Development Record: BCP (Basic Computational Physics)

Program Plan: Bounce

Aim} produce a ball which freefall and bounces, include coefficient of restitution

Motive} get me started on basics of VPython, familiarize syntax

Pseudocode}}

loop=True

WHILE loop=True

displayscreen()

update(object.velocity)

update(object.position)

IF checkbounce(object.position)

update\_bounce(object.velocity)

{{

Version Log: Bounce 1

24/1/2017 --- 1.0

-bounce.py created

-ball object created

-added bounce mechanism

-added variable to control coefficient of restitution

-added simple event handling via pygame:

-spacebar leads to restart

-left and right arrow keys to add horizontal velocity to the ball

Version Review: Bounce 1.0

25/1/2017

The simple program was meant to start familiarizing me with the practice of coding motion of a 3D object. Program has worked as intended, and has served it’s purpose: I learnt how to use a dt variable to control rates, create 3D objects and set properties for these objects. No other sub-versions will be made.

Links

19/1/2017

https://www.wired.com/2015/08/coding-physics-course/

https://www.wired.com/2014/05/5-reasons-you-should-consider-a-different-physics textbook/

https://matterandinteractions.org/

https://www.glowscript.org/#/user/GlowScriptDemos/folder/matterandinteractions/program/MatterAndInteractions

21/1/2017

https://www.youtube.com/watch?v=E1B4UoSQMFw

https://en.wikipedia.org/wiki/L-system

22/1/2017

https://books.google.co.th/books?id=Y3jQNKC5dp8C&pg=PA8&lpg=PA8&dq=bounceability+measurement+unit&source=bl&ots=MU4SjIniNb&sig=hkEkOKMVXW0wurqZwKt3CuKZ\_VY&hl=en&sa=X&ved=0ahUKEwi35t\_ytdjRAhULtI8KHSjqCh0Q6AEIKzAD#v=onepage&q&f=false (page 9)

https://en.wikipedia.org/wiki/Coefficient\_of\_restitution

Overview Plan: Inverted Pendulum

25/1/2017 --- 1

Aim} simulate an inverted pendulum control system using A-lvl/self derived maths

Motive} to gain experience in approaching more advanced and practical computational physics. Also, to make a more expandable program than Bounce.

Pseudocode}}

While True:

calctheta(ball.pos.x)

clacthetadotdot(theta)

clacaccel(thetadotdot)

move(objects)

{{

Version Plan: Inverted Pendulum 1

25/1/2017 --- 1.0

-add floor, cart, bar, ball objects

-add gravity

-add motion of ball

27/1/2017 --- 1.1

-add graphing of data

-improve constant factor calculation (by using physics)

-improve damping calculation (by using physics)

Program Log: Inverted Pendulum 1

26/1/2017 --- 1.0

-invertpend.py created

-floor,cart,bar,ball objects created and linked

-added motion of ball

-added calculations for acceleration of cart

-added motion of cart

-added improvised constant factor

-added improvised damping mechanism

29/1/2017 --- 1.1

-abstraction created: arc length

-tinker control: changed from a multiplying factor to an incremented factor

-completed control system

-debug time/increment unit error

-debug time/increment derivation error

-realised incorrect formula derivation

Program Review: Inverted Pendulum 1

27/1/2017 --- 1.0

This program is a huge step away from Bounce, it is much more sophisticated and has many more potential applications. When looking at university level control system for an inverted pendulum I was fazed by the high level of maths involved (Lagarian equations) and was worried that high-school maths may not be enough to start the simulation. However I’ve been pleasantly surprised at how well the program simulates the motion and have gained confidence using equations of motion I derived myself. The control system can certainly be improved as it does not have the intended outcome from damping and the balance mechanism doesn’t work forever, however this has exceeded far beyond my expectations for a first version and I’m looking forward to working on the next one.

Also, I will need to write down my derivations of the equations used in an academic fashion.

31/1/2017 --- 1.1

Compared to sub-version 1.0, sub-version 1.1 is a much more completed sub-version of the version. At this point, I realised that I had derived incorrect equations of motion, though this realisation is only possible due to the program being more completed. Many bugs/error has been recognized and fixed, while an improved and functional control system was created. This sub-version had taken a significantly longer time to code, and the focus of this sub-version was on fixing and improving upon earlier faults in logic; this is the value in this project. Although the control system had not functioned according to reality, it functions properly according to the incorrect physics I had derived and has served its purpose as an exploration into the inverted pendulum problem.

Along the way, I had became familiar with coding in increments of time, which is the basis of all computational physics. Additionally, I’ve gained experiences of correcting my flaws in logic which would not be easy to accumulate had I not made these flaws myself. Thus,I strongly feel that this final version has satisfied my motive: “to gain experience in approaching more advanced and practical computational physics”.

The first version of this program has been a great learning experience and will assist me in building other programs or further versions. I feel that the Agile Development philosophy has quickened the pace of production of this program, due to its flexible style I did not need to waste time planning out everything (as it would have created errors anyways). New sub-versions of this program will not be made, however a new versions which uses different derivations is something to look forward to.

Links

25/1/2017

https://en.wikipedia.org/wiki/Inverted\_pendulum

https://en.wikipedia.org/wiki/Angular\_acceleration

https://www.control.isy.liu.se/student/tsrt19/ht2/file/invpendpmenglish.pdf

http://www.toddsifleet.com/projects/inverted-pendulum