Comp3252 Modelling and Simulation. Worksheet Apocalypse

CBPrice January 2021

Color Coding for this Worksheet	
	General Information
	Information you will need for the activities
	From Math Model to Code
	Guided Activities
	Investigations

Purpose

(a) To understand how to convert a simple mathematical model into computer code. (b) To plan and conduct investigations of a model. (c) To learn how to interpret data and extract information.

Files Required

All files required are installed.

PARAMETERS (default) **for the Apocalypse Model**

gravity	9.8
releaseHeight	54.0
brakingHeight	13.5
brakingForce	39.2
mass	1.0

Console Commands for the Apocalypse Model

The following console commands are available for this simulation. Here "X" refers to a number which you will be asked to input. Remember to include spaces as indicated below.

Console Command	Effect
setParam brakingForce X	The amount of braking force applied to the car
setParam brakingHeight X	The height at which the braking force starts to be applied.
setParam mass X	Sets the mass of the car. Will change according to how many riders are in the car
setParam gravity X	Sets the value of gravity. This is 9.8 for all places on the Earth's surface. On the surface of the Moon it is $1/6^{th}$ of this

Declared Variables and relationship to the Maths symbols

The variables on the left are already declared in the code.

Variables	**Maths**
forceZ	F
accelZ	a
velyZ	v
dispZ	z
Parameters	
mmass	m
gravity	g
brakingForce	F_b
brakingHeight	z_b
velyLift	v_L
Constants	
releaseHeight	z_r
dT	Δt

Activities

1 **Reading the Code Template**

(a) Open up the level **SciencePark_Level.udk** and run the level in the Editor. Find the Apocalypse (you can't miss it, crosshair and left click on the car and run the simulation (F1). Nothing happens! Of course, there is no code! That's your task.

(b) Navigate to the folder **C:\UDK\UDK-2015-01\Development\Src\MAS14\classes** and open up the file MAS14_Apocalypse_FSM.uc in Notepad++ or a text editor of your choice.

(c) Spend a little time looking at the code in this file, in particular

(i) Look for the global variables **dispZ, velyZ, accelZ** towards the start of the file
(ii) Look for the function **SendValuesToHUDX()** to see which variables will be sent to the HUD
(iii) Look for the function **logDataRecord()** which shows you which variables will be written to the log file ready to import into Excel.
(iv) Look for the function **computation(float dT).** The body of this function will be empty apart from the FSM structure. Try to get your head around the FSM, especially understand the exit condition for each stage.

(c) The line **local float forceZ;** in this function is commented out (around line 172). Remove the comment-outs.

2 **Coding the Math Model**

(a) The Dynamics

Using the table provided above, code the following lines of math in the order given. **Put this in the template code file after the Finite State Machine code**, where it says "Now let's compute the DYNAMICS based on the forces set up in each state". To know which statements to use, consult the yellow table above, to go from maths to code.

(a) $a = {}^{F}/_{m}$

(b) $v = v + a\Delta t$

(c) $z = z + v\Delta t$

(b) The forces and velocities for each state.

Now enter code for each state using the variable names in the above table. Remember **forceZ** is a local variable.

(i) STATE_RISING:
* There is no net force since the car is being lifted up at a constant speed. So code this force.
* The speed should be set to **velyLift**

STATE_PAUSEATTOP:
* The force is zero since the car is being held. So code this force.
* The velocity is zero since the car is being held. So code this velocity
* Note that the variable "time" is being updated in the **computeTimer()** function

STATE_FALLING:
* Here the force on the car is provided by gravity $F = mg$. So code this force.
* Note that the variables **velyZ** and **dispZ** are being updated by the "dynamics" code

STATE_BRAKING:
* Here the force is provided by gravity *plus* the braking force applied $F = mg + F_b$.
So code this force.

2 **Interpreting the Apocalypse Data**

Navigate to UDK's logs folder. Mine looks like this **C:\UDK\UDK-2015-01\UDKGame\Logs**
Open up Octave-GUI and in the console point Octave to this folder by typing this
cd C:\UDK\UDK-2015-01\UDKGame\Logs

1. Exit any running UDK level by pressing Esc then restart the level and let the car fall and stop. Hit Esc to exit UDK. In Octave at the command line type **plotApocalypse** to plot the logged data, the filename will be **Apoc_FSM** with an additional run number. Remember to put this in single quotes at Octave command.

(a) Look at the graphs of velyZ and dispZ up to about 5.5 seconds. What is happening to the car during this time? Explain.

(b) From the same two graphs, what is happening after about 5.5 seconds up to about 7.5 seconds? Explain.
(c) Look at the acceleration graph. There are three horizontal segments. Why is the acceleration of the first equal to zero?
(d) Again from the acceleration graph, the value becomes negative for some time then positive for a shorter time. What is happening to the car here and why the different signs? **Hint:** acceleration is force / mass.
(e) Back to the velocity graph. At the end there are two segments both of which are negative. What is the car doing during these times?
(f) At what time does the braking kick in? How did you deduce this using one or more graphs?

3 Extension Work

Have a look at the extension sheet if you have time to do some investigations.