Quiz 1 solutions

Question 1

Your friend is writing a game where it should create an object 15 seconds into the game. It's free-running, rather than locked to monitor refresh because they want the screen to update as fast as the CPU and GPU can possibly do it. Their code looks like this:

```
void Update() {
  if (Time.time == 15) {
     Instantiate(SomePrefab, SomePosition, SomeRotation);
  }
}
```

However, their game doesn't always create the object. It does on some runs on some machines, but doesn't on others. What's the problem? Please write the fixed version of the code below. Feel free to add fields or methods, if you need to.

Answer: Time.time is a floating point number. The odds of Update getting run exactly 15 seconds after the start of the game is near zero. Instead, do something like this:

```
bool created = false;
void Update() {
  if (!created && Time.time >= 15) {
     Instantiate(SomePrefab, SomePosition, SomeRotation);
     created = true;
  }
}
```

Question 2

Fill in the following table (check all that apply

	Expressible as a matrix operation in linear coordinates	Expressible as a matrix operation in projective coordinates
Scaling	Yes	Yes
Translation	No	Yes
Rotation	Yes	Yes
Translation + Rotation	No	Yes
Shift of coordinate system origin	No	Yes

Question 3

Fill in the following table (check all that apply)

	Conserves energy in real life	Conserves momentum IRL
Spring force	Yes	Yes
Gravity	Yes	Yes
Drag	No	No
Electrostatic force	Yes	Yes
Friction	No	No

Question 4

Your friend wrote this method of the Player class for their game:

```
void Fire() {
  var b = Instantiate(BulletPrefab, transform.position + transform.right,
                      Quaternion.identity);
  b.GetComponent<Rigidbody2D>().velocity = transform.right;
}
```

When your friend runs their game and fires, the bullet appears but doesn't move. Instead the player starts moving. Write the correction to the code on top of the code above.
Answer: in boldface, above.

Question 5

Your game is throwing an exception periodically. You know because the except is displayed in the Console window.

A. How do you find out what line of code is throwing the exception?
Answer: click on the exception message. It will display the methods on the stack and their files and line numbers. The top entry will be the line throwing the exception.

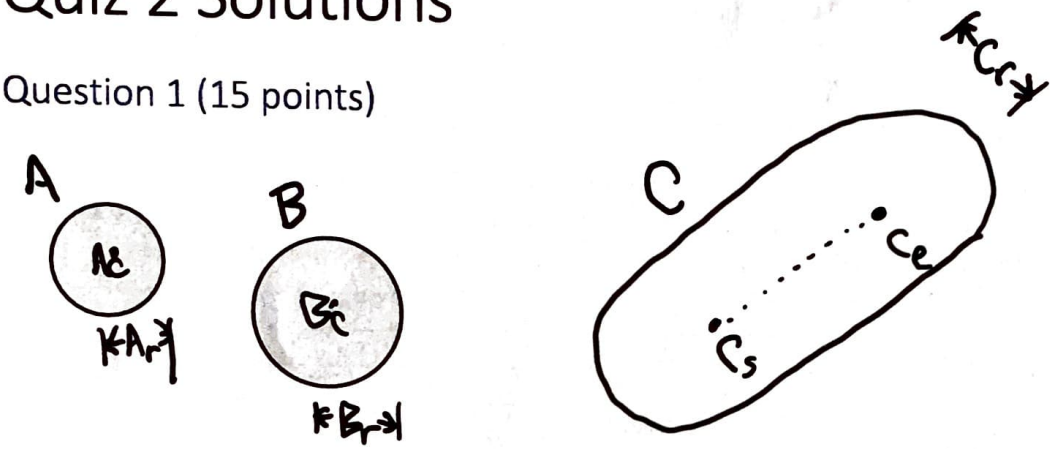
B. Great! Now that you know what line you need to breakpoint, explain how to connect Visual Studio to unity.
Answer: On windows, press Attach to Unity. On Mac, press the Play button in Visual Studio.

C. Now explain how to set the breakpoint.
Answer: Click to the left of the line you want to breakpoint. Alternatively, there's button in the debugger part of Visual Studio for Mac that lets you make a new breakpoint.

D. Finally, explain how to find the values of the variables in the line of code that's throwing the exception.
Answer: we will accept any of: look in the Locals window, look in the Autos window, or mouse over each variable in turn. We would even accept typing into the immediate window. I think that should work too, but I haven't tried it.

Game design and development
Quiz 2 Solutions

Question 1 (15 points)



Suppose you have a two spheres, A and B, and a capsule C:

- The centers of A and B are at A_c and B_c, respectively
- C has two "centers", C_s and C_e.
- The radii of the objects are A_r, B_r, and C_r, respectively

A. Give the expression for testing if A and B intersect you can give this as code, or just as math, whichever you prefer.

Answer: $\|A_c - B_c\| < A_r + B_r$

B. Recall that a line between two points x and y can be written as the set of points $x + s(y - x)$ for all s. When $s = 0$, this gives us x, and when $s = 1$, it gives us y, for values of s between 0 and 1, it gives us points along the segment connecting x and y, and for other values, it gives up points on the line, but beyond x and y. Suppose you have a function $\mathrm{f}(x, y, z)$ that tells you what value of s gives you the closest point in the line to z. Use this function to write an expression for the value of s for the closest point to z on just the line segment from x to y. You may want to use the functions $\min(a, b)$, which gives the smaller of a and b, and/or $\max(a, b)$, which gives you the larger of the two.

Answer: $\mathrm{f}(x, y, z)$ gives us the closest point along the line of xy to the z. However, that point may be outside of the segment connecting x, and y. To find the closest point on that segment, we just restrict the values of s to the range 0-1. So the answer is: $\max(0, \min(1, \mathrm{f}(x, y, z)))$.

C. Now assume you've used the answer to B to write a new function, $\mathrm{closest}(x, y, z)$, that gives you the actual point (rather than s value) on the segment xy to the point z. That is, even if you

couldn't answer B, you can still assume closest exists and use it in your answer. Give an expression to test whether sphere A intersects capsule C.

Answer: $\|A_c - \mathbf{closest}(C_s, C_e, A_c)\| < A_r + C_r$

Question 2 (20 points)

Suppose you are rendering with projection matrix P, model matrix M, and view matrix V.

A. For each of the following, give the coordinate system (model, world, camera, or screen) the matrix would map *from* and *to*. If the matrix doesn't make any sense as a coordinate transform, write "N/A". (12 points)

Matrix	Maps from	Maps to
Model	Model	World
View	World	Camera
Projection	Camera	Screen

B. Give the expression for the matrix that maps model coordinates to screen coordinates. (8 points)

Answer: PVM

Question 3 (20 points)

What is/are the difference(s) between perspective and orthographic projection? (check all that apply)

	Perspective	Orthographic
Varies object size with distance	X	
Can be written as a matrix in linear coordinates		X
Can be written as a matrix in projective coordinates	X	X
Simulates a camera lens	X	
Is parameterized by focal length	X	

Question 4 (20 points)

We talked in class about the theory of analyzing games in terms of the MDA framework. In this framework,

A. Which of part of the framework (mechanics, dynamics, or aesthetics) is the ultimate goal of the designer?
Answer: aesthetics

B. Which part does the designer have direct control over?
Answer: mechanics

C. Which part would a feedback loop fall under?
Answer: dynamics

D. Which parts drive/determine the others, as the player plays the game?
Answer: mechanics -> dynamics -> aesthetics

Question 5 (5 points)

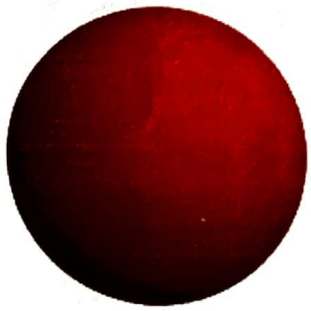
Suppose you are using quaternions for representing rotations. And suppose someone has given you an efficient procedure for raising a quaternion to a power, so that $q^2 = qq$ and $q^{0.5}$ is some other quarternion that when multiplied by itself gives you q, and so on. They've also given you an efficient procedure for computing Slerp, if you prefer to use that.

Suppose you have two rotations represented by quaternions q_0 and q_1. How could you compute a rotation halfway between the two?

Acceptable answers:

- $(q_1 q_0^{-1})^{0.5} q_0$
- $(q_0 q_1^{-1})^{0.5} q_1$
- $\mathbf{Slerp}(q_0, q_1, 0.5)$
- $\mathbf{Slerp}(q_1, q_0, 0.5)$

Question 6 (20 points)



A. Suppose you are rendering a mesh using diffuse (Lambertian) reflection, but not specular or ambient reflection. Which of the following properties would the vertices of the mesh need to be tagged with? That is, although you can always add extra attributes, what attributes would be *required* to be able to render the mesh with diffuse reflection? (check all that apply)

Coordinates of the vertex (i.e. where it is in space)	X
Texture coordinates	
Normal	X
Bone weights	
Material	

B. In this case, which of the following would the color of a given pixel on the mesh depend on? (check all that apply)

The orientation of the surface at that point	X
The direction of the light relative to the surface	X
The direction of the camera relative to the surface	
The color of the surface material	
The focal length of the projection	

X
X

ambient
diffuse
specular

Quiz 3 Solution

Q1. What Cannot be serialized by Unity.

- subclasses
- DAG cycle.
- graph.
- null reference.

Q2. correct / incorrect / fastest (in a row).

	Painter Algo. back → front	Rever painter front → back	Unsorted.
opaque z-buffering	correct	correct fastest	incorrect
opaque.	correct	Incorrect	correct
transparent z-buffering	incorrect	incorrect	incorrect. ← need α.
transparent.	correct.	correct.	

?

Q3. need coroutine.

```
IEnumerator Blah() {
    ...
    StartCoroutine();
    while ( -> t--) {
        if (...) {
            ...
        }
        yield return null;
    }
}
```

Q4. update position. ~~with Force & Mass given.~~ ~~for~~ if an object being hit by given Force & Mass. object its own.

```
void FixedUpdate() {
    ~~new po~~ ...
    Vector3 new position = ((Force/Mass) * Time.fixedDeltaTime) * fixedDeltaTime;
                                                                 Time.
    transform.position = new position;
    ...
}
```

Q5. Transformation from object's local coor to screen.

L_{lf}. left foot

L_{ul}. left leg.

L_b body.

M_c. camera center coor. (world to camera = viewMatrix^{-1}).

P.

$P M_c^{-1} L_b L_{ul} L_{lf}$.