This question paper consists of 5 printed pages, each of which is identified by the Code Number COMP5813M.

This is a closed book examination. No material is permitted.

© UNIVERSITY OF LEEDS

School of Computing

January 2018

COMP5813M

Game Engines and Workflow

Answer 4 out of 4 questions

Time allowed: 2 hours

Question 1

(a) In platform games such as Super Mario, which subsystem prevents the character from falling through a platform?

    (A) Rendering

    (B) Physical simulation

    (C) Memory management

    (D) User I/O

[2 marks]

(b) Below, what functionality is NOT typically included in a version control system?

    (A) Log the history of a source file

    (B) Resolve conflicts between different versions of the same file

    (C) Roll back to a specific version

    (D) Log a bug found in the source code

[2 marks]

(c) M is a local-to-world transformation matrix and its equivalent is quaternion q. Given a local position represented by a column vector v, how to compute its global position w?

    (A) $w = vM$

    (B) $w = M^{-1}v$

    (C) $w = qv$

    (D) $w = qvq^{-1}$

[2 marks]

(d) When interpolating rotations by quaternions, what advantages does linear interpolation have?

    (A) It is usually quick

    (B) It can maintain a constant speed

    (C) It can maintain a constant angular speed

    (D) None of the above

[2 marks]

TURN OVER FOR QUESTIONS

(e) What is the key reason for having detailed control on the subsystem start and shut-down in game engines?

   (A) Programmers need to know all the details
   (B) To ensure the resources to be safely allocated/deallocated
   (C) So that the players do not have to worry about it
   (D) None of above

[**2 marks**]

(f) Why do programmers worry less about running out of memory on PCs or Macs compared to other game consoles, even if when they have relative smaller memory?

   (A) Even the memory is relatively small, it is still larger than the memory on game consoles
   (B) PCs/Macs have faster memory
   (C) Virtual memory mechanism
   (D) PC games are usually smaller thus require less memory

[**2 marks**]

(g) Character motion blending is NOT for?

   (A) Synthesising new motions
   (B) Correcting motion errors
   (C) Making style variations
   (D) Making motion transitions

[**2 marks**]

(h) When logging for your game, which of the below is the main disadvantage of using log files instead of printing information on the console?

   (A) Logging into files is usually slower
   (B) It is less straightforward to read lines in log files
   (C) It is hard to back-trace in log files
   (D) Log files take more spaces in hard drive

[**2 marks**]

TURN OVER FOR QUESTIONS

(i) What advantage does continuous collision detection have compared to discrete collision detection?

   (A) It makes simulation faster

   (B) It simplifies the collision detection process

   (C) It does not miss collisions between simulation steps

   (D) It has simpler mathematical representations.

[2 marks]

[question 1 total: 18 marks]

Question 2

(a) What are the pros and cons of different memory management strategies? Compare stack, double-ended stack and pool. **[6 marks]**

(b) When dealing with human interface devices (e.g. controllers, game pads, joysticks, etc.), the input is usually analogue signals. Describe what signal processing needs to be done and why. Give at least two strategies. **[6 marks]**

[question 2 total: 12 marks]

Question 3

(a) In games, accurately computing time is very important as it directly influences the perceived speed. Different strategies have been employed. Old games often specified the speeds in terms of meters (or pixels, or some other distance units) per frame. However, this causes distorted perceived speeds. Explain why. **[2 marks]**

(b) Later games used more advanced techniques including elapsed time and running average. How do they mitigate the previous problems? **[2 marks]**

(c) Finally, explain how a more principled approach, frame governing, can solve this problem. **[2 marks]**

[question 3 total: 6 marks]

Question 4

(a) In simulating a bouncing ball, a ball is positioned at a certain height, then it free-falls and bounces back when it hits the ground. However, even with restitution coefficient 1, when recording the highest positions each time the ball bounces back into the air, the positions recorded are not always the same. Give at least THREE possible reasons for this. **[6 marks]**

(b) When doing physical simulations, integrating an Ordinary Differential Equation/Partial Differential Equation is important. Integration error is a main source of error for animation. Explain how Forward Euler and Verlet Integration work and compare them. Mathematical explanations should be given on a simple example $\ddot{y} = \dot{y} = y$, where y is the position, $\dot{y}$ and $\ddot{y}$ are its first and second derivative, t is time and when $t_0 = 0$, $y_0 = \dot{y}_0 = 1$. Use the y_1 computed by Forward Euler to start the Verlet Integration. **[8 marks]**

[question 4 total: 14 marks]

[grand total: 50 marks]