CS 381 Test 1

Sushil J. Louis

March 8, 2010

You have 75 minutes for this test. There are a total of 50 points. Write your name and your major on the top right. Good luck.

1. (1 point) In python 8/3 evaluates to
   (a) 2
   (b) 3
   (c) 2.333
   (d) 2.666
2. (2 points) In python 7/-3 evaluates to
   (a) -2
   (b) -3
   (c) -2.333
   (d) -2.666
3. (2 points) In python, `word = 'Help' + 'A'` assigns
   (a) HelpA
   (b) Help A

   to `word`.
4. (5 points) In python, if `word = 'Helper'`, then
   (a) `word[-1]` evaluates to ________
   (b) `word[2:4]` evaluates to ________
   (c) `word[1:]` evaluates to ________
   (d) `word[:4]` evaluates to ________
   (e) `word[4] = 'x'` evaluates to ________
5. (5 points) In python, if `foo = ['spam', 'eggs', 100, 1234]`, then
   (a) `foo[2]` evaluates to ________________
   (b) After `foo[2] + 100`, `foo` evaluates to ________________

(c) After executing `foo[0:2] = [0, 0]` then `foo` evaluates to ________________

(d) Next (in sequence after c), after executing `foo[1:1] = ['yellow', 'sub']` then `foo` evaluates to ________________

(e) In sequence after d, `foo[:]  = []` then `foo` evaluates to ________________

6. (2 points) In python ogre, if the `frameStarted` method of a `frameListener` class instantiation returns False

   (a) Python-ogre runs the `frameStarted` method again

   (b) Python-ogre runs the `frameEnded` method immediately

   (c) Python-ogre shuts down

   (d) Python-ogre continues running

7. (2 points, True/False) In python ogre, all scene node positions are relative to the root scene node which is at the origin (0, 0, 0). ________________

8. (1 point, True/False) In python ogre, the positive z axis points into the screen away from the user. ________________

9. (5 points) Write the equation for a PID controller. Define **all** terms in your equation.

10. (5 points) Draw a very high level boxes and arrows diagram of the architecture of a game engine. Boxes represent major components and arrows represent directional information flow (method calls, pointers held, ...).

11. (5 points) Write a python function that computes and returns $n!$, where $0! = 1$ and $n! = n * (n-1)!$ for integer n

12. (5 points) Write a python function that would be called either one of two ways

    (a) factorial(n) or

    (b) factorial(n, factorialList = True)

    In the first case, the function behaves exactly as in the previous question: It returns $n!$. However, in the second case the function returns a tuple consisting of $n!$ and the list of all the factorials from 1 to $n!$. So, `factorial (3, list = True)` would return the tuple `(6, (1, 2, 6))` and `factorial (4, list = True)` would return the tuple `(24, (1, 2, 6, 24))`.

13. (10 points) Assume you have a vector class `Vector3`. The vector class implements `rotate2d, +, -, +=, -=` and you can construct a vector variable using `vectorVar = Vector3(0.0, 1.0, 2.0)`. Assume `vel, acc, pos` are variables containing the velocity vector, acceleration vector, and position vector respectively, of a boat entity in your game engine. Furthermore, `yaw, deltaYaw` are scalars representing the yaw (rotation around the y axis in python-ogre), and change in yaw respectively. `deltaT` is the time since the last update - you may not need `deltaT`. Write python game physics code to update the position and orientation of your boat between every frame rendered to the screen. Include "simple" friction acting opposite to engine thrust and proportional to velocity. You do not need to include rotational friction.