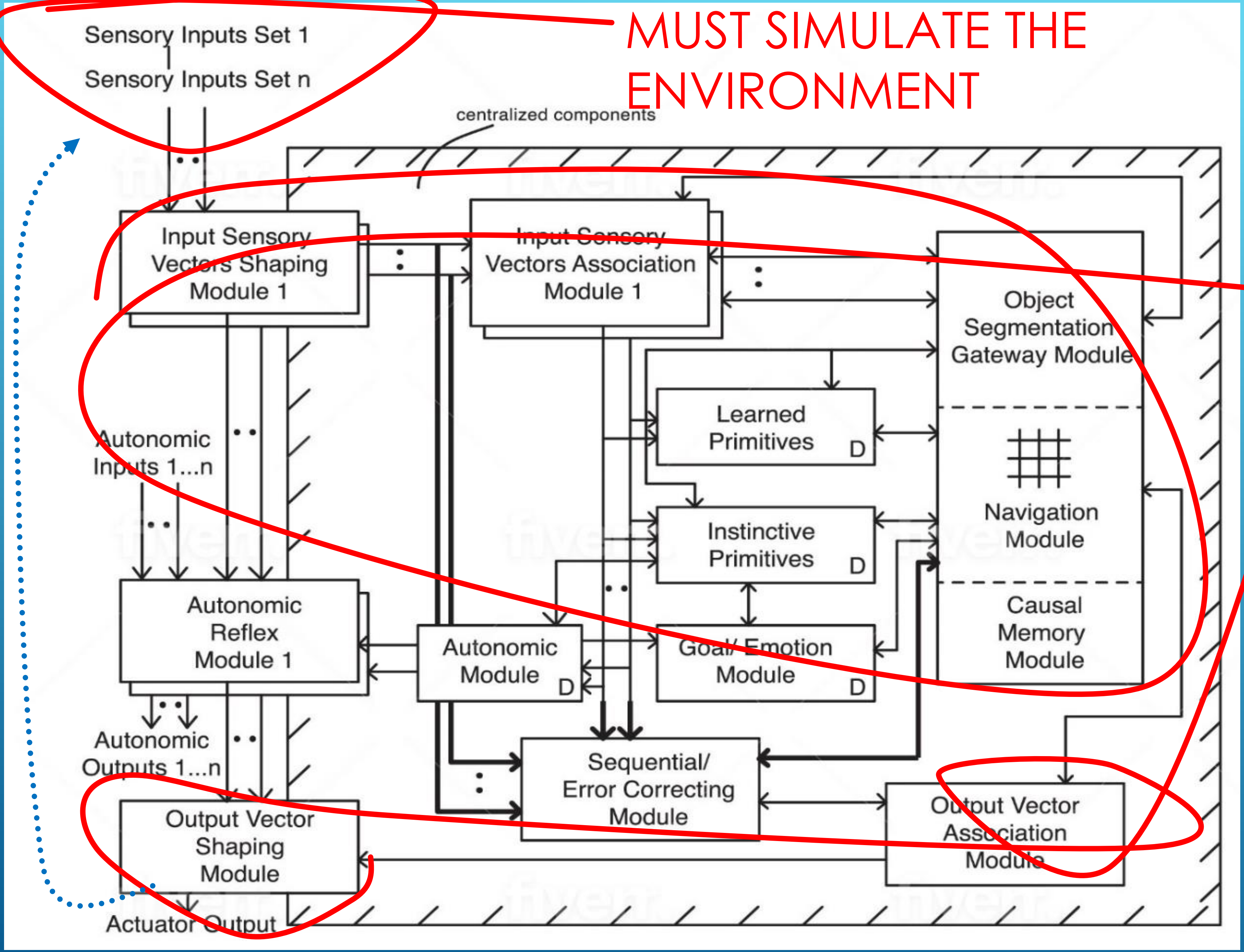
Equations 3 – 92 each "cycle"



```
def cycles(d, g, h, m):
    # -->SENSORY INPUTS -> CCA3 -> MOTOR
    for d.evaluation_cycles in range(sys.
        autonomic_check(g)
        next_scene_from_envrt = (h.envrt_
        h.input_sensory_vectors_associati
        h.sequential_error_correcting_mod
        h.object_segmentation_module(g)
        h.navigation_module(d, g)
        h.output_vector_assocation_module
        if (next_scene_from_envrt < 0 or
            d, g, h = update_expected_val
            return d, g, h, m
```



"cycle" == "processing cycle"



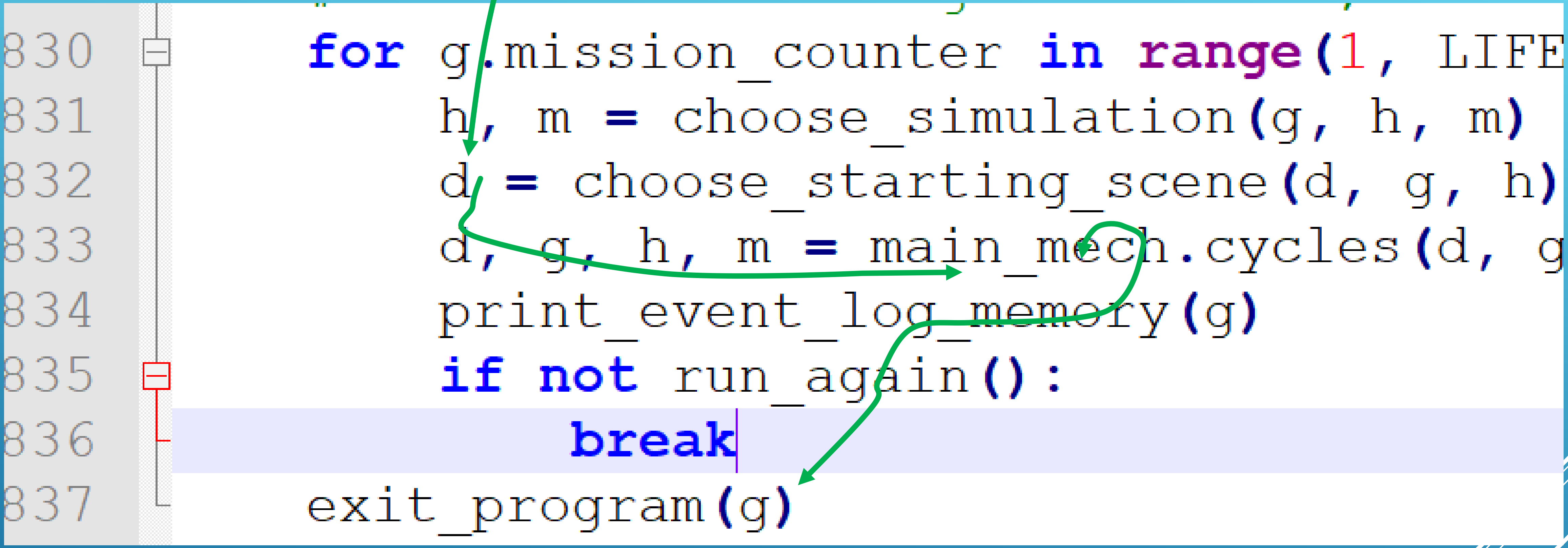
MUST SIMULATE THE ENVIRONMENT

MUST SIMULATE THE CCA3

MUST SIMULATE THE OUTPUTS AND THE EFFECT ON THE ENVIRONMENT

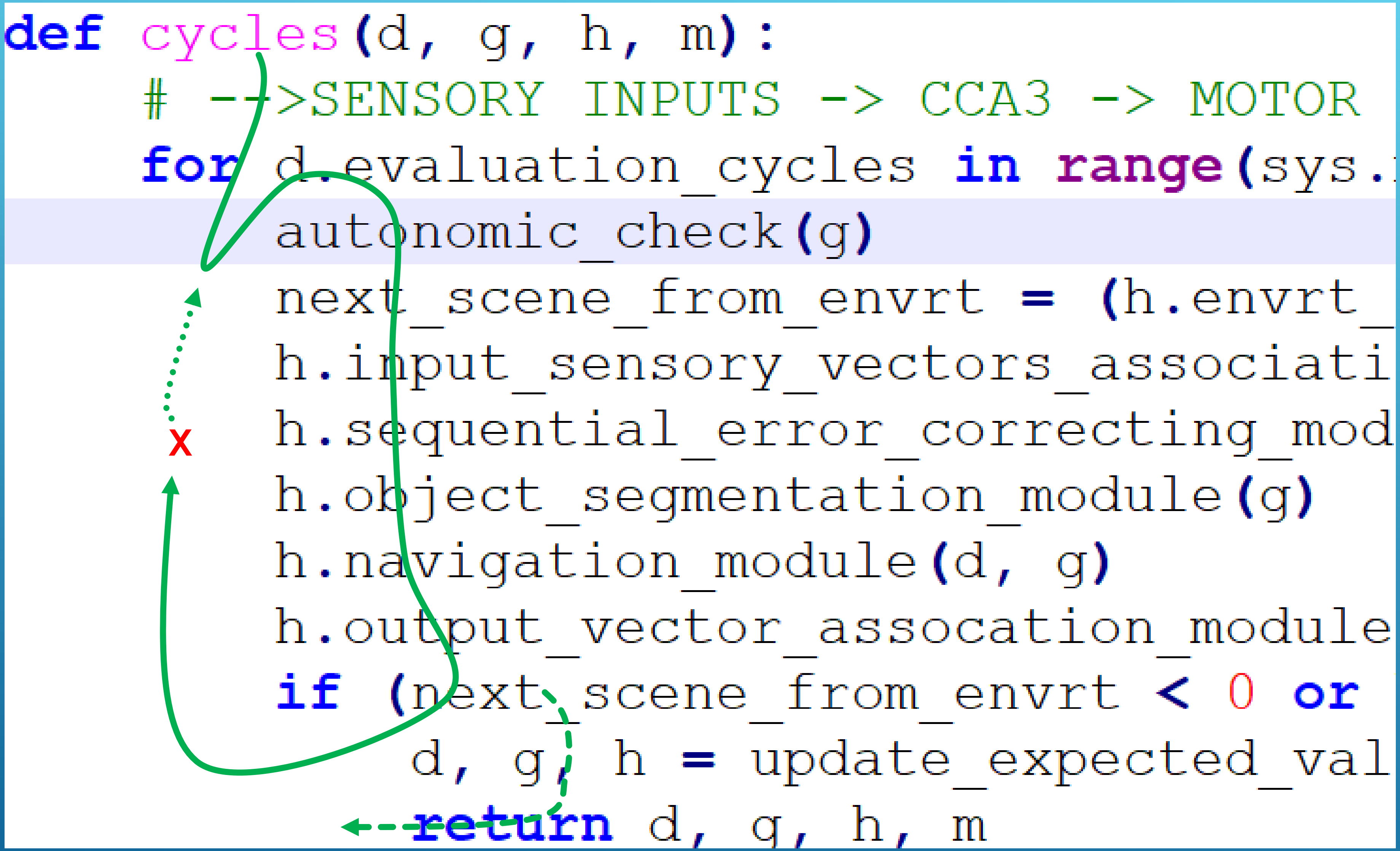
CCA3 SIMULATION SOFTWARE – OPERATIONS OVERVIEW

"main()" of CCA3



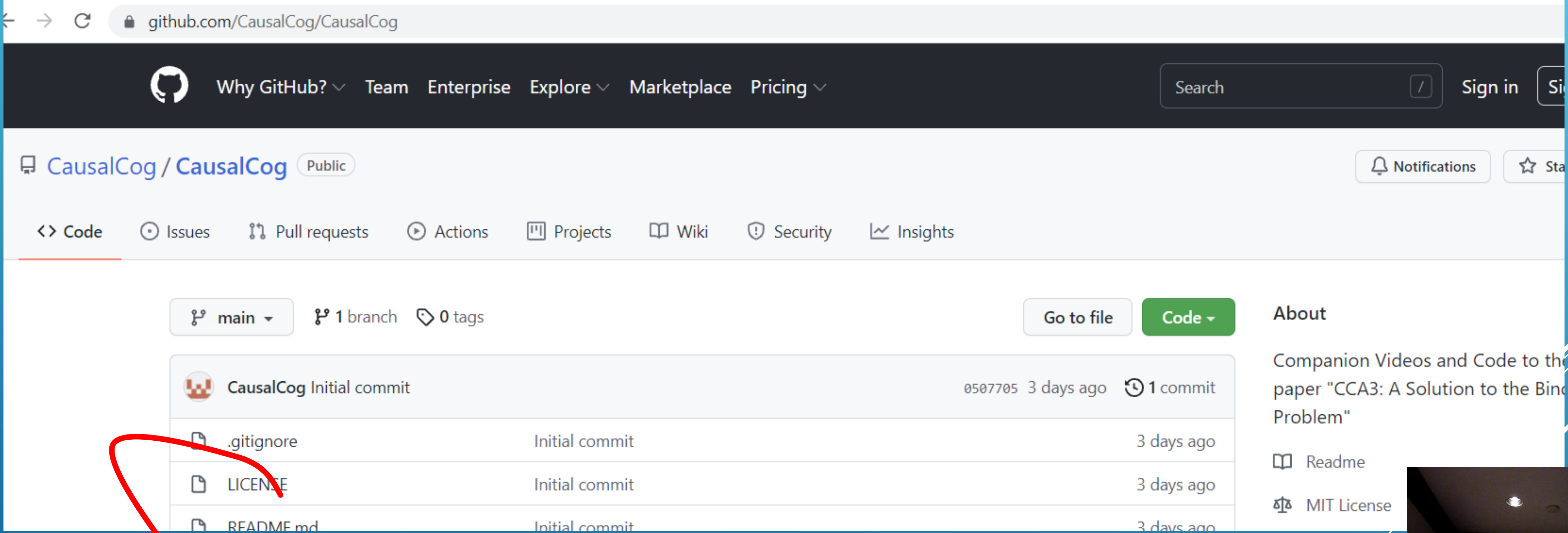
```
    for g.mission_counter in range(1, LIFE
        h, m = choose_simulation(g, h, m)
        d = choose_starting_scene(d, g, h)
        d, g, h, m = main_mech.cycles(d, g
        print_event_log_memory(g)
        if not run_again():
            break
    exit_program(g)
```

main_mech.cycles()



```
def cycles(d, g, h, m):
    # -->SENSORY INPUTS -> CCA3 -> MOTOR
    for d.evaluation_cycles in range(sys.
        autonomic_check(g)
        next_scene_from_envrt = (h.envrt_
        h.input_sensory_vectors_associati
        h.sequential_error_correcting_mod
        h.object_segmentation_module(g)
        h.navigation_module(d, g)
        h.output_vector_assocation_module
        if (next_scene_from_envrt < 0 or
            d, g, h = update_expected_val
            return d, g, h, m
```

Copy all files (listed in requirements.txt) into a Python 3.9 environment, install non-PyPI imports



github.com/CausalCog/CausalCog

CausalCog / CausalCog Public

1 branch 0 tags

CausalCog Initial commit	0507705 3 days ago	1 commit
.gitignore	Initial commit	3 days ago
LICENSE	Initial commit	3 days ago
README.md	Initial commit	3 days ago

About

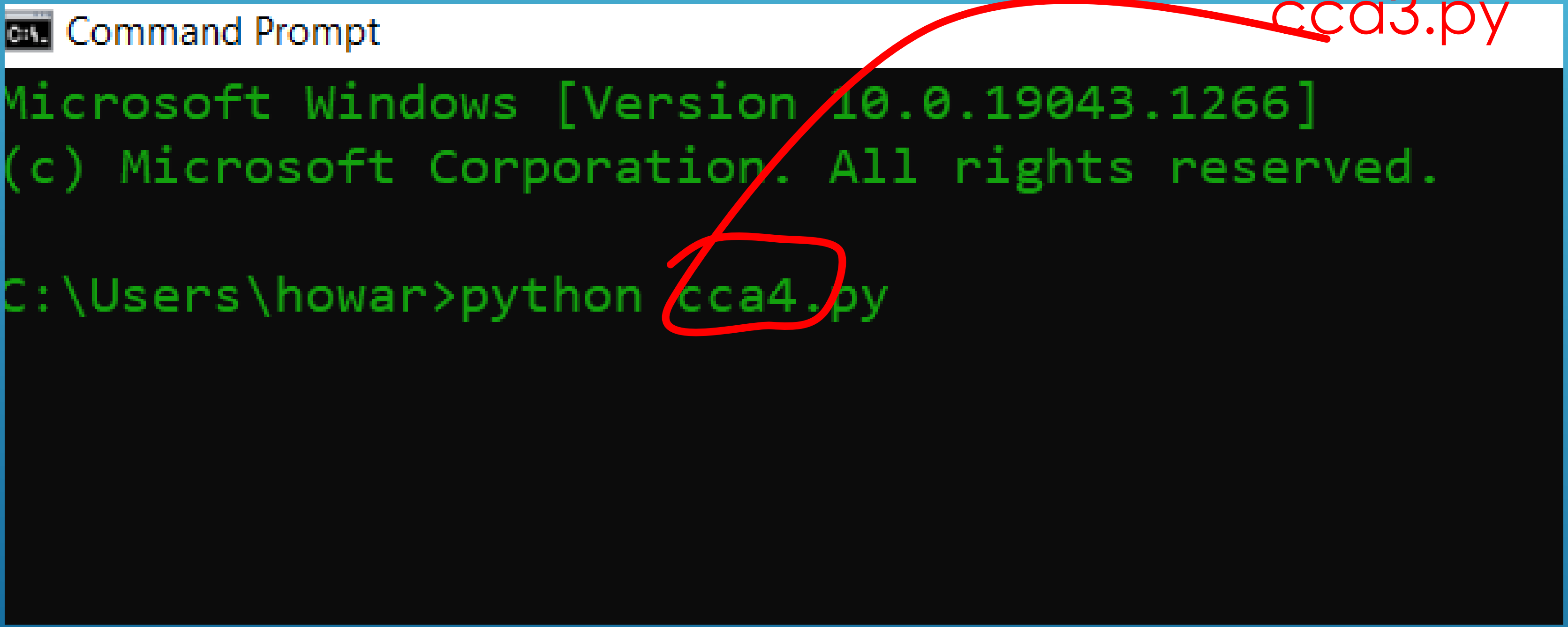
Companion Videos and Code to the paper "CCA3: A Solution to the Bin Problem"

Readme

MIT License

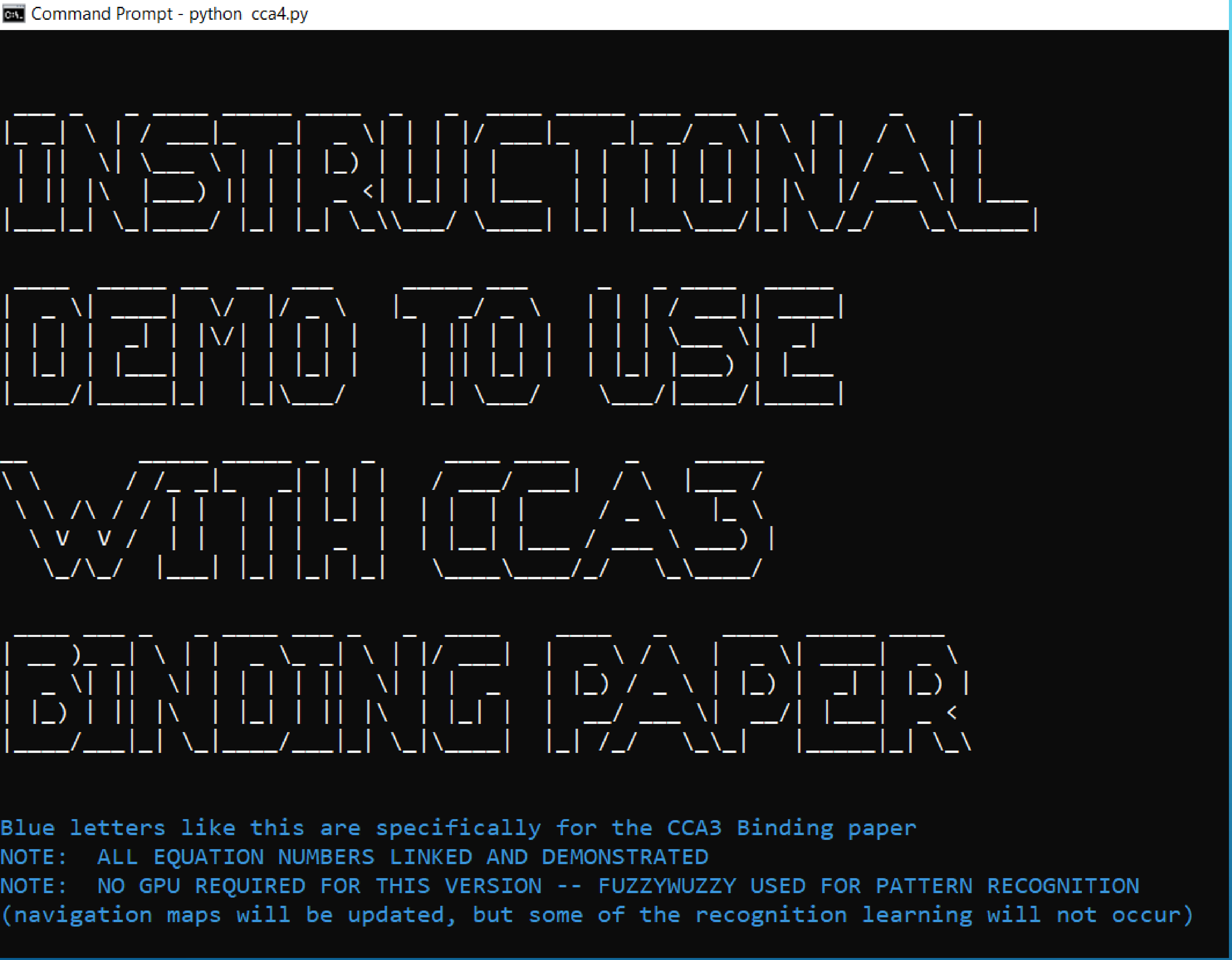
cca4.py (special cca3.py version created for paper)

Run code – we will consider here one processing cycle of operations

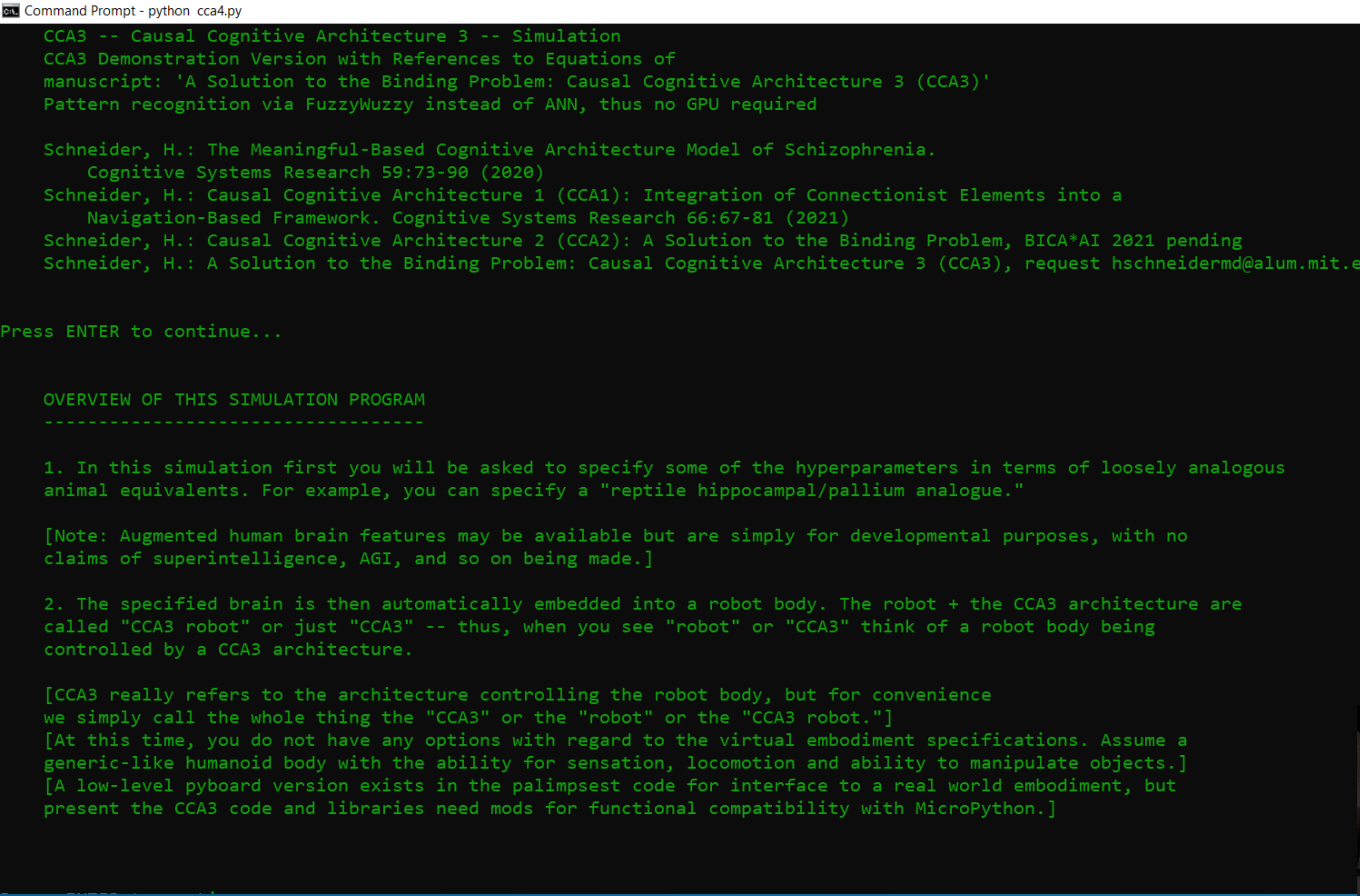


```
Microsoft Windows [Version 10.0.19043.1266]
(c) Microsoft Corporation. All rights reserved.

C:\Users\howar>python cca4.py
```

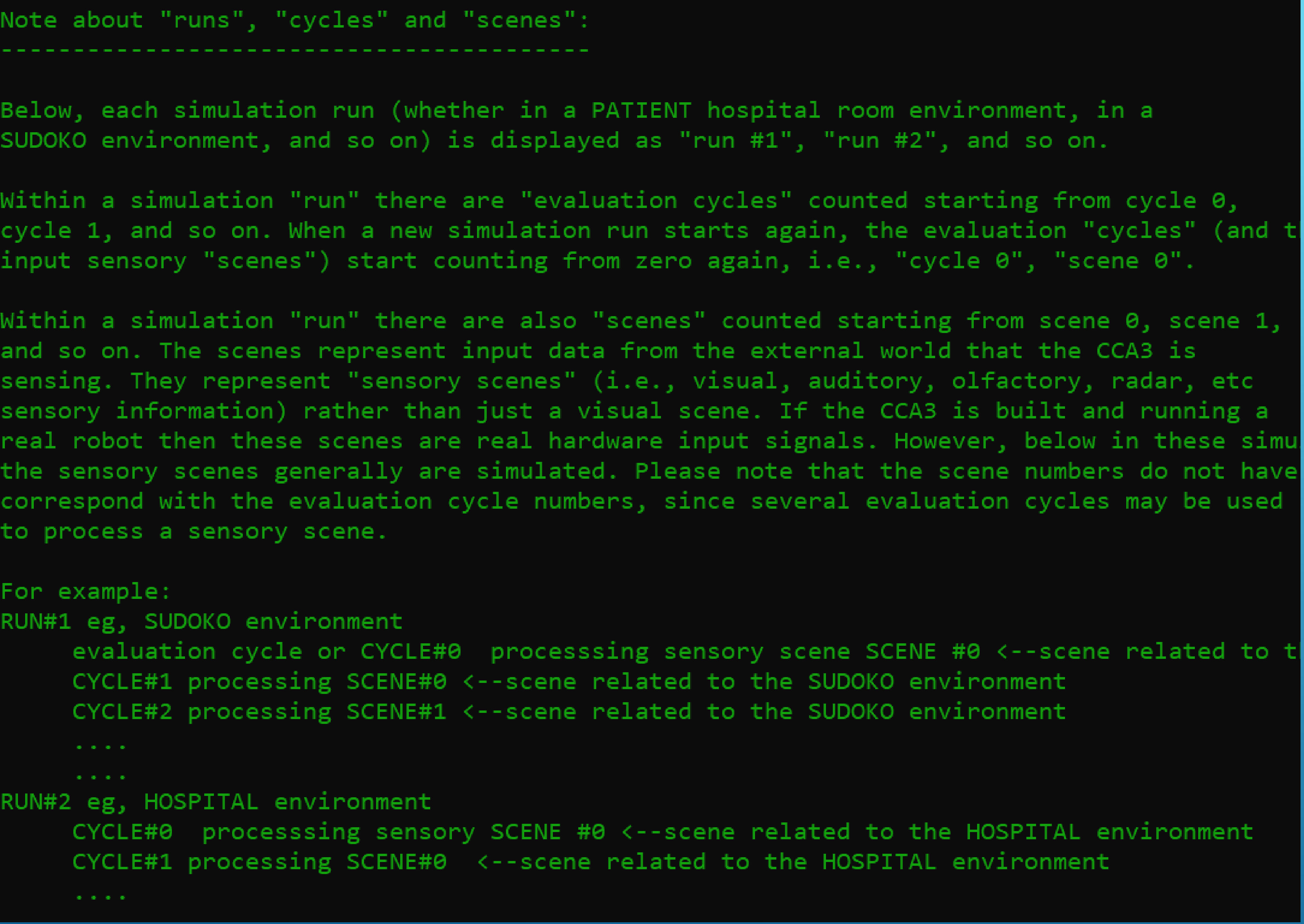
special version of cca3.py



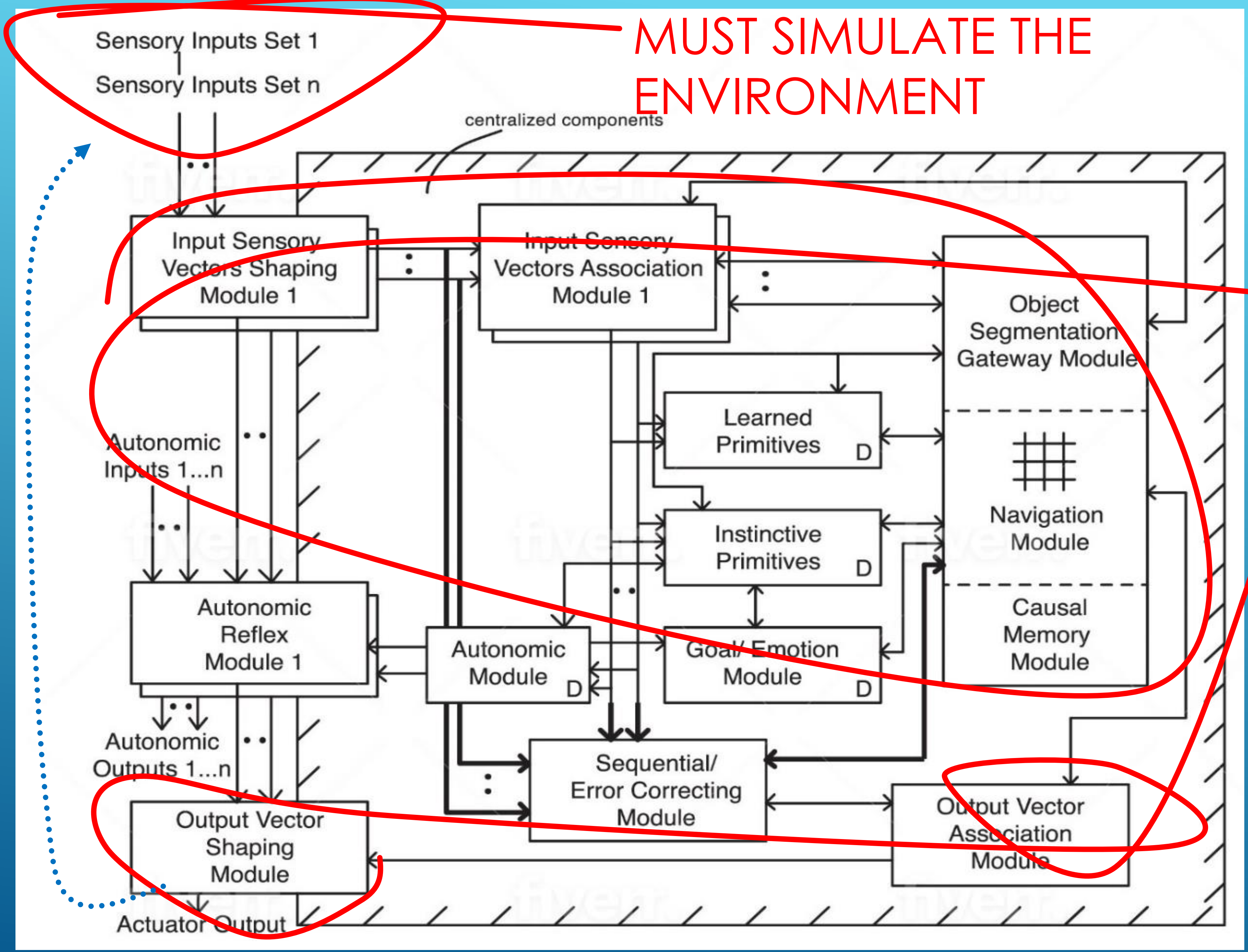
```
 ___ _   _ ____ _____ ____  _   _  ____ _____ ___ ___  _   _    _    _
|_ _| \ | / ___|_   _|  _ \| | | |/ ___|_   _|_ _/ _ \| \ | |  / \  | |
 | ||  \| \___ \ | | | |_) | | | | |     | |  | | | | |  \| | / _ \ | |
 | || |\  |___) || | |  _ <| |_| | |___  | |  | | |_| | |\  |/ ___ \| |___
|___|_| \_|____/ |_| |_| \_\\___/ \____| |_| |___\___/|_| \_/_/   \_\_____|

 ____  _____ __  __  ___    _____ ___    _   _ ____  _____
|  _ \| ____|  \/  |/ _ \  |_   _/ _ \  | | | / ___|| ____|
| | | |  _| | |\/| | | | |   | || | | | | | | \___ \|  _|
| |_| | |___| |  | | |_| |   | || |_| | | |_| |___) | |___
|____/|_____|_|  |_|\___/    |_| \___/   \___/|____/|_____|

__        _____ _____ _   _    ____ ____    _    _____
\ \      / /_ _|_   _| | | |  / ___/ ___|  / \  |___ /
 \ \ /\ / / | |  | | | |_| | | |  | |     / _ \   |_ \
  \ V  V /  | |  | | |  _  | | |__| |___ / ___ \ ___) |
   \_/\_/  |___| |_| |_| |_|  \____\____/_/   \_\____/

 ____ ___ _   _ ____ ___ _   _  ____   ____   _    ____  _____ ____
| __ )_ _| \ | |  _ \_ _| \ | |/ ___| |  _ \ / \  |  _ \| ____|  _ \
|  _ \| ||  \| | | | | ||  \| | |  _  | |_) / _ \ | |_) |  _| | |_) |
| |_) | || |\  | |_| | || |\  | |_| | |  __/ ___ \|  __/| |___|  _ <
|____/___|_| \_|____/___|_| \_|\____| |_| /_/   \_\_|   |_____|_| \_\

Blue letters like this are specifically for the CCA3 Binding paper
NOTE:  ALL EQUATION NUMBERS LINKED AND DEMONSTRATED
NOTE:  NO GPU REQUIRED FOR THIS VERSION -- FUZZYWUZZY USED FOR PATTERN RECOGNITION
(navigation maps will be updated, but some of the recognition learning will not occur)
```



```
    CCA3 -- Causal Cognitive Architecture 3 -- Simulation
    CCA3 Demonstration Version with References to Equations of
    manuscript: 'A Solution to the Binding Problem: Causal Cognitive Architecture 3 (CCA3)'
    Pattern recognition via FuzzyWuzzy instead of ANN, thus no GPU required

    Schneider, H.: The Meaningful-Based Cognitive Architecture Model of Schizophrenia.
        Cognitive Systems Research 59:73-90 (2020)
    Schneider, H.: Causal Cognitive Architecture 1 (CCA1): Integration of Connectionist Elements into a
        Navigation-Based Framework. Cognitive Systems Research 66:67-81 (2021)
    Schneider, H.: Causal Cognitive Architecture 2 (CCA2): A Solution to the Binding Problem, BICA*AI 2021 pending
    Schneider, H.: A Solution to the Binding Problem: Causal Cognitive Architecture 3 (CCA3), request hschneidermd@alum.mit.edu


Press ENTER to continue...


    OVERVIEW OF THIS SIMULATION PROGRAM
    -----------------------------------

    1. In this simulation first you will be asked to specify some of the hyperparameters in terms of loosely analogous
    animal equivalents. For example, you can specify a "reptile hippocampal/pallium analogue."

    [Note: Augmented human brain features may be available but are simply for developmental purposes, with no
    claims of superintelligence, AGI, and so on being made.]

    2. The specified brain is then automatically embedded into a robot body. The robot + the CCA3 architecture are
    called "CCA3 robot" or just "CCA3" -- thus, when you see "robot" or "CCA3" think of a robot body being
    controlled by a CCA3 architecture.

    [CCA3 really refers to the architecture controlling the robot body, but for convenience
    we simply call the whole thing the "CCA3" or the "robot" or the "CCA3 robot."]
    [At this time, you do not have any options with regard to the virtual embodiment specifications. Assume a
    generic-like humanoid body with the ability for sensation, locomotion and ability to manipulate objects.]
    [A low-level pyboard version exists in the palimpsest code for interface to a real world embodiment, but
    present the CCA3 code and libraries need mods for functional compatibility with MicroPython.]
```



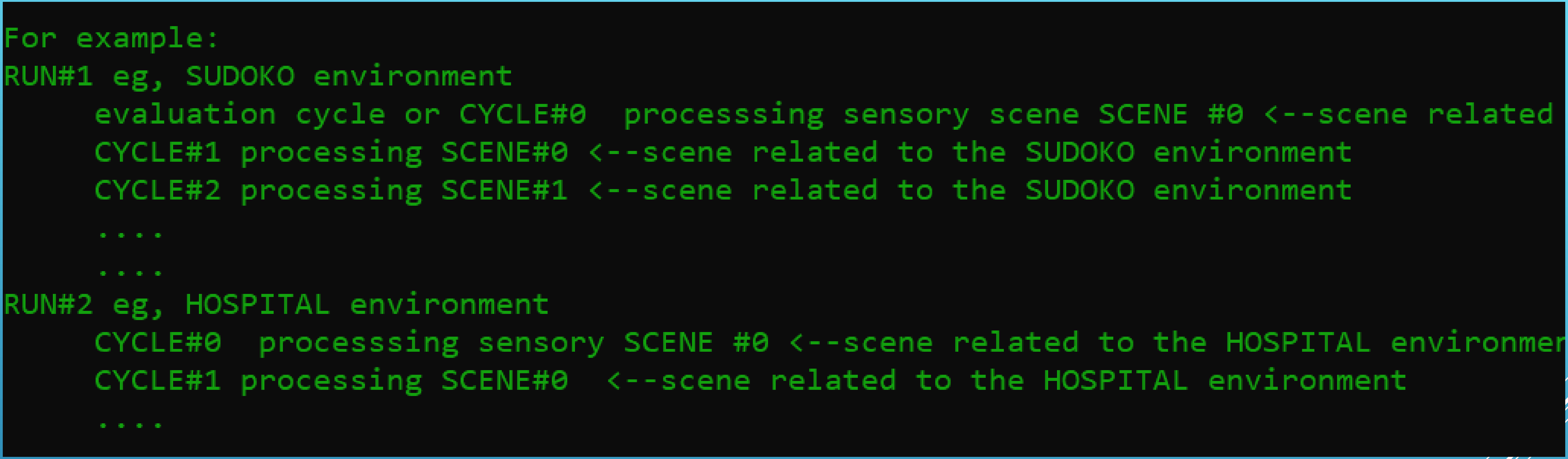
```
Note about "runs", "cycles" and "scenes":
-----------------------------------------

Below, each simulation run (whether in a PATIENT hospital room environment, in a
SUDOKO environment, and so on) is displayed as "run #1", "run #2", and so on.

Within a simulation "run" there are "evaluation cycles" counted starting from cycle 0,
cycle 1, and so on. When a new simulation run starts again, the evaluation "cycles" (and t
input sensory "scenes") start counting from zero again, i.e., "cycle 0", "scene 0".

Within a simulation "run" there are also "scenes" counted starting from scene 0, scene 1,
and so on. The scenes represent input data from the external world that the CCA3 is
sensing. They represent "sensory scenes" (i.e., visual, auditory, olfactory, radar, etc
sensory information) rather than just a visual scene. If the CCA3 is built and running a
real robot then these scenes are real hardware input signals. However, below in these simu
the sensory scenes generally are simulated. Please note that the scene numbers do not have
correspond with the evaluation cycle numbers, since several evaluation cycles may be used
to process a sensory scene.

For example:
RUN#1 eg, SUDOKO environment
     evaluation cycle or CYCLE#0  processsing sensory scene SCENE #0 <--scene related to t
     CYCLE#1 processing SCENE#0 <--scene related to the SUDOKO environment
     CYCLE#2 processing SCENE#1 <--scene related to the SUDOKO environment
     ....
     ....
RUN#2 eg, HOSPITAL environment
     CYCLE#0  processsing sensory SCENE #0 <--scene related to the HOSPITAL environment
     CYCLE#1 processing SCENE#0  <--scene related to the HOSPITAL environment
     ....
```



MUST SIMULATE THE ENVIRONMENT

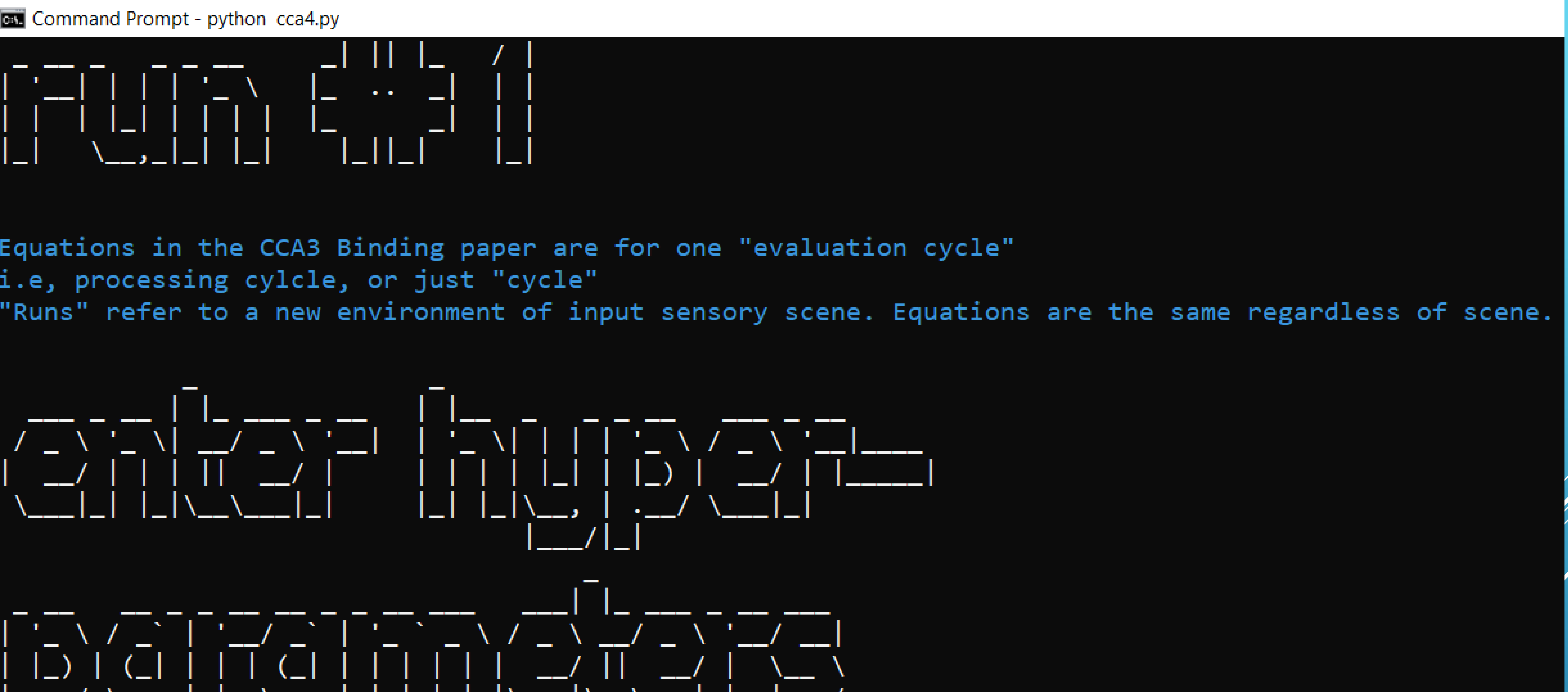
MUST SIMULATE THE CCA3

MUST SIMULATE THE OUTPUTS AND THE EFFECT ON THE ENVIRONMENT



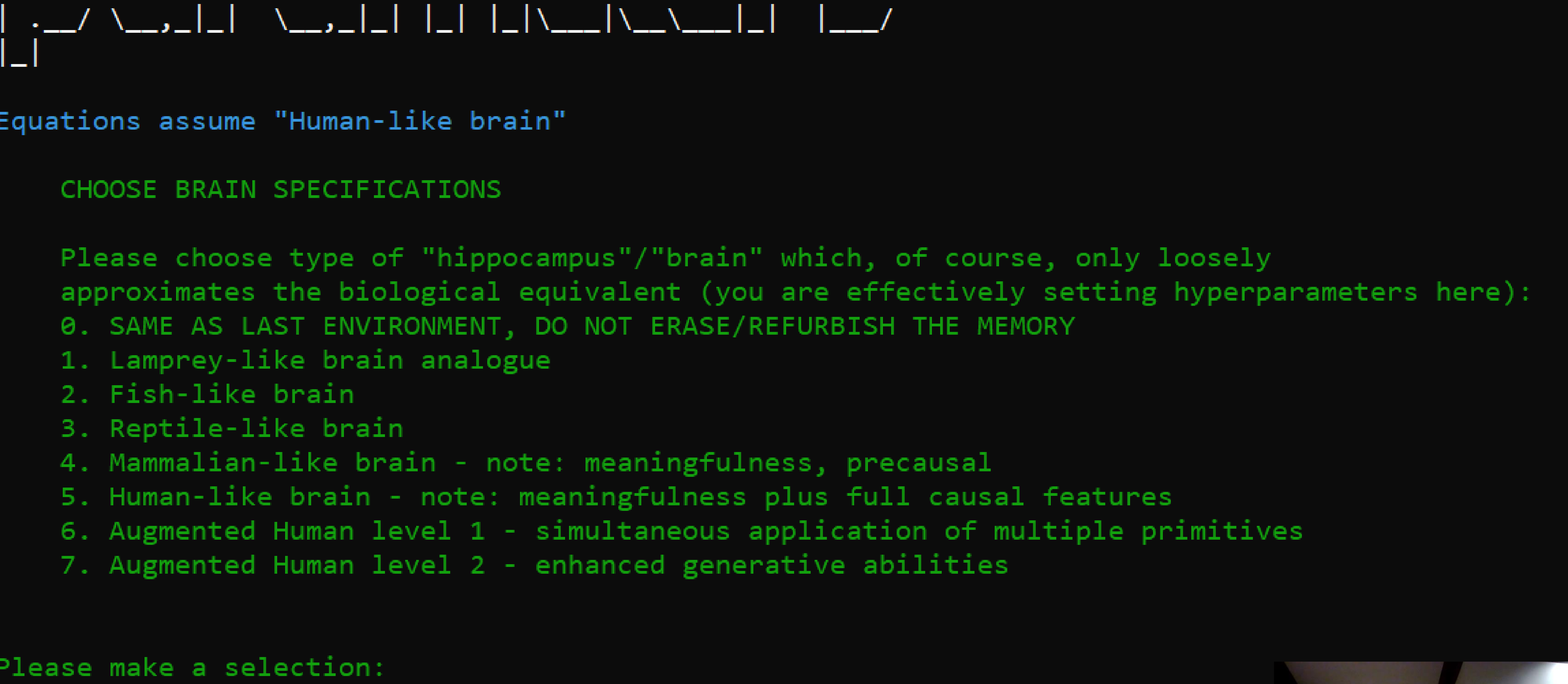
```
For example:
RUN#1 eg, SUDOKO environment
     evaluation cycle or CYCLE#0  processsing sensory scene SCENE #0 <--scene related
     CYCLE#1 processing SCENE#0 <--scene related to the SUDOKO environment
     CYCLE#2 processing SCENE#1 <--scene related to the SUDOKO environment
     ....
     ....
RUN#2 eg, HOSPITAL environment
     CYCLE#0  processsing sensory SCENE #0 <--scene related to the HOSPITAL environmen
     CYCLE#1 processing SCENE#0  <--scene related to the HOSPITAL environment
     ....
```

Just some examples to illustrate concept of “Run”, “Cycle”, “Scene”

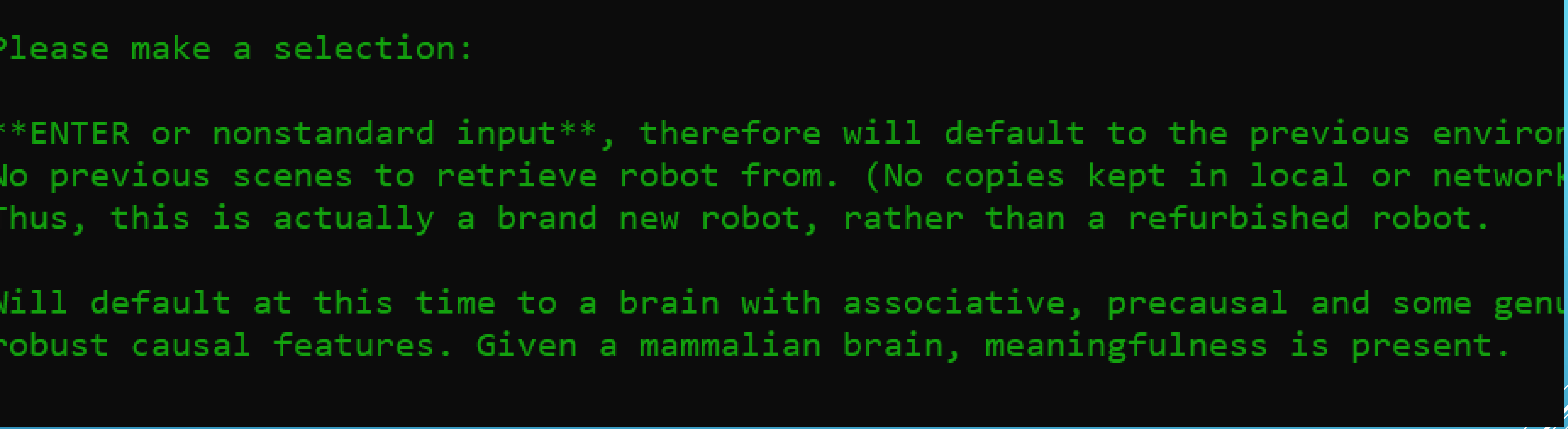


Command Prompt - python cca4.py

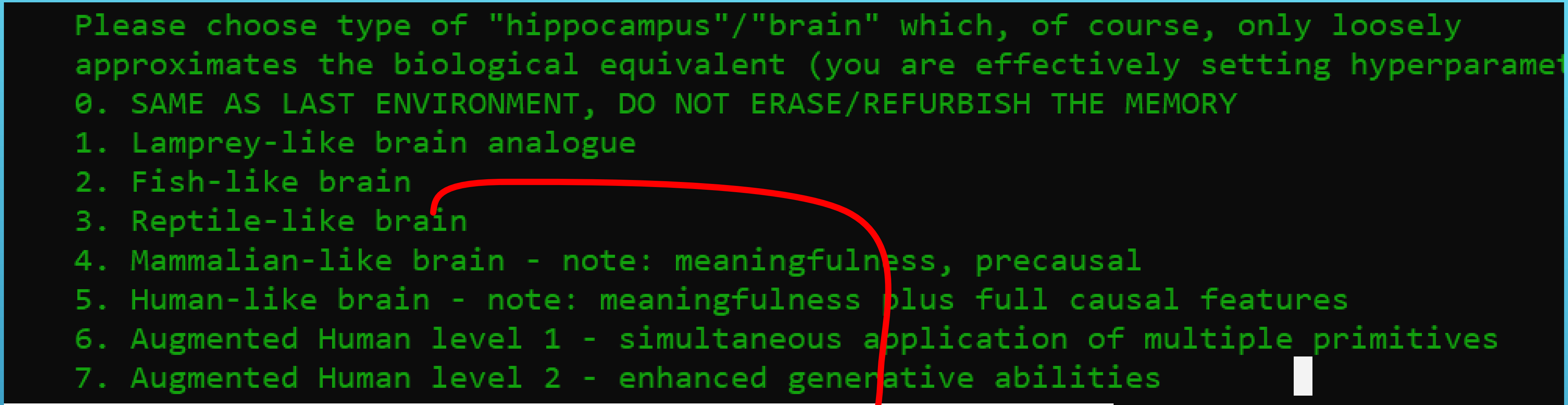
```
 _ __ _   _ _ __    _| || |_  / |
| '__| | | | '_ \  |_  ..  _| | |
| |  | |_| | | | | |_      _| | |
|_|   \__,_|_| |_|   |_||_|   |_|


Equations in the CCA3 Binding paper are for one "evaluation cycle"
i.e, processing cylcle, or just "cycle"
"Runs" refer to a new environment of input sensory scene. Equations are the same regardless of scene.

            _              _
  ___ _ __ | |_ ___ _ __  | |__  _   _ _ __   ___ _ __
 / _ \ '_ \| __/ _ \ '__| | '_ \| | | | '_ \ / _ \ '__|_____
|  __/ | | | ||  __/ |    | | | | |_| | |_) |  __/ | |_____|
 \___|_| |_|\__\___|_|    |_| |_|\__, | .__/ \___|_|
                                 |___/|_|
                                      _
 _ __   __ _ _ __ __ _ _ __ ___   ___| |_ ___ _ __ ___
| '_ \ / _` | '__/ _` | '_ ` _ \ / _ \ __/ _ \ '__/ __|
| |_) | (_| | | | (_| | | | | | |  __/ ||  __/ |  \__ \
| .__/ \__,_|_|  \__,_|_| |_| |_|\___|\__\___|_|  |___/
|_|

Equations assume "Human-like brain"

    CHOOSE BRAIN SPECIFICATIONS
```



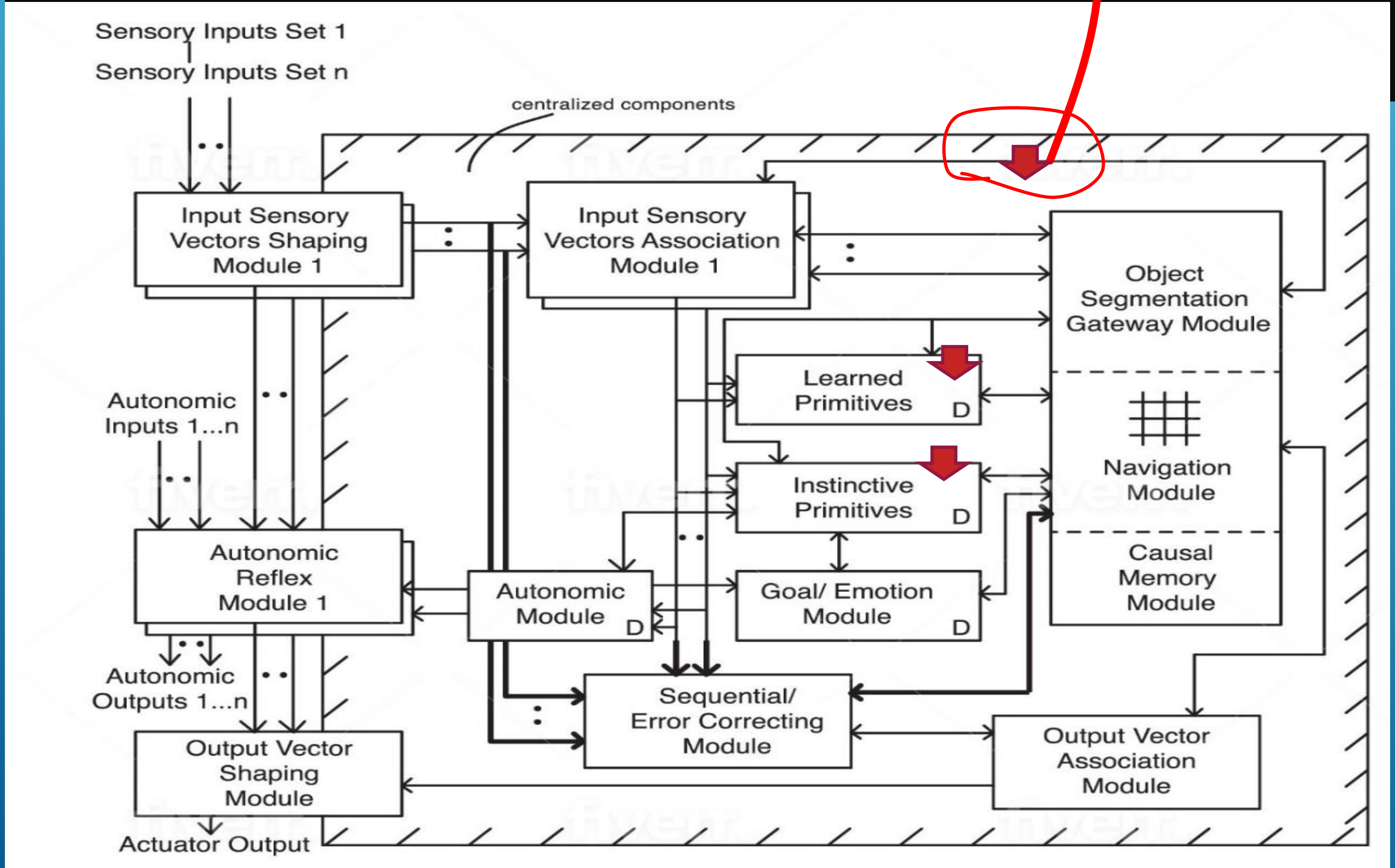
```
| .__/ \__,_|_|  \__,_|_| |_| |_|\___|\__\___|_|  |___/
|_|

Equations assume "Human-like brain"

    CHOOSE BRAIN SPECIFICATIONS

    Please choose type of "hippocampus"/"brain" which, of course, only loosely
    approximates the biological equivalent (you are effectively setting hyperparameters here):
    0. SAME AS LAST ENVIRONMENT, DO NOT ERASE/REFURBISH THE MEMORY
    1. Lamprey-like brain analogue
    2. Fish-like brain
    3. Reptile-like brain
    4. Mammalian-like brain - note: meaningfulness, precausal
    5. Human-like brain - note: meaningfulness plus full causal features
    6. Augmented Human level 1 - simultaneous application of multiple primitives
    7. Augmented Human level 2 - enhanced generative abilities


Please make a selection:
```



```
Please make a selection:

**ENTER or nonstandard input**, therefore will default to the previous environ
No previous scenes to retrieve robot from. (No copies kept in local or network
Thus, this is actually a brand new robot, rather than a refurbished robot.

Will default at this time to a brain with associative, precausal and some genu
robust causal features. Given a mammalian brain, meaningfulness is present.
```

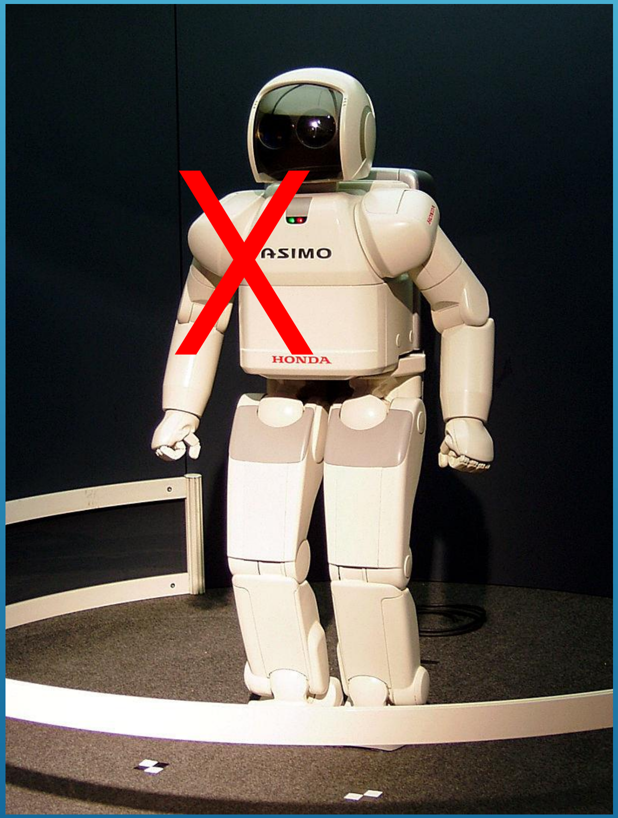


```
Please choose type of "hippocampus"/"brain" which, of course, only loosely
approximates the biological equivalent (you are effectively setting hyperparamet
0. SAME AS LAST ENVIRONMENT, DO NOT ERASE/REFURBISH THE MEMORY
1. Lamprey-like brain analogue
2. Fish-like brain
3. Reptile-like brain
4. Mammalian-like brain - note: meaningfulness, precausal
5. Human-like brain - note: meaningfulness plus full causal features
6. Augmented Human level 1 - simultaneous application of multiple primitives
7. Augmented Human level 2 - enhanced generative abilities
```



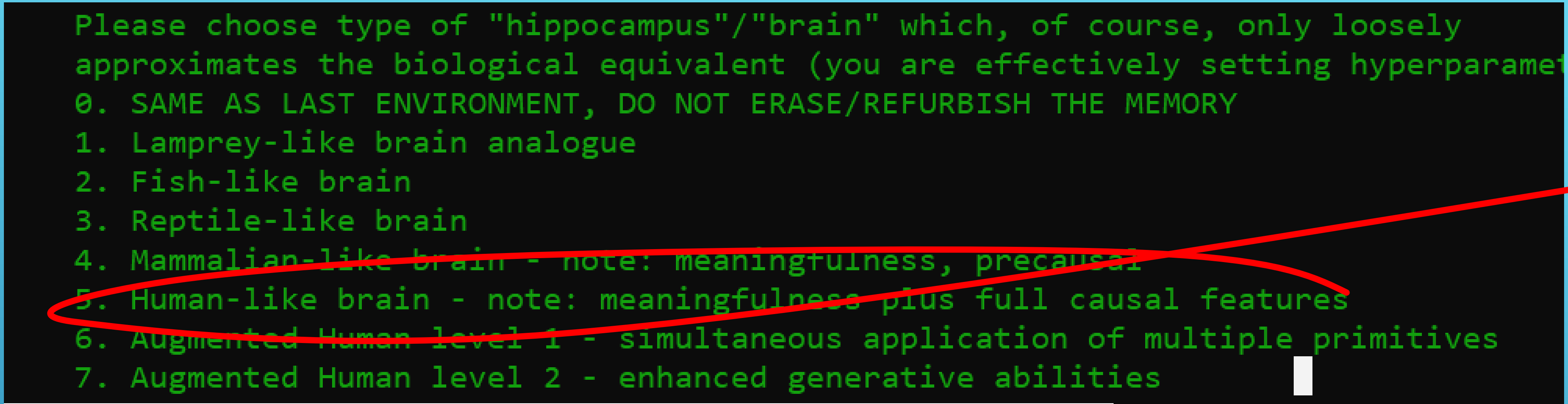
If allowed (which it is not now, since equations 3- 92 do not model this), how does this animal-like choice actually work?

No AGI (Artificial General Intelligence)…. just experiments



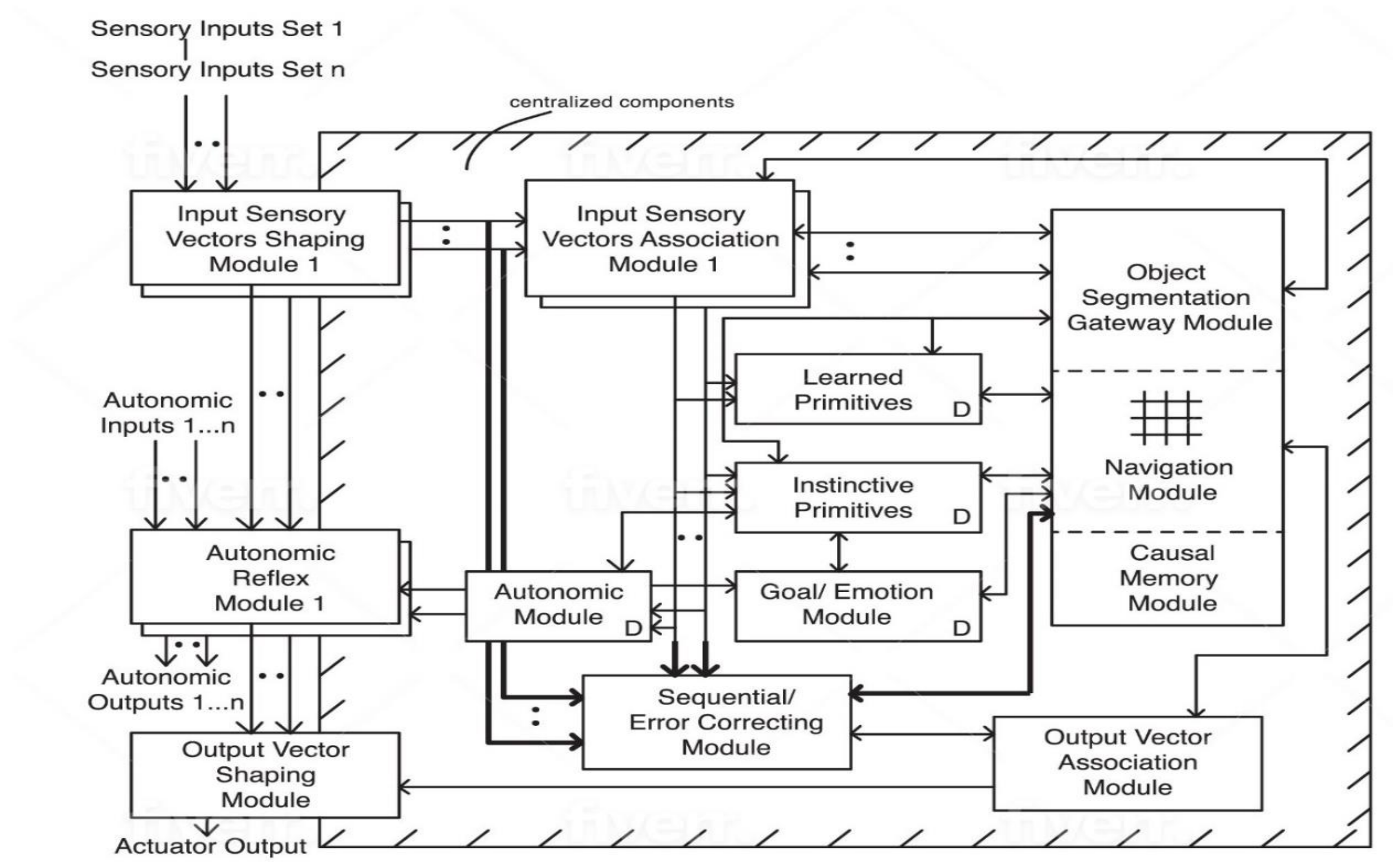
CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=140450

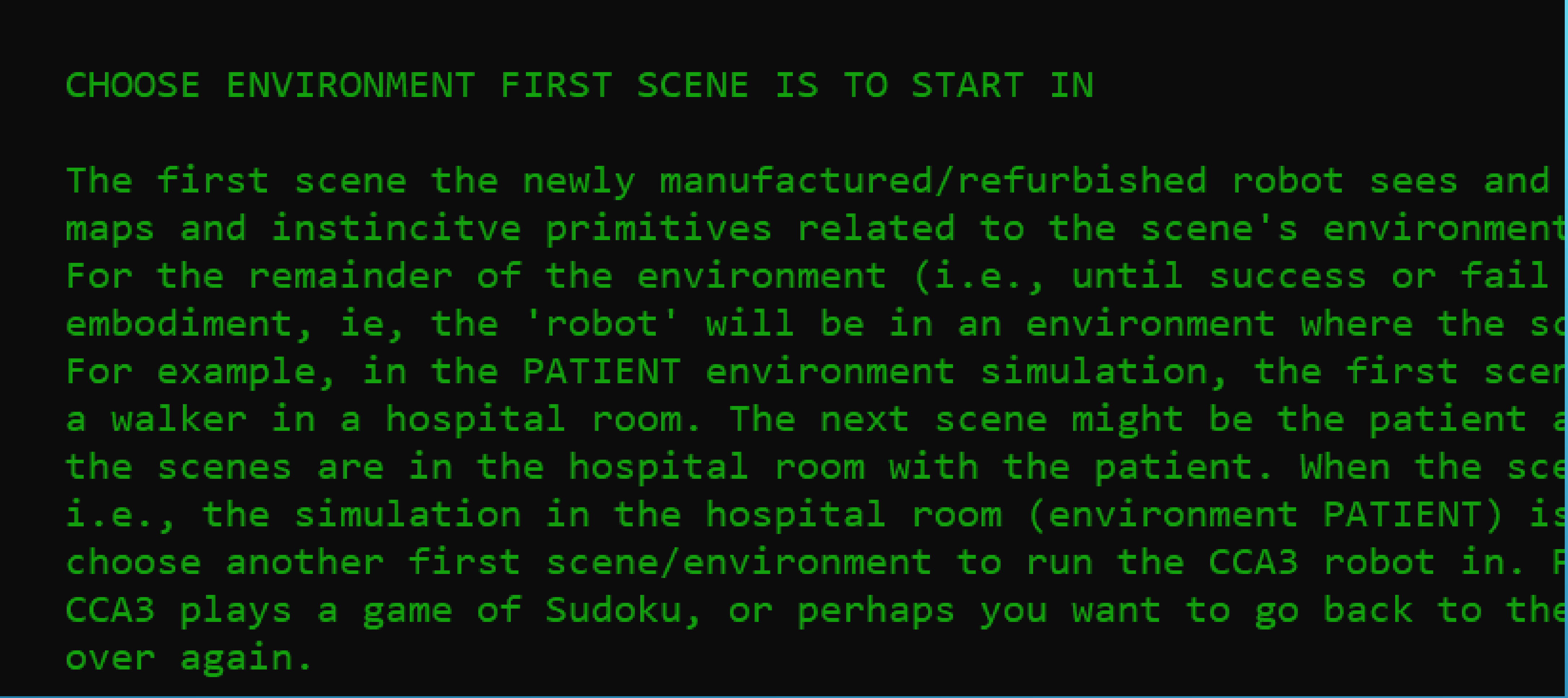
Rodin – Le Penseur CC BY-SA 4.0 wikipedia



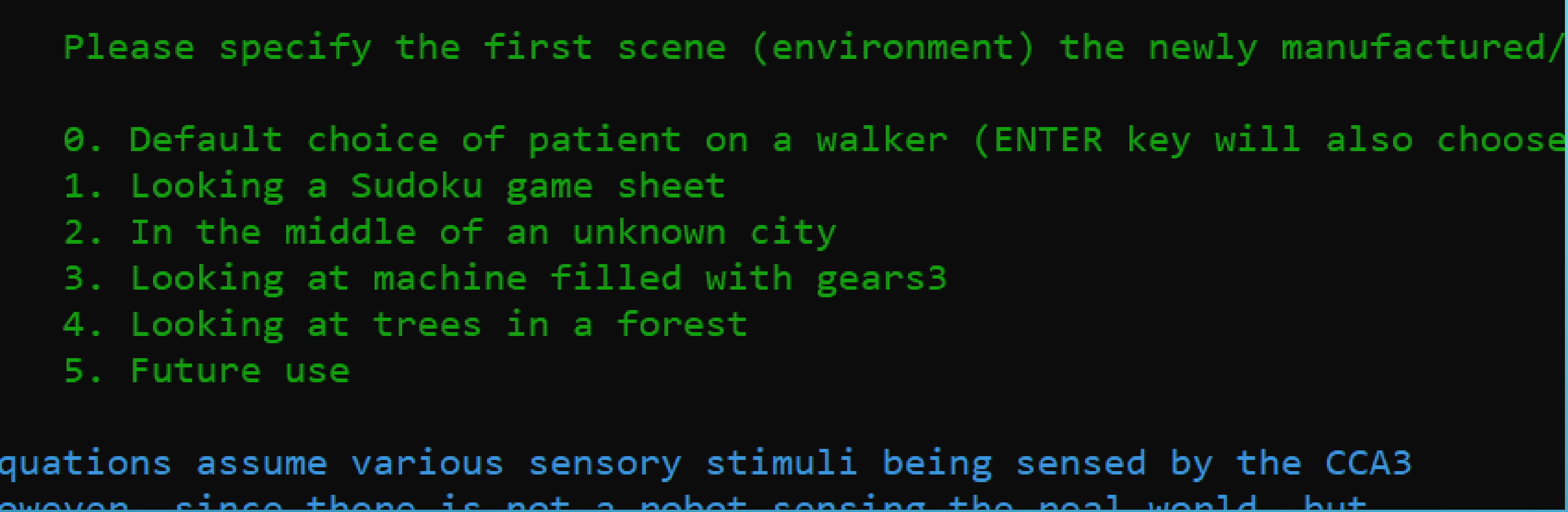
```
Please choose type of "hippocampus"/"brain" which, of course, only loosely
approximates the biological equivalent (you are effectively setting hyperparamet
0. SAME AS LAST ENVIRONMENT, DO NOT ERASE/REFURBISH THE MEMORY
1. Lamprey-like brain analogue
2. Fish-like brain
3. Reptile-like brain
4. Mammalian-like brain - note: meaningfulness, precausal
5. Human-like brain - note: meaningfulness plus full causal features
6. Augmented Human level 1 - simultaneous application of multiple primitives
7. Augmented Human level 2 - enhanced generative abilities
```

full, normal CCA3 architecture simulated

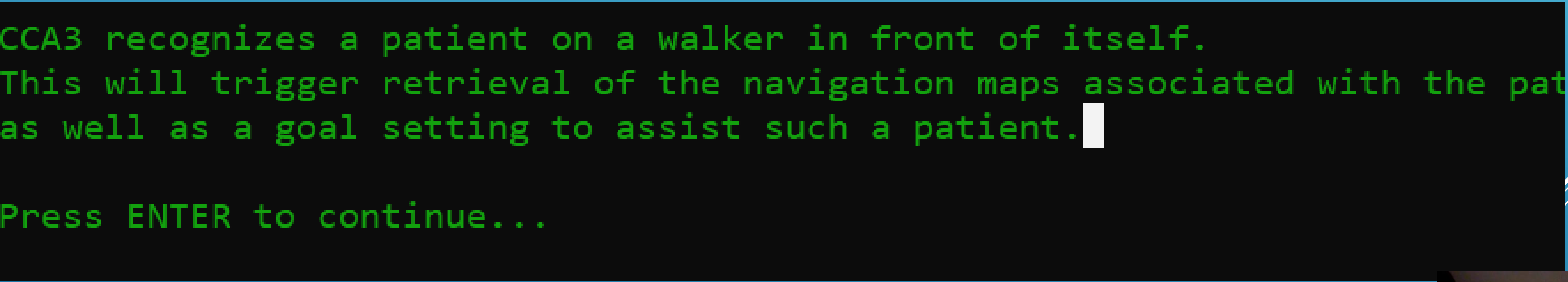




```
CHOOSE ENVIRONMENT FIRST SCENE IS TO START IN

The first scene the newly manufactured/refurbished robot sees and
maps and instincitve primitives related to the scene's environment
For the remainder of the environment (i.e., until success or fail
embodiment, ie, the 'robot' will be in an environment where the sc
For example, in the PATIENT environment simulation, the first scen
a walker in a hospital room. The next scene might be the patient a
the scenes are in the hospital room with the patient. When the sce
i.e., the simulation in the hospital room (environment PATIENT) is
choose another first scene/environment to run the CCA3 robot in. F
CCA3 plays a game of Sudoku, or perhaps you want to go back to the
over again.
```

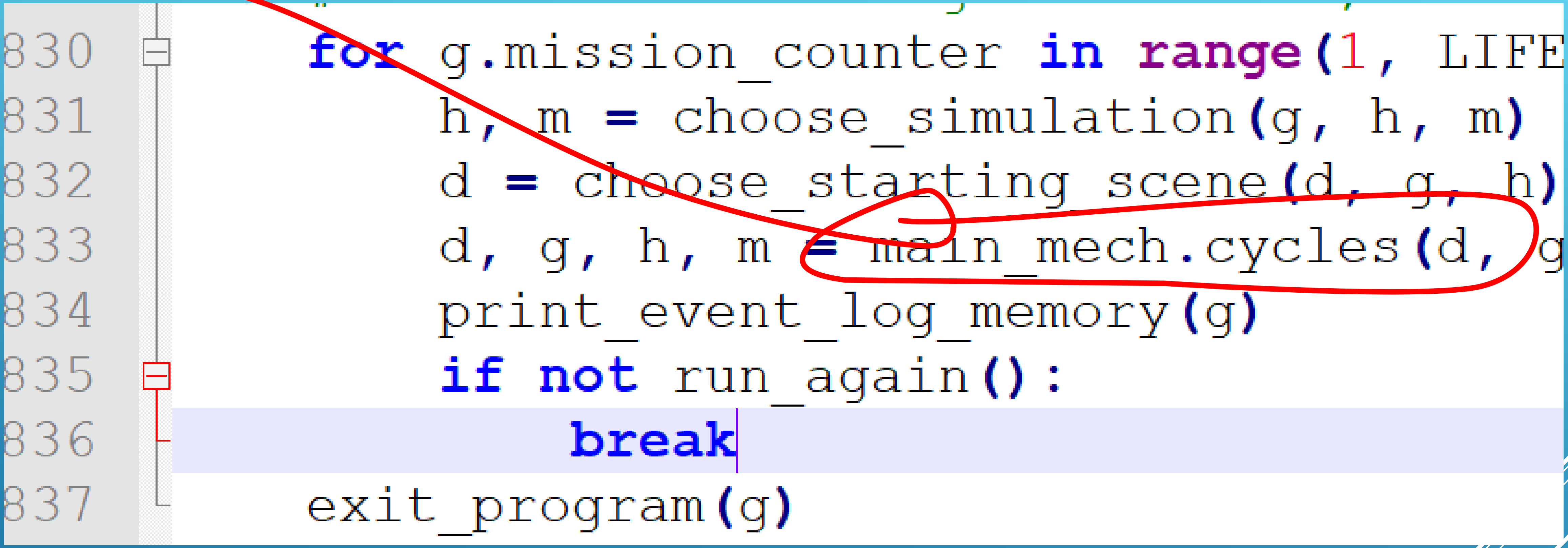


```
   Please specify the first scene (environment) the newly manufactured/

   0. Default choice of patient on a walker (ENTER key will also choose
   1. Looking a Sudoku game sheet
   2. In the middle of an unknown city
   3. Looking at machine filled with gears3
   4. Looking at trees in a forest
   5. Future use

quations assume various sensory stimuli being sensed by the CCA3
owever, since there is not a robot sensing the real world, but
```



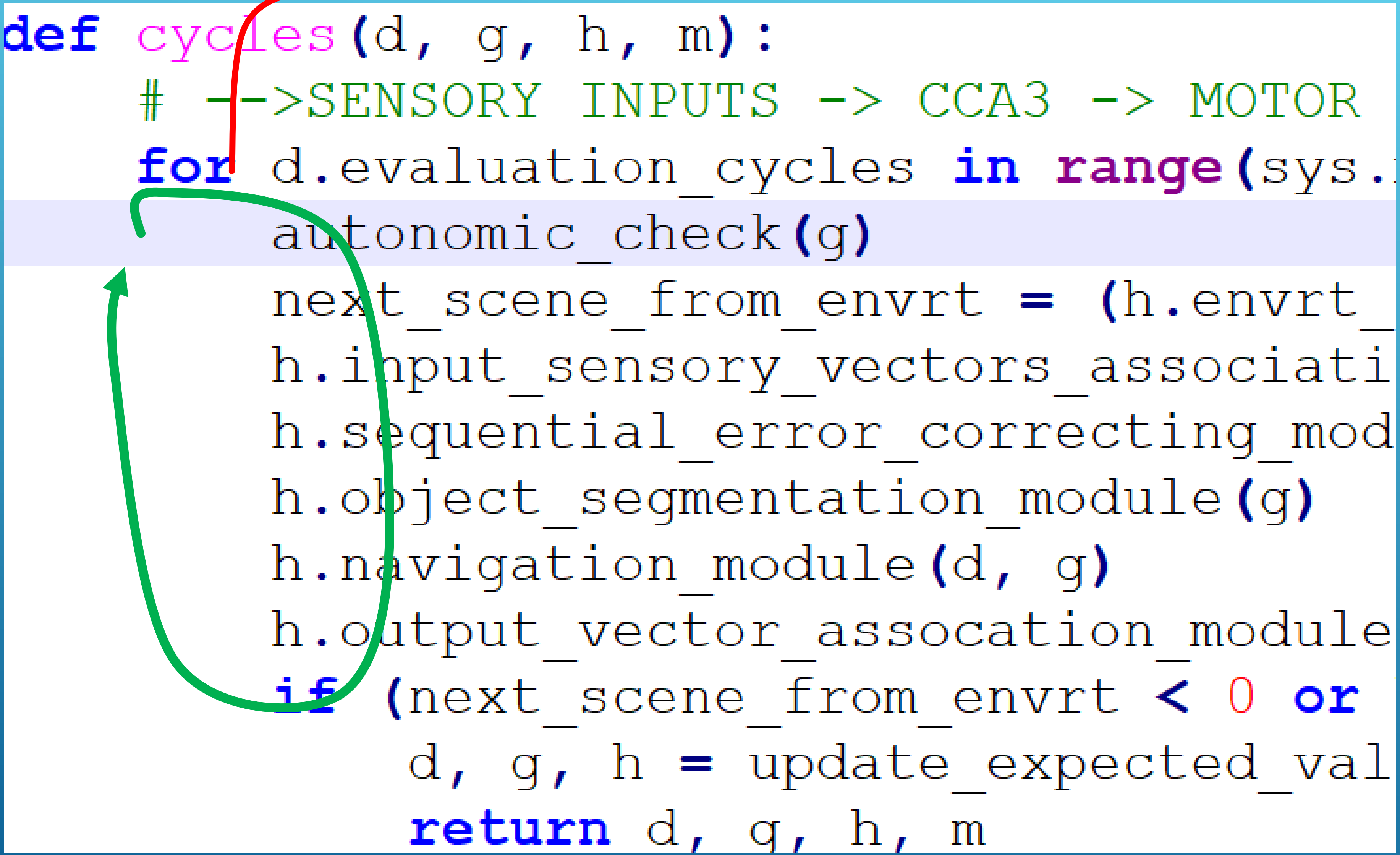
```
CCA3 recognizes a patient on a walker in front of itself.
This will trigger retrieval of the navigation maps associated with the pat
as well as a goal setting to assist such a patient.

Press ENTER to continue...
```

“main()” of CCA3

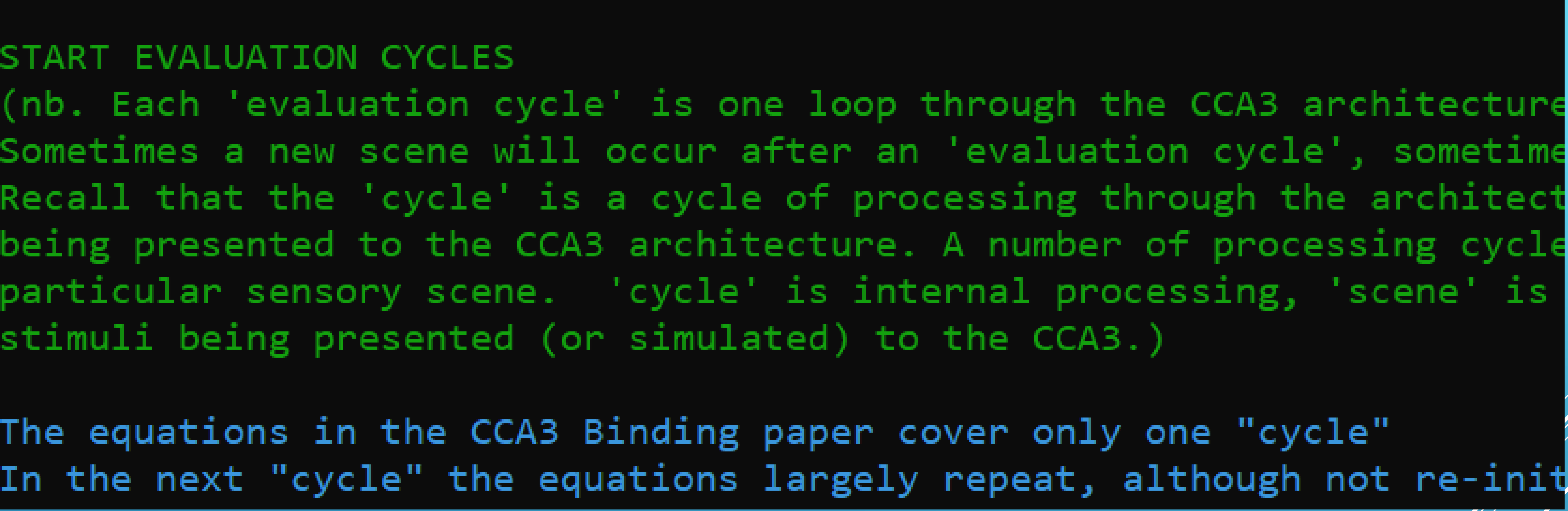


```
    for g.mission_counter in range(1, LIFE
        h, m = choose_simulation(g, h, m)
        d = choose_starting_scene(d, g, h)
        d, g, h, m = main_mech.cycles(d, g
        print_event_log_memory(g)
        if not run_again():
            break
    exit_program(g)
```

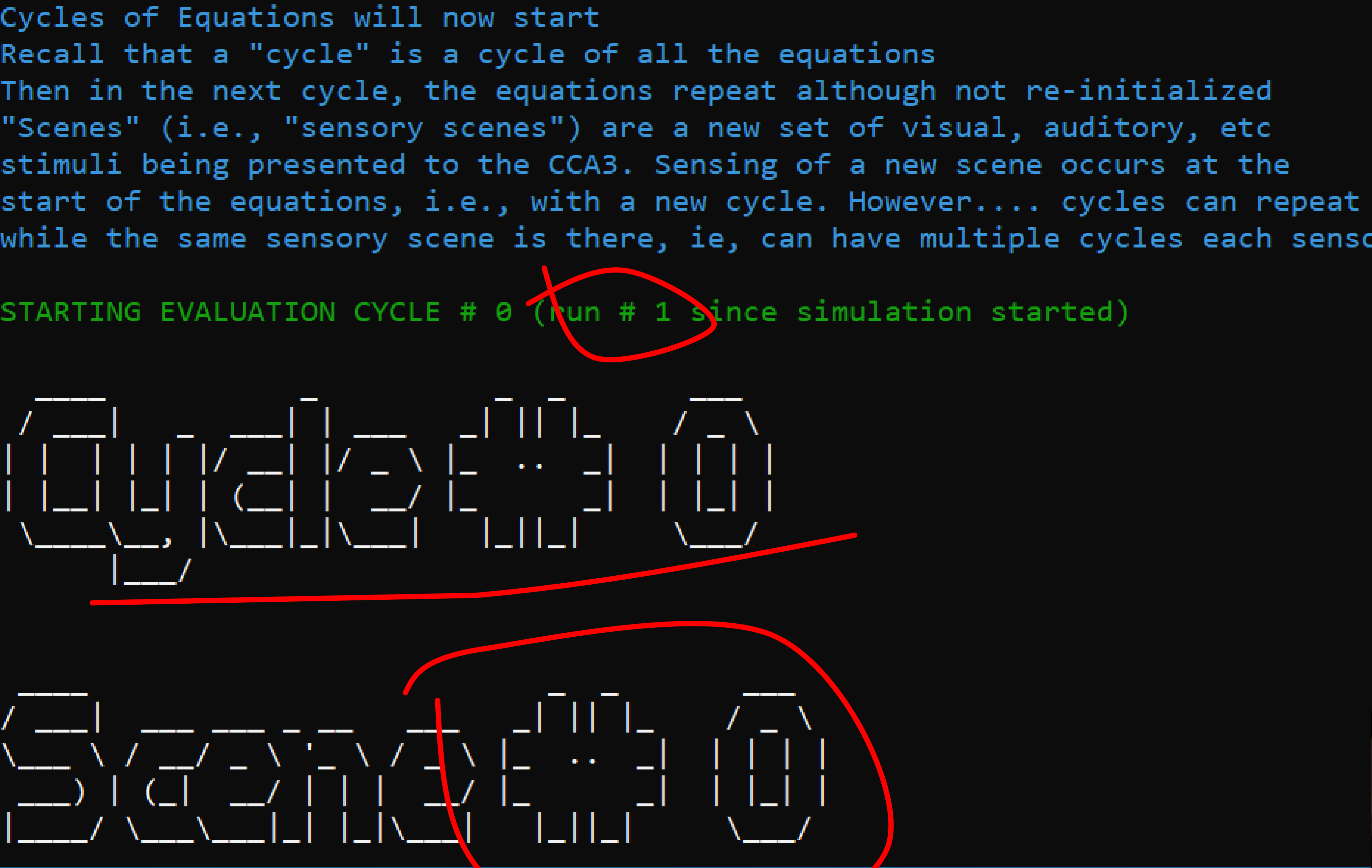
“cycle” == “processing cycle”



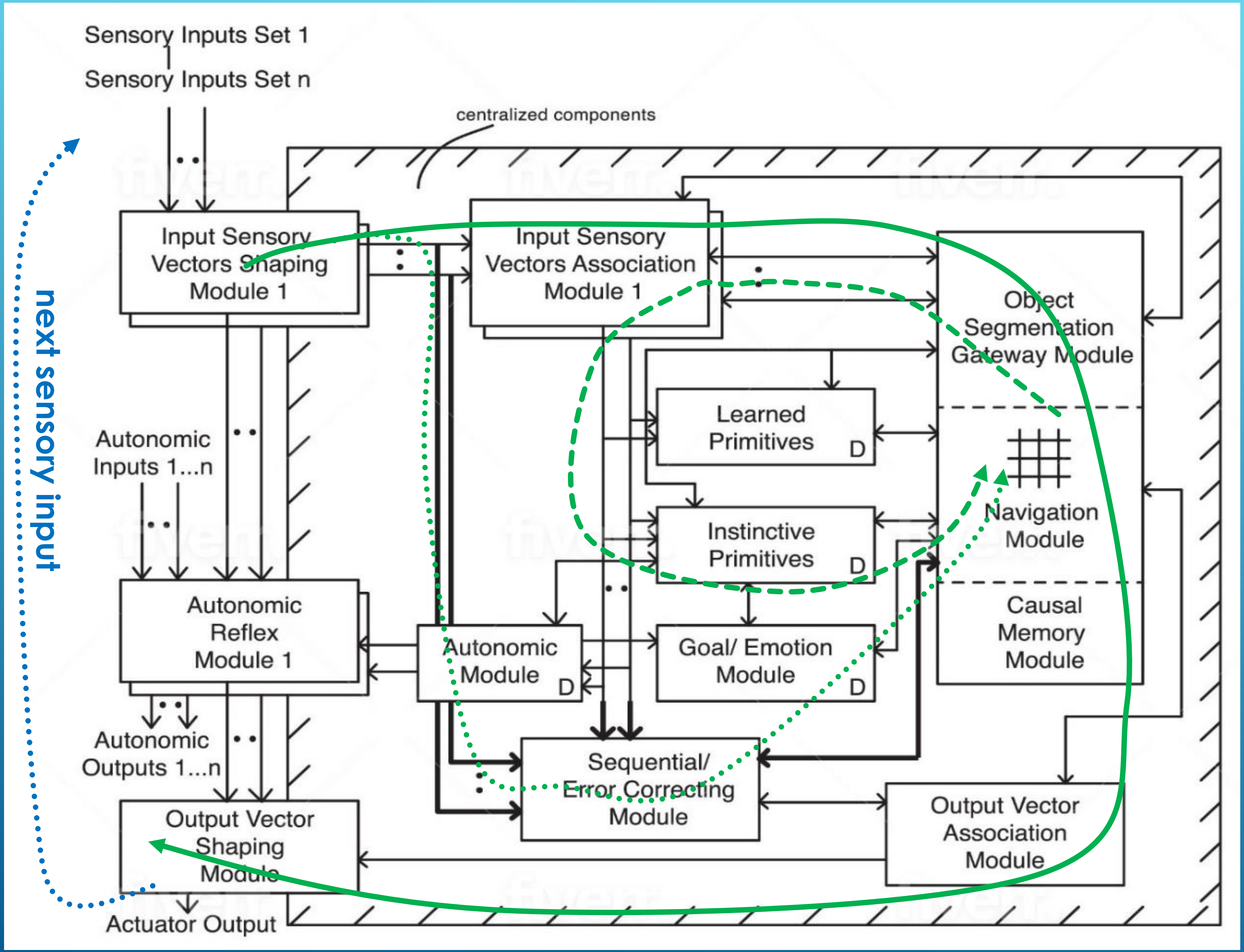
```
def cycles(d, g, h, m):
    # -->SENSORY INPUTS -> CCA3 -> MOTOR
    for d.evaluation_cycles in range(sys.
        autonomic_check(g)
        next_scene_from_envrt = (h.envrt_
        h.input_sensory_vectors_associati
        h.sequential_error_correcting_mod
        h.object_segmentation_module(g)
        h.navigation_module(d, g)
        h.output_vector_assocation_module
        if (next_scene_from_envrt < 0 or
            d, g, h = update_expected_val
            return d, g, h, m
```



```
START EVALUATION CYCLES
(nb. Each 'evaluation cycle' is one loop through the CCA3 architecture
Sometimes a new scene will occur after an 'evaluation cycle', sometime
Recall that the 'cycle' is a cycle of processing through the architect
being presented to the CCA3 architecture. A number of processing cycle
particular sensory scene.  'cycle' is internal processing, 'scene' is
stimuli being presented (or simulated) to the CCA3.)

The equations in the CCA3 Binding paper cover only one "cycle"
In the next "cycle" the equations largely repeat, although not re-init
```

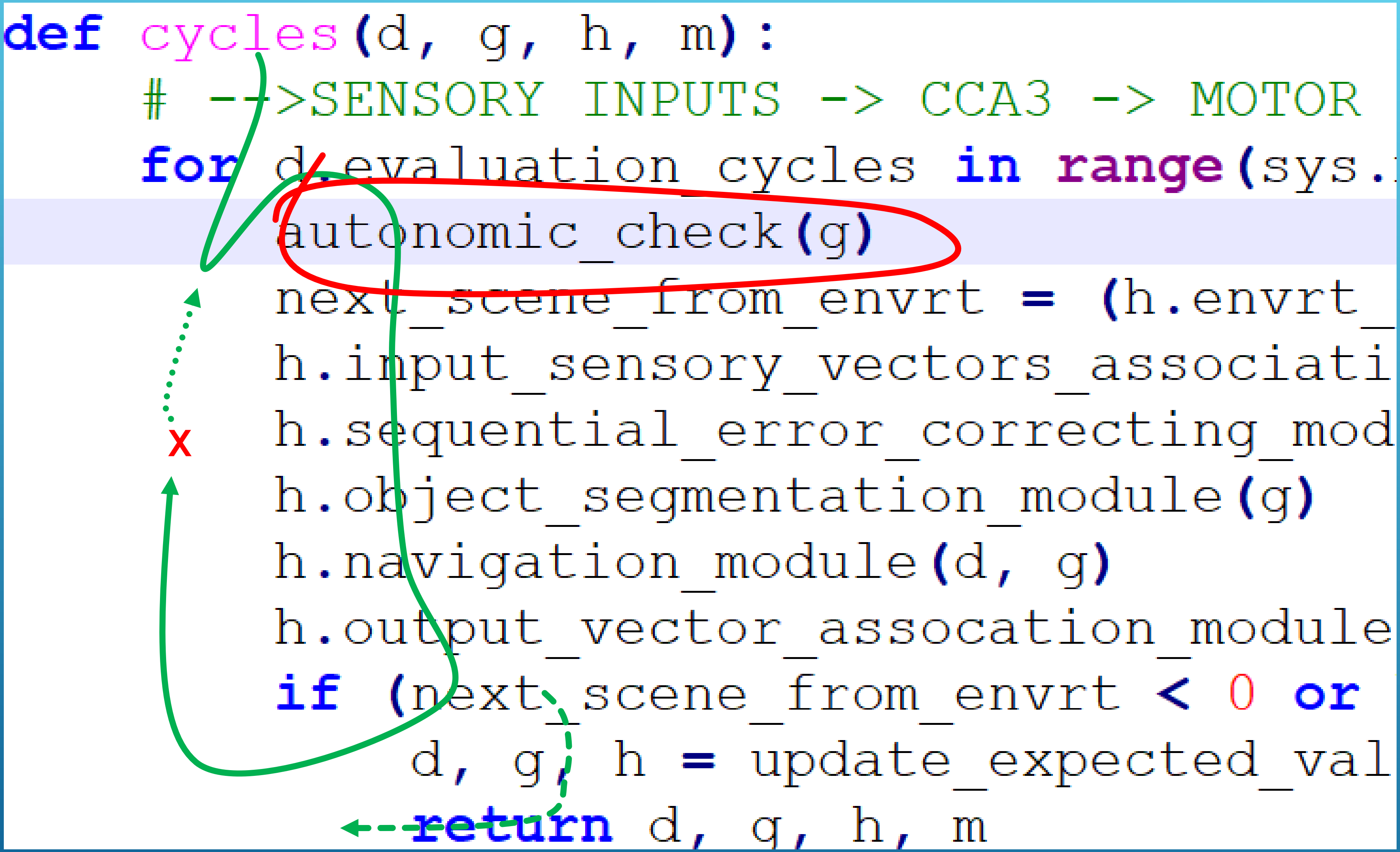


```
Cycles of Equations will now start
Recall that a "cycle" is a cycle of all the equations
Then in the next cycle, the equations repeat although not re-initialized
"Scenes" (i.e., "sensory scenes") are a new set of visual, auditory, etc
stimuli being presented to the CCA3. Sensing of a new scene occurs at the
start of the equations, i.e., with a new cycle. However.... cycles can repeat
while the same sensory scene is there, ie, can have multiple cycles each sensory scene

STARTING EVALUATION CYCLE # 0 (run # 1 since simulation started)

  ____           _        _  _      ___
 / ___|   _  ___| | ___ _| || |_   / _ \
| |  | | | |/ __| |/ _ \_  ..  _| | | | |
| |__| |_| | (__| |  __/_      _| | |_| |
 \____\__, |\___|_|\___| |_||_|    \___/
      |___/


 ____                      _  _      ___
/ ___|  ___ ___ _ __   ___| || |_   / _ \
\___ \ / __/ _ \ '_ \ / _ \_  ..  _| | | | |
 ___) | (_|  __/ | | |  __/_      _| | |_| |
|____/ \___\___|_| |_|\___| |_||_|    \___/
```

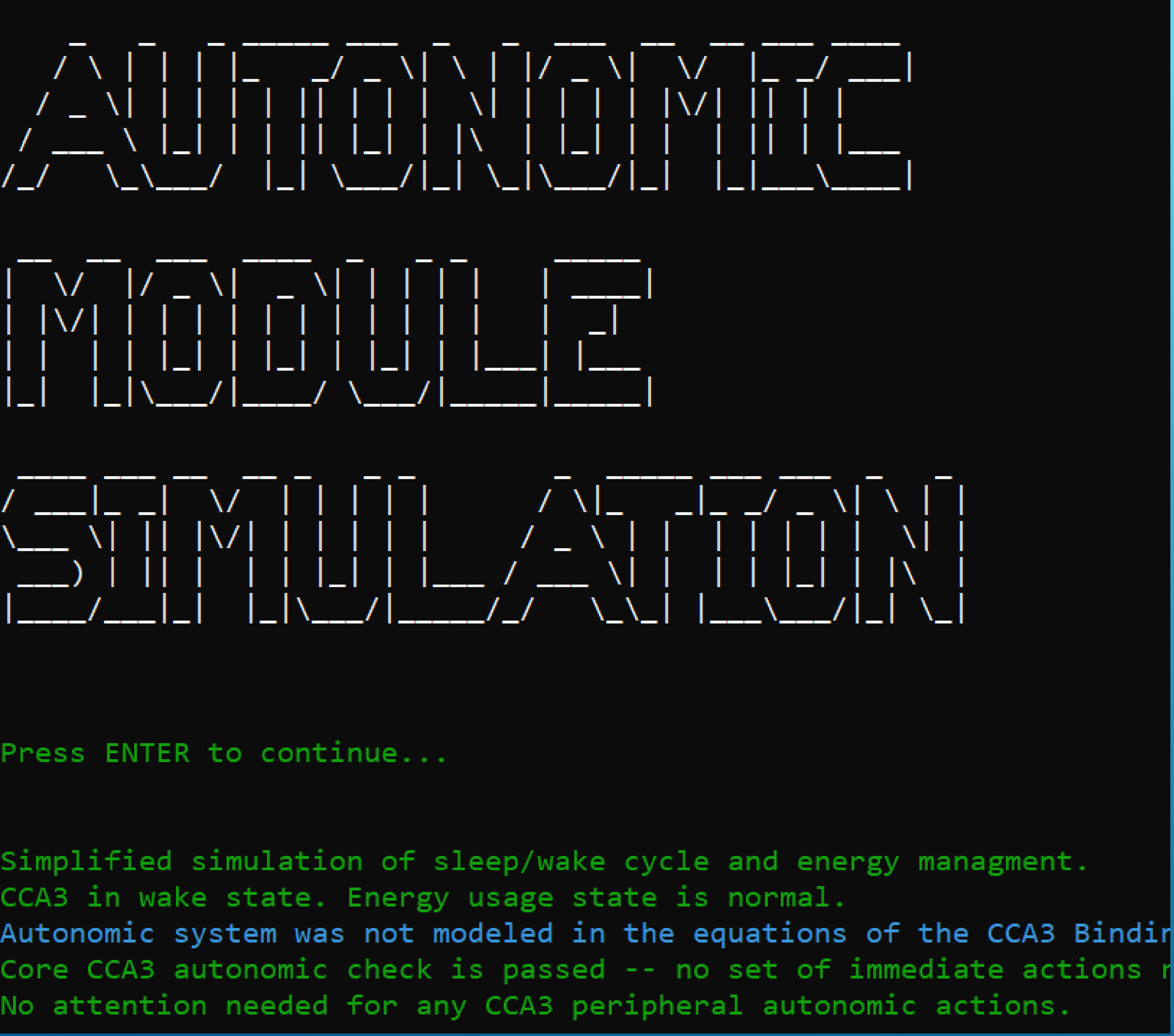


“cycle” == “processing cycle”

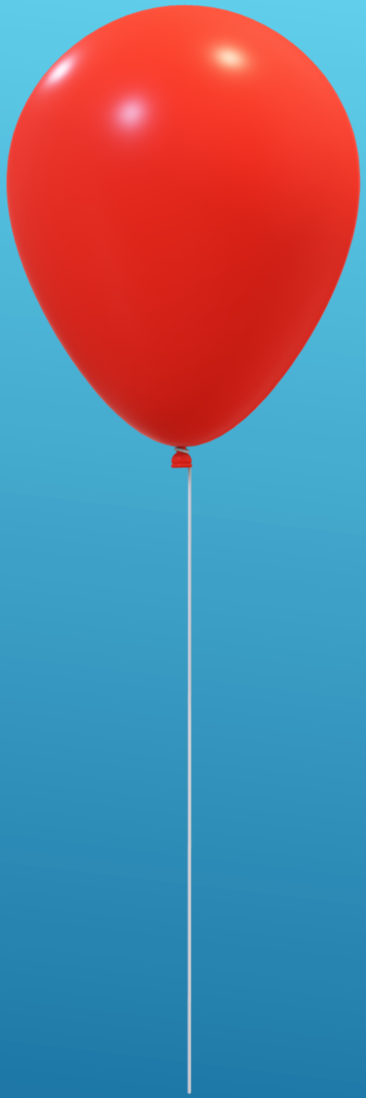
main_mech.cycles()



```
def cycles(d, g, h, m):
    # -->SENSORY INPUTS -> CCA3 -> MOTOR
    for d.evaluation_cycles in range(sys.
        autonomic_check(g)
        next_scene_from_envrt = (h.envrt_
        h.input_sensory_vectors_associati
        h.sequential_error_correcting_mod
        h.object_segmentation_module(g)
        h.navigation_module(d, g)
        h.output_vector_assocation_module
        if (next_scene_from_envrt < 0 or
            d, g, h = update_expected_val
            return d, g, h, m
```



```
    _   _   _ _____ ___  _   _  ___  __  __ ___ ____ 
   / \ | | | |_   _/ _ \| \ | |/ _ \|  \/  |_ _/ ___|
  / _ \| | | | | || | | |  \| | | | | |\/| || | |    
 / ___ \ |_| | | || |_| | |\  | |_| | |  | || | |___ 
/_/   \_\___/  |_| \___/|_| \_|\___/|_|  |_|___\____|

 __  __  ___  ____  _   _ _     _____ 
|  \/  |/ _ \|  _ \| | | | |   | ____|
| |\/| | | | | | | | | | | |   |  _|  
| |  | | |_| | |_| | |_| | |___| |___ 
|_|  |_|\___/|____/ \___/|_____|_____|

 ____ ___ __  __ _   _ _        _  _____ ___ ___  _   _ 
/ ___|_ _|  \/  | | | | |      / \|_   _|_ _/ _ \| \ | |
\___ \| || |\/| | | | | |     / _ \ | |  | | | | |  \| |
 ___) | || |  | | |_| | |___ / ___ \| |  | | |_| | |\  |
|____/___|_|  |_|\___/|_____/_/   \_\_| |___\___/|_| \_|


Press ENTER to continue...


Simplified simulation of sleep/wake cycle and energy managment.
CCA3 in wake state. Energy usage state is normal.
Autonomic system was not modeled in the equations of the CCA3 Bindir
Core CCA3 autonomic check is passed -- no set of immediate actions r
No attention needed for any CCA3 peripheral autonomic actions.
```



....continued in VIDEO 4

balloon from powerpoint stock image