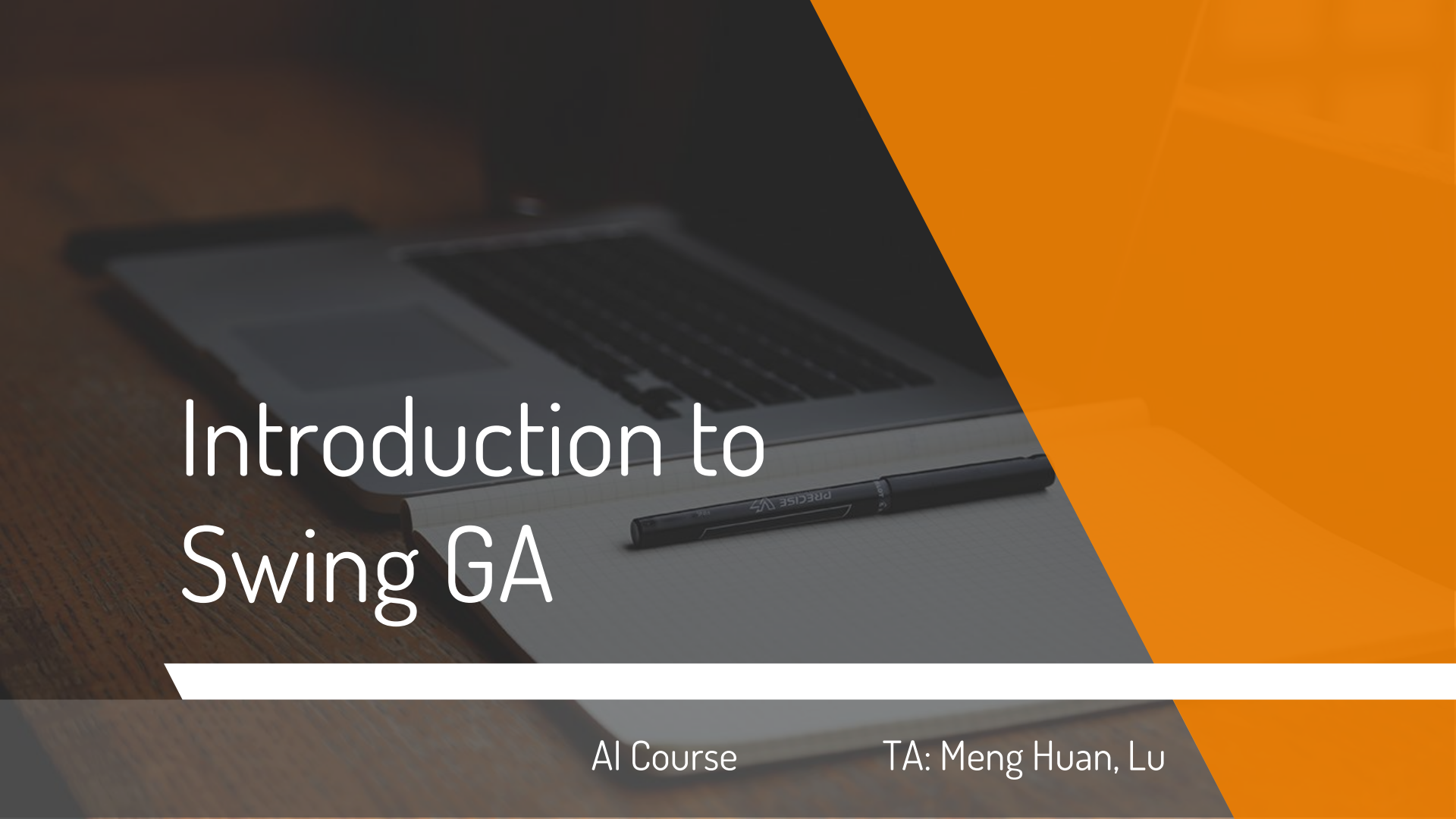


Introduction to Swing GA

AI Course

TA: Meng Huan, Lu

Outline

- **Introduction to V-rep**
- **Framework of swing GA**
- **Code Tracing**
- **Referece**

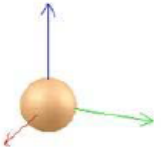
1.
What is V-rep

V-rep

- What is V-rep ?
  - **General purpose robot simulator** with integrated development environment.
- What can it do ?
  - Sensors, mechanisms, robot and whole systems can be modelled and simulated in various ways.
- V-rep is composed by three central elements:
  - Scene Objects
  - Calculation Modules
  - Control Mechanisms

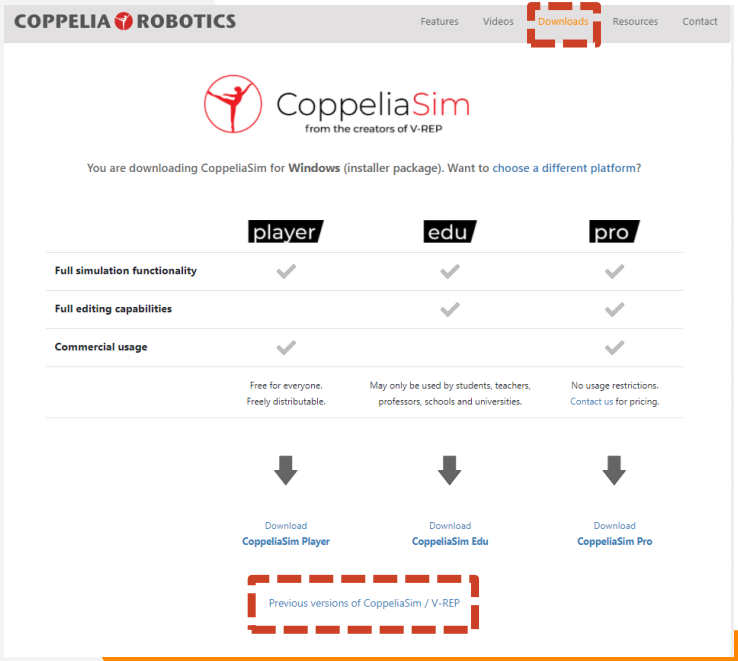
Joints

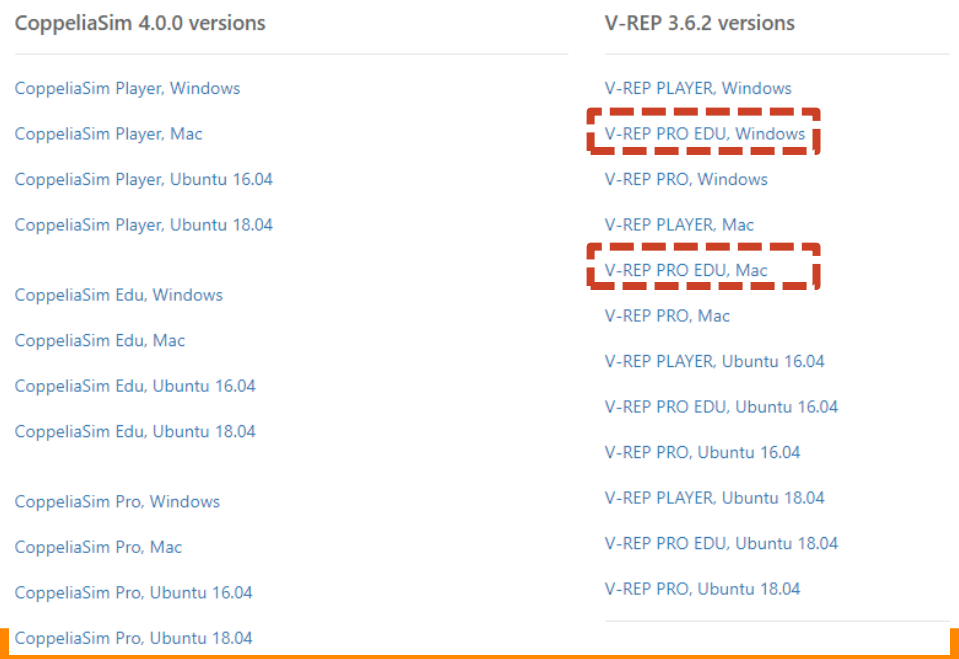
Dummies



Downloads

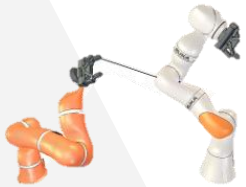
- Version: **V-REP 3.6.2 PRO EDU Windows/Mac**
- Link: https://www.coppeliarobotics.com/downloads



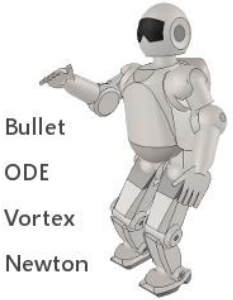


V-rep (Cont.)

Minimum Distance Calculation



Dynamics/Physic



Calculation Model

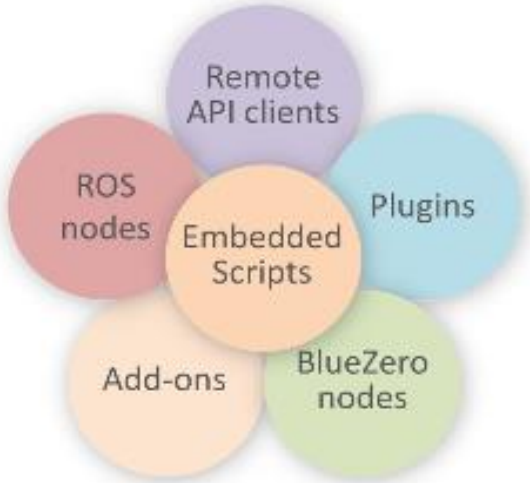
Collision Detect

Path Planing

V-rep (Cont.)

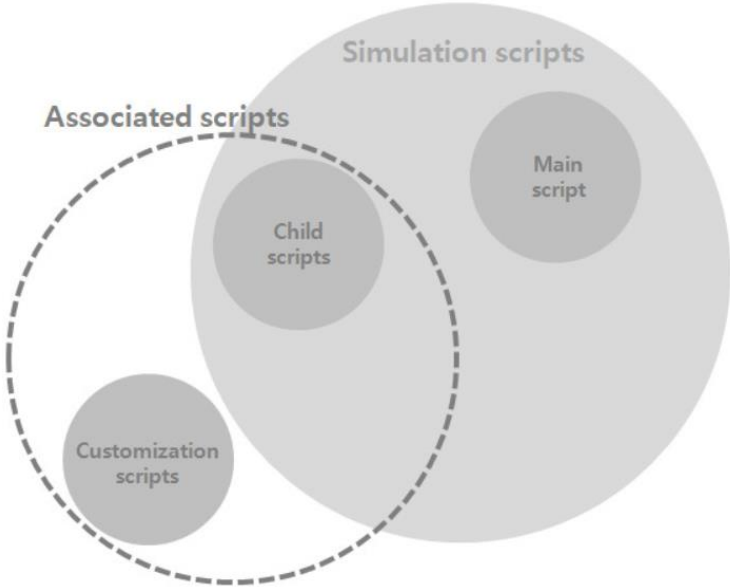
Control Model (embedded scripts)

- Local Interface
- Over 500 API functions
  (https://www.coppeliarobotics.com/helpFiles/en/apiFunctionListAlphabetical.htm)
- Can be attached to an object
- **Lua interface**
  - feature: lightweight、easy to program



V-rep (Cont.)

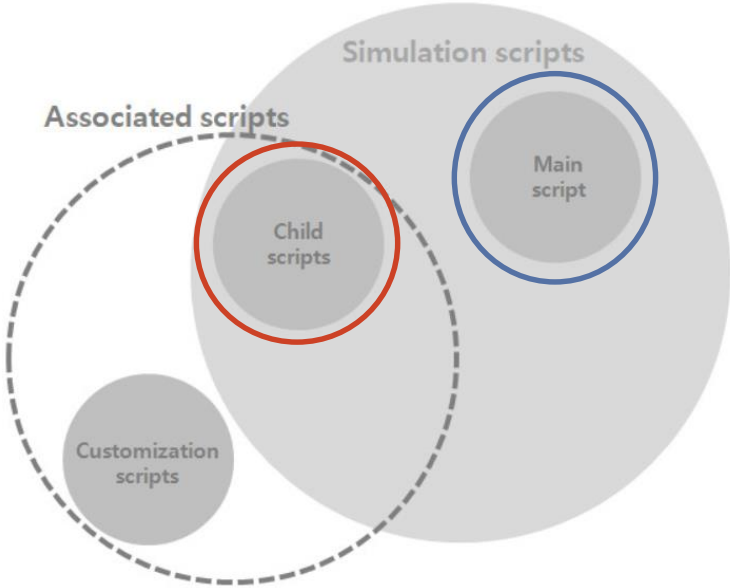
- Simulation scripts:
  - Executed only during simulation.
  - Usually divided into four parts, initiation, actuation, sensing, and restoration.
- Customization scripts:
  - Executed outside. (meaning not controlled by simulation script)
  - **Not used this time.**



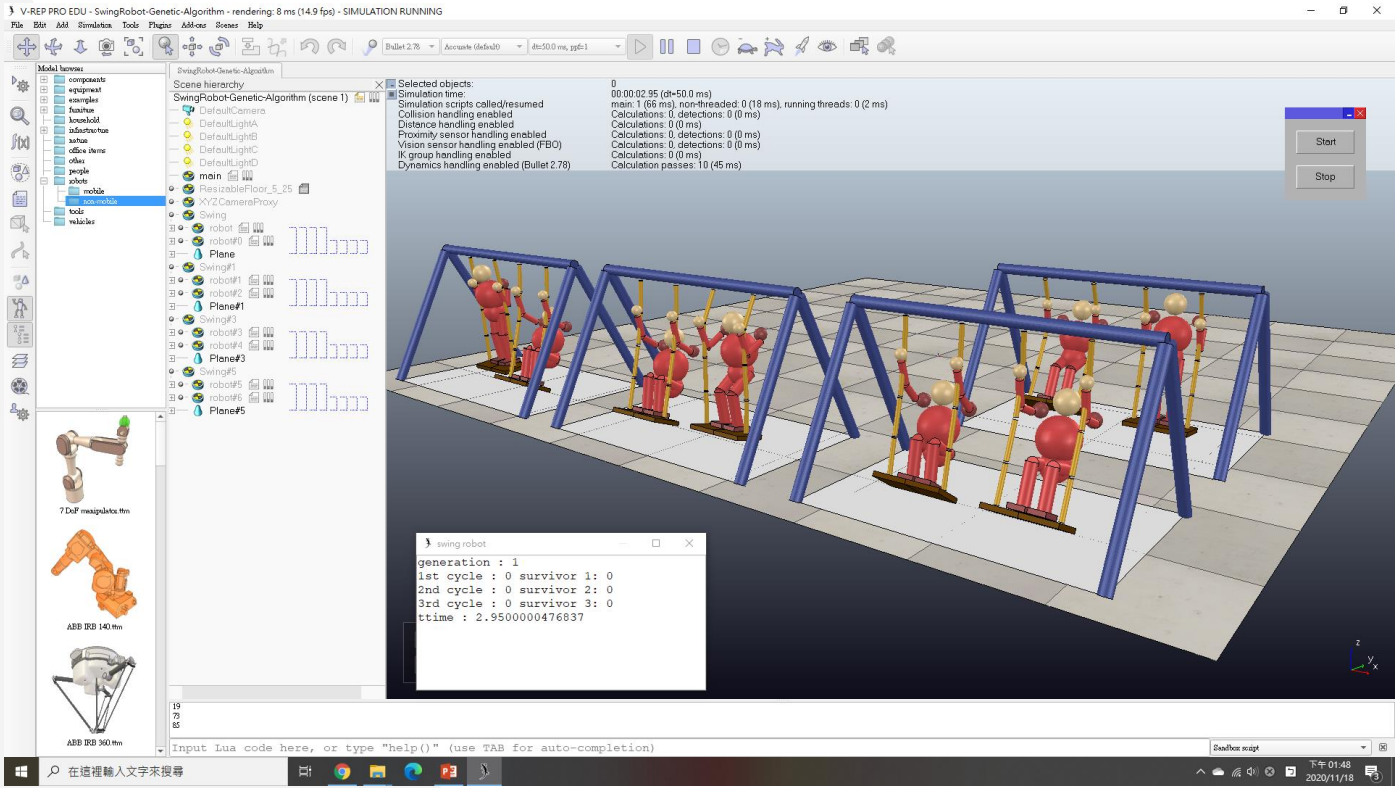
V-rep (Cont.)

- Main script:
  - By default, each scene has own a main script which handles all the functionality.
  - Contains simulation loop code, so simulation can’t run without main script.
  - Not recommended to modify main script.

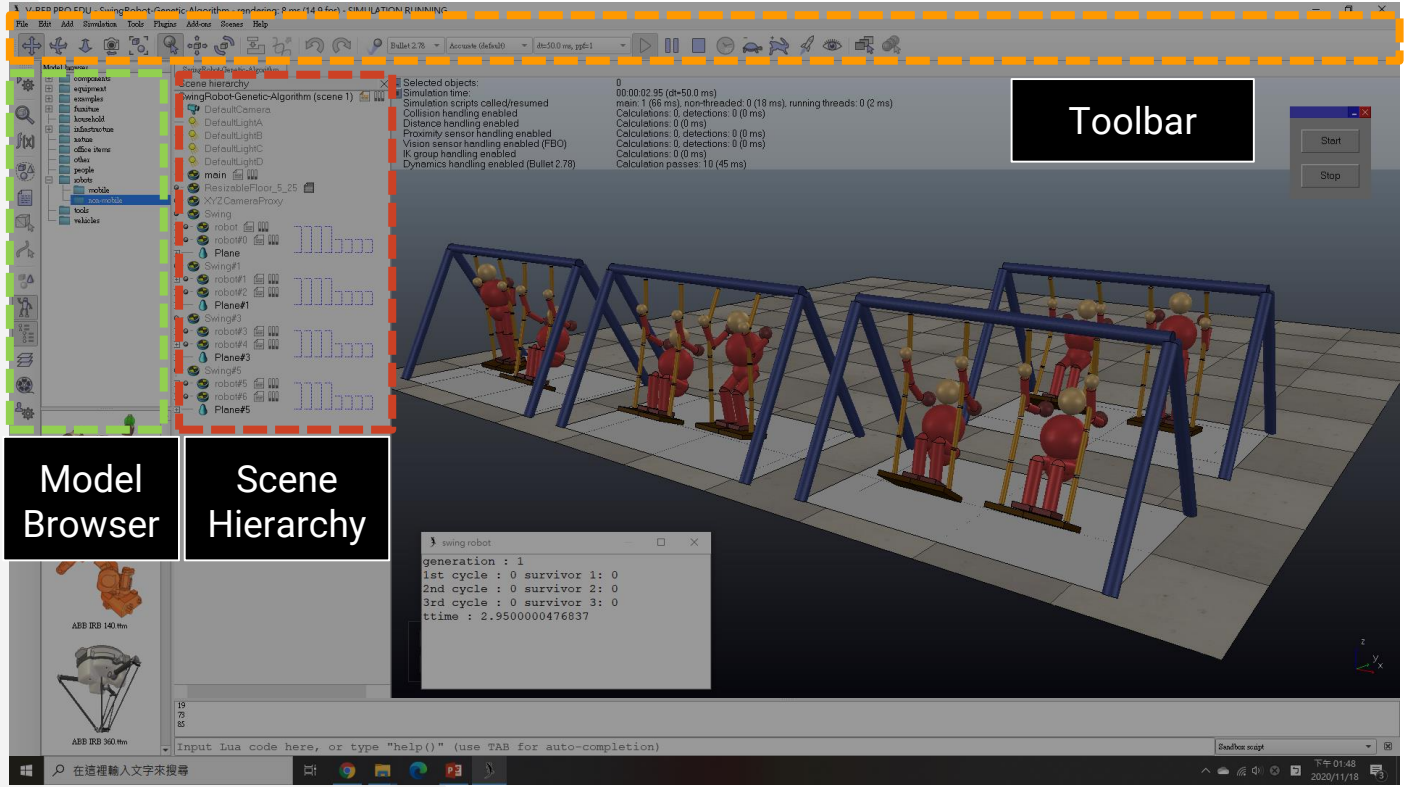
- Child script:
  - Contains typical code of objects or models.
  - Executed in cascaded way, starting with root objects and ending with leaf objects.



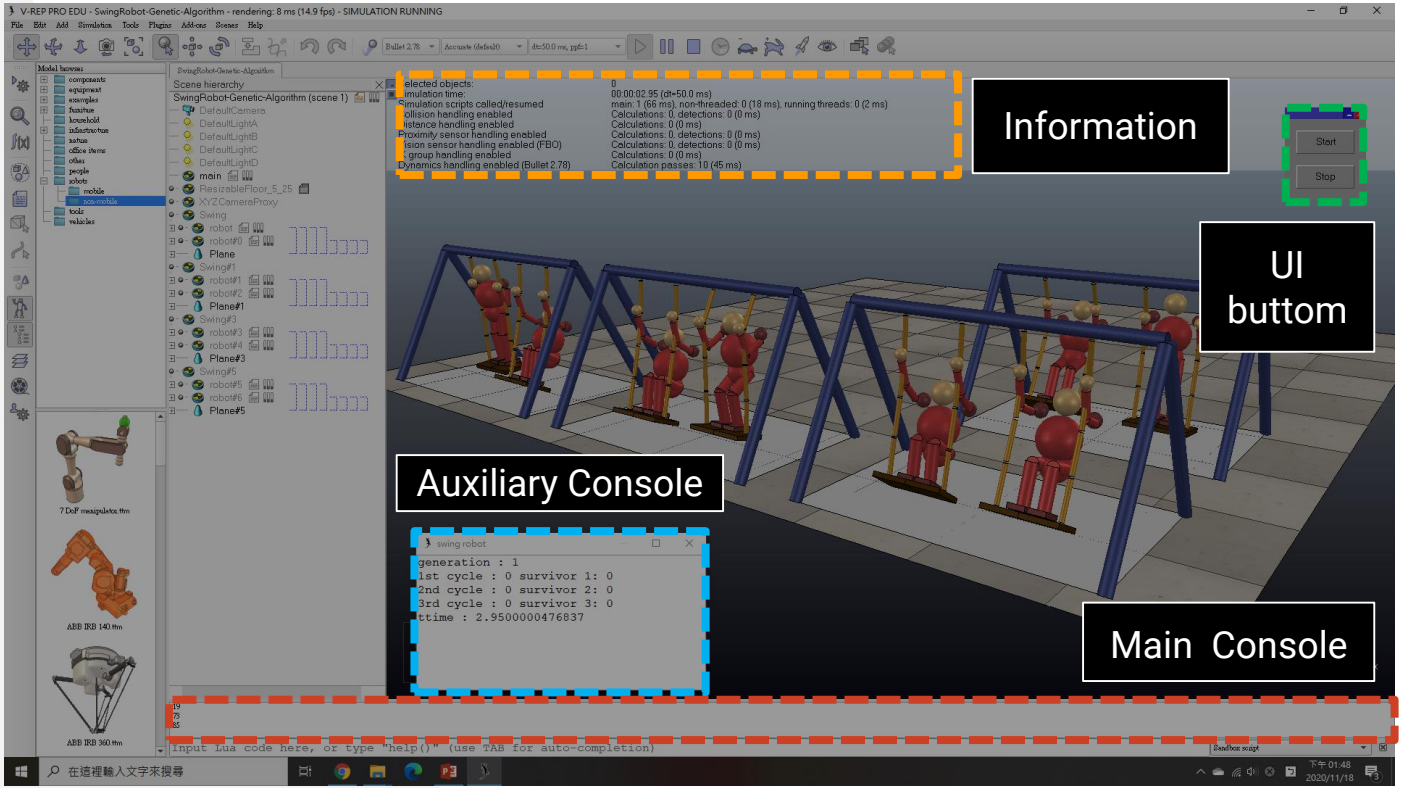
User Interface



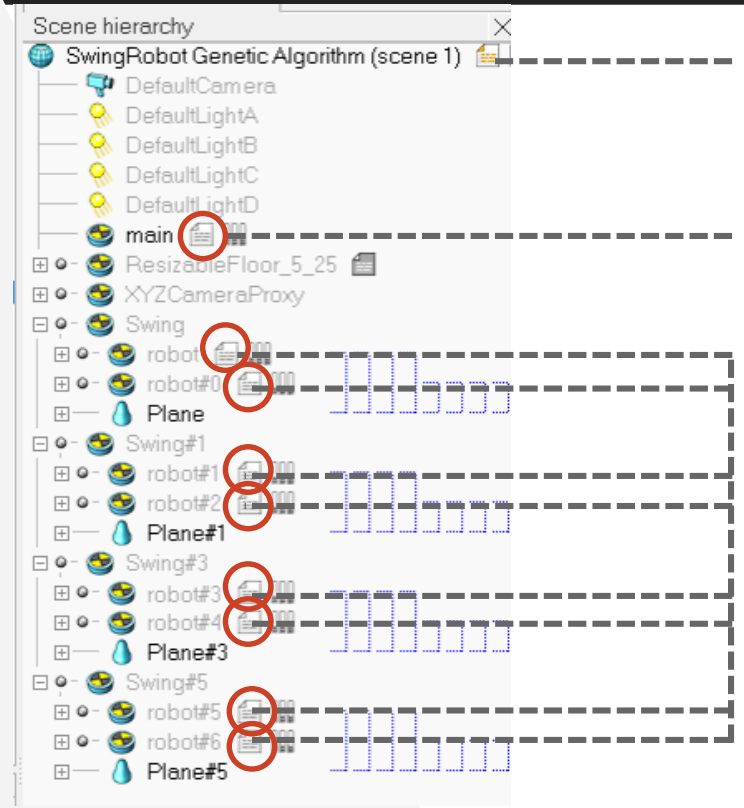
User Interface



User Interface



Scene Hierarchy



Main Script (not use)

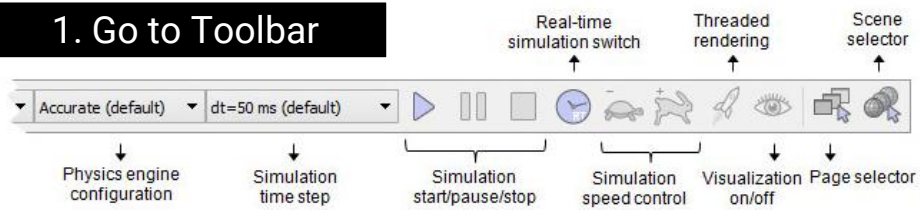
main Child Script, implement your GA here

robot Child Script, not necessary to modify

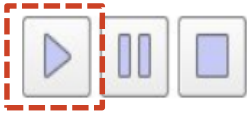
Simulation

How to start simulation of Swing GA:

1. Go to Toolbar

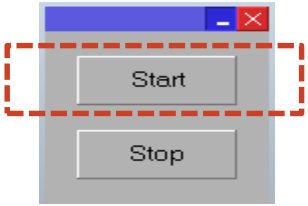


2. Press simulation start



[Simulation start/pause/stop toolbar buttons]

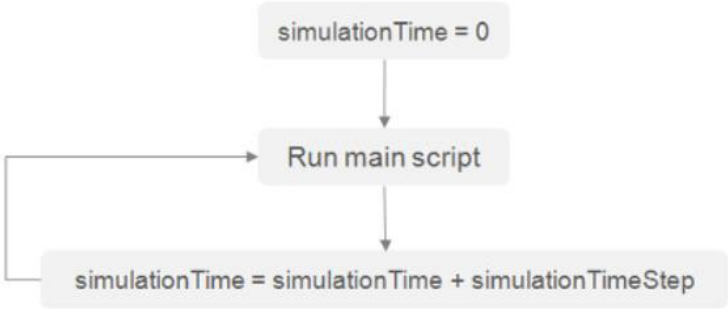
3. Press UI start



Simulation loop:

Each simulation cycle is composed by following sequential operations:

1. Executing the main script
2. Rendering the scene



[Main simulation loop]

Simulation(Cont.)

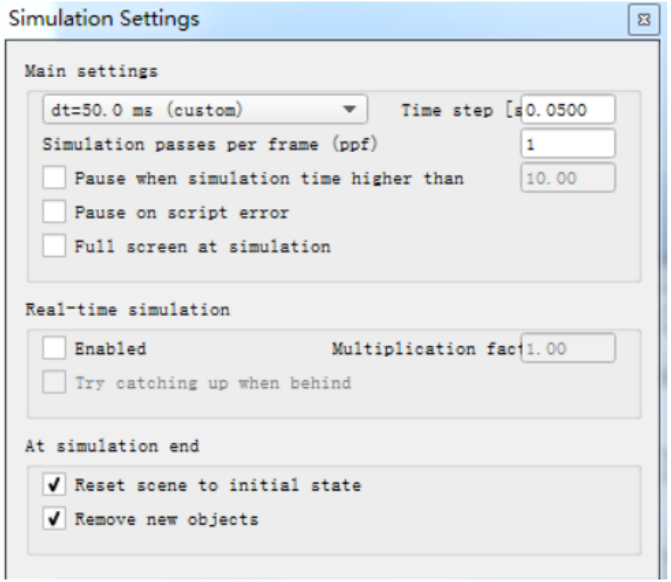
Time step:

Each time the main script was executed, the simulation time is incremented by the simulation time step.

It is highly recommended to keep a default time step.

Simulation passes per frame (ppf):

The number of simulation passes for one rendering pass.



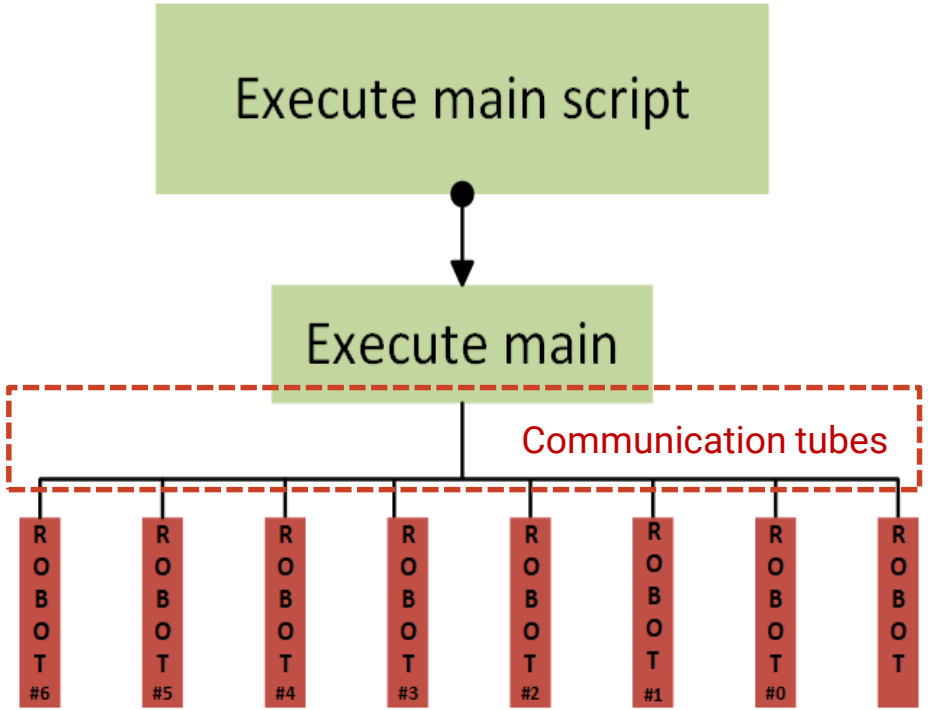
[Simulation settings dialog]

2.
Details of swing GA

Structure of Swing GA

main (child script)

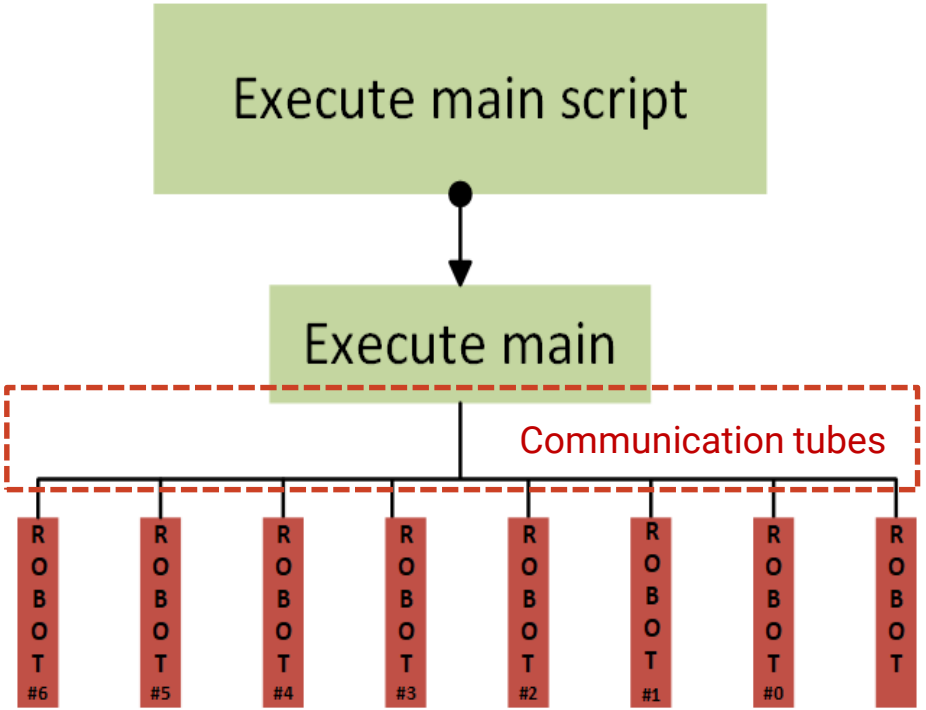
- Initiation stage
  - parameter declaration
- Actuation stage
  - GA implementation
    - ***init_gene* function** (produce chromosomes)
  - Survivor selection
    - record best fitness individuals from robots



Structure of Swing GA

robot (child script owned by every robot)

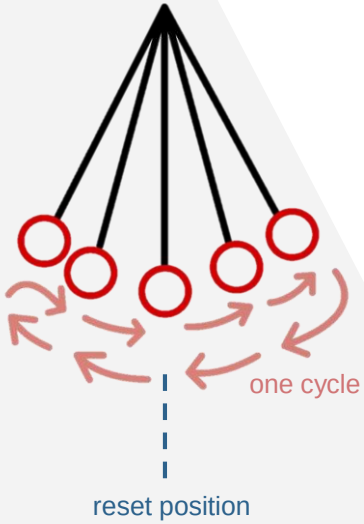
- Initiation stage
  - initiate robots
  - parameter declarations
- Actuation stage
  - pop action
  - fitness evaluation
- Finish
  - wait for next generation

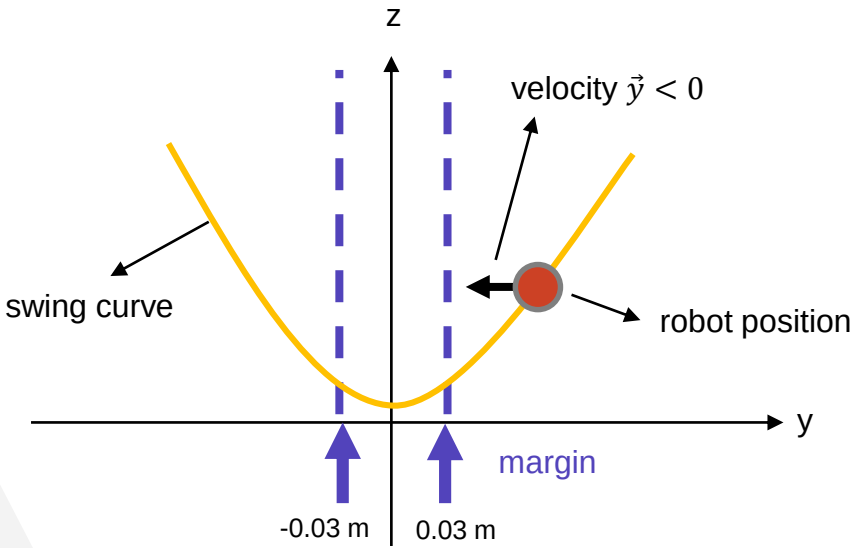


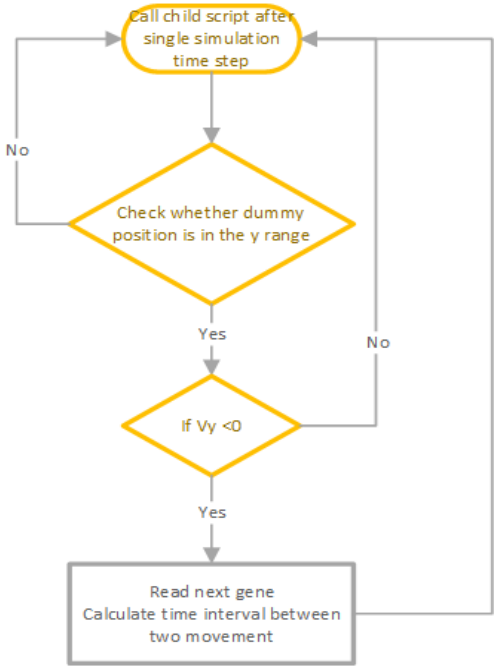
GA Flowchart and Parameter Setting of Sample Code

1. Population size: 8 (one robot means one individual, you can add by yourself)
2. Chromosome length: 100 bits (“1” represents “Up” and “0” represents “Down”)
3. GA operators:
   - **crossover**: multiple crossover
   - **mutation**: combined with crossover
   - **survivor selection**: maintain top three individuals as parents

Evaluation Mechanism

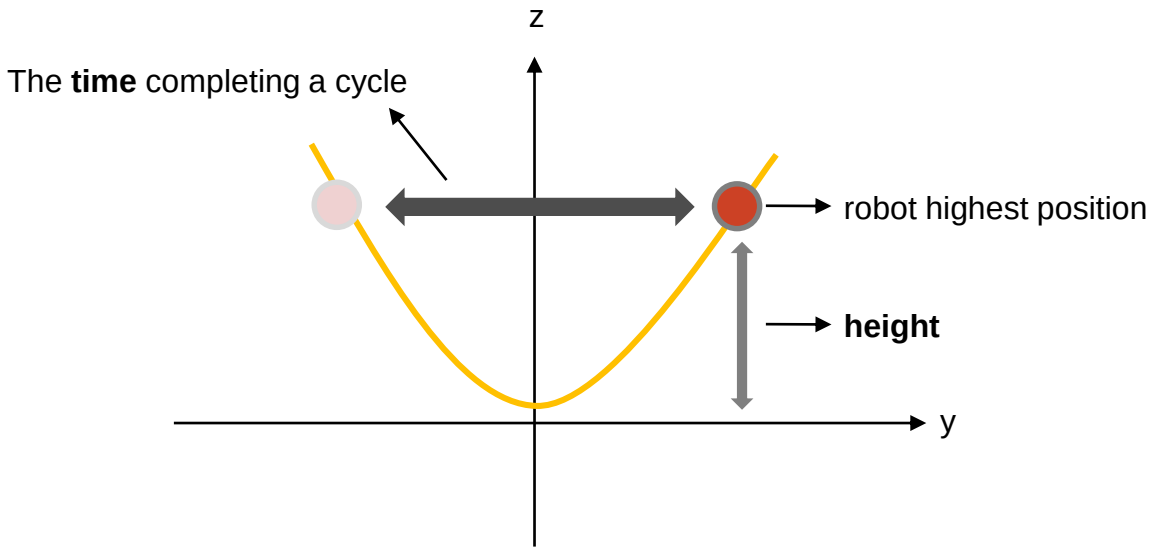






- A "**cycle**" means one complete back-and-forth motion.
- If robot pass the margin with velocity $\vec{y} < 0$, then evaluate the fitness.

Evaluation Mechanism



- The fitness is evaluated by multiplying the **time** completing a cycle, the **height** (the highest position robot reach in a single cycle**)** and a scaler, which means the robot will get higher fitness only when robot swing longer and higher.

3.
Code Tracing

Global Varible

- In Swing GA, all the variables are set to be **global variables** by default, so beware some mistakes in your coding.
- Lua is a **dynamically typed language**, so the variables don't have types, only the values have types. The right table shows the list of value types in Lua.

Sr.No	Value Type & Description
1	**nil** Used to differentiate the value from having some data or no(nil) data.
2	**boolean** Includes true and false as values. Generally used for condition checking.
3	**number** Represents real(double precision floating point) numbers.
4	**string** Represents array of characters.
5	**function** Represents a method that is written in C or Lua.
6	**userdata** Represents arbitrary C data.
7	**thread** Represents independent threads of execution and it is used to implement coroutines.
8	**table** Represent ordinary arrays, symbol tables, sets, records, graphs, trees, etc., and implements associative arrays. It can hold any value (except nil).

Resource: https://www.tutorialspoint.com/lua/lua_data_types.htm

Comments

- There are some **comments** written in source code, which may help you to understand how the function works. (Welcome to discuss if have any problem)

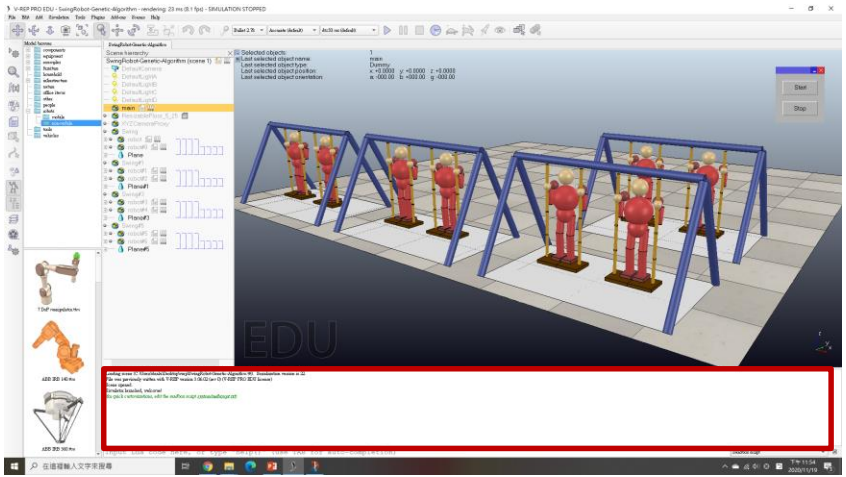
```
]--[[
    All function can be found at https://www.coppeliarobotics.com/helpFiles/en/apiFunctionList

    Some functions need to know:

    sim.tubeOpen(dataHeader, dataName, readBufferSize):
        Tube will only be able to connect if its two sides have the same dataHeader and dataNa
        to simplify the code, dataHeader always keep 1, readBufferSize always keep 0.
        This function will return a tubeHandle and then we can use sim.tubeWrite function to p

    sim.tubeWrite(tubeHandle, data):
        tubeHandle: the handle of the tube that was returned by the sim.tubeOpen function.
        data: the data want to pass, need to convert to string type. (each character can have
```

Console

- By default, we can console the data by using **print** function.

```
print(math.random(0, 100))
```

which would output the data here.

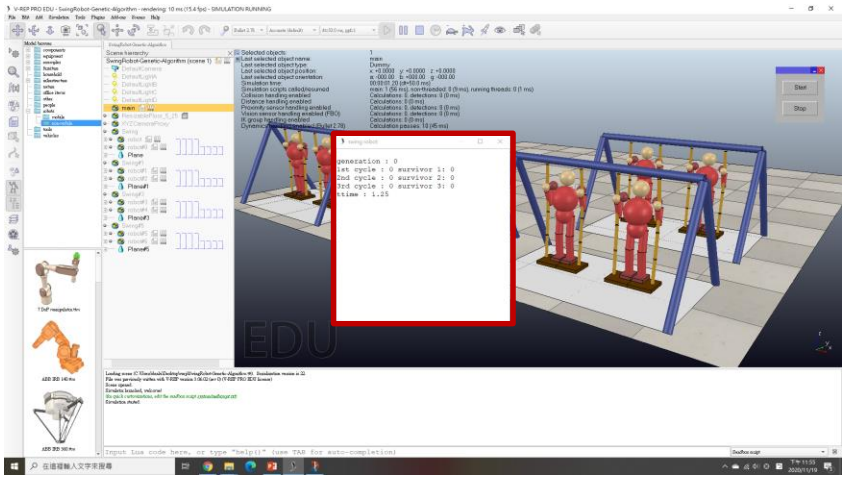


Auxiliary Console

- If we want to identify some variables more clearly, we can use **auxiliary console**.

```
consoleHandler = sim.auxiliaryConsoleOpen("swing robot", 8, 6, {0, 0},
sim.auxiliaryConsolePrint(consoleHandler, "\n")
```

which would output the data here.



Output CSV files

```
PATH = "YourPath"
file = io.open(PATH.."/output.csv", "w")
```

```
-- Use to write CSV file
CSV_write = function(x)
    file:write(x)
    file:write(',')
end

CSV_nextLine = function()
    file:write('\n')
end
```

- We have implement the writing .csv file function so feel free to use it. Just mention it is recommended to assign the absolute path for the .csv file, otherwise the default path will be in C:\Program Files\V-REP3\V-REP_PRO_EDU. (in Windows 10 OS)

V-REP font size

- The default font size in V-REP program may be too small for us.
- We can change the font size by modify the **usrset.txt** file in C:\Program Files\V-REP3\V-REP_PRO_EDU\system. (in Windows 10 OS)
- For Windows users, please modify the **guiFontSize_Win** value.
- For Mac users, please modify the **guiFontSize_Mac** value.

Some Suggestions

1. You should **design genetic algorithm yourself** in *init_gene* function. We'll check if you have revised it. (It is allowed to change or define some parameters yourself)
2. The *init_gene* function must **output a chromosome** to make robot do the action.
3. Do the **experiments** according to homework documents. (compare the parameter setting, plot anytime behavior, etc.)
4. Strongly recommended to **trace the source code**.
5. If you have any question or want to discuss some problems, feel free to contact TA. (come to lab 406 or send email)

4.
Reference

Reference Document

1. V-rep user manual
   Help->about topic

2. Lua tutorial
   https://www.lua.org/pil/contents.html
   https://www.tutorialspoint.com/lua/index.htm
   https://michaelchen.tech/lua-programming/

3. First author
   https://github.com/HaJaeKwon/Vrep-Genetic-Swing-robot

4. V-rep download
   https://www.coppeliarobotics.com/downloads.html

THANKS!