



- [Go to your profile](#)
- [Hire a developer](#)
- [Apply as a developer](#)
- [Log in](#)

-
- [Top 3%](#)
- [Why](#)
- [Clients](#)
- [Enterprise](#)
- [Community](#)
- [Blog](#)
- [About Us](#)

[Go to your profile](#)
[Hire a developer](#)
[Apply as a developer](#)
[Log in](#)

- Questions?
- [Contact Us](#)

-
-
-

- Questions?
- [Contact Us](#)

-
-
-

[Hire a developer](#)

Find a world-class Unity or Unity3D developer for your team. [Hire Toptal's Unity or Unity3D developers](#)

12 Essential Unity or Unity3D Interview Questions [*](#)

- 973shares



-



-



-



-

[Submit an interview question](#)[Submit a question](#)

Looking for experts? Check out Toptal's [Unity or Unity3D developers](#).



Answer the following questions about threading. Explain your answers:

1. Can threads be used to modify a Texture on runtime?
2. Can threads be used to move a GameObject on the scene?
3. Consider the snippet below:

```
class RandomGenerator : MonoBehaviour
{
    public float[] randomList;

    void Start()
    {
        randomList = new float[1000000];
    }

    void Generate()
    {
        System.Random rnd = new System.Random();
        for(int i=0;i<randomList.Length;i++) randomList[i] = (float)rnd.NextDouble();
    }
}
```

Improve this code using threads, so the 1000000 random number generation runs without spoiling performance.

View the answer →Hide answer



1. No. Texture and Meshes are examples of elements stored in GPU memory and Unity doesn't allow other threads, besides the main one, to make modifications on these kinds of data.
2. No. Fetching the Transform reference isn't thread safe in Unity.
3. When using threads, we must avoid using native Unity structures like the `Mathf` and `Random` classes:

```
class RandomGenerator : MonoBehaviour
{
    public float[] randomList;

    void Start()
    {
        randomList = new float[1000000];
        Thread t = new Thread(delegate()
        {
            while(true)
            {
                Generate();
                Thread.Sleep(16); // trigger the loop to run roughly every 60th of a second
            }
        });
        t.Start();
    }

    void Generate()
    {
        System.Random rnd = new System.Random();
        for(int i=0;i<randomList.Length;i++) randomList[i] = (float)rnd.NextDouble();
    }
}
```



Explain what a vertex shader is, and what a pixel shader is.



View the answer →Hide answer



Vertex shader is a script that runs for each vertex of the mesh, allowing the developer to apply transformation matrixes, and other operations, in order to control where this vertex is in the 3D space, and how it will be projected on the screen.

Pixel shader is a script that runs for each fragment (pixel candidate to be rendered) after three vertexes are processed in a mesh's triangle. The developer can use information like the `UV / TextureCoords` and sample textures in order to control the final color that will be rendered on screen.



Explain why deferred lighting optimizes scenes with a lot of lights and elements.

View the answer →Hide answer



During rendering, each pixel is calculated whether it should be illuminated and receive lightning influence, and this is repeated for each light. After approximately eight repeated calculations for different lights in the scene, the overhead becomes significant.

For large scenes, the number of pixels rendered is usually bigger than the number of pixels in the screen itself.

Deferred Lighting makes the scene render all pixels without illumination (which is fast), and with extra information (at a cost of low overhead), it calculates the illumination step only for the pixels of the screen buffer (which is less than all pixels processed for each element). This technique allow much more light instances in the project.

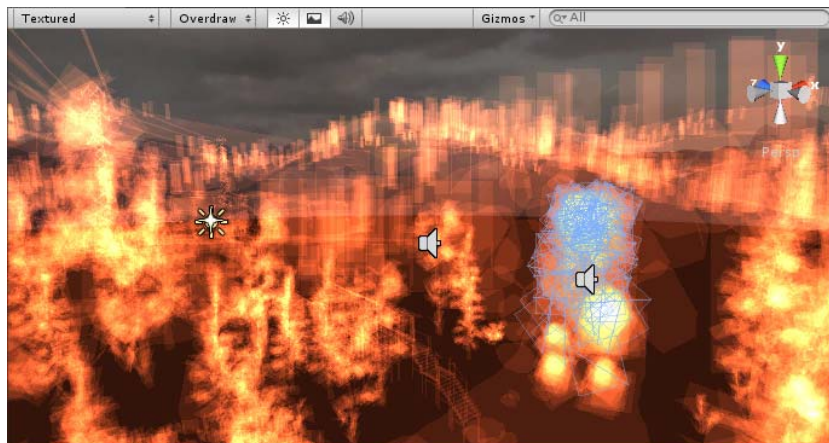
Find top Unity or Unity3D developers today. Toptal can match you with the best engineers to finish your project.

[Hire Toptal's Unity or Unity3D developers](#)



What are the benefits of having a visualization mode for rendering optimization, as shown on the picture bellow?





View the answer → Hide answer



The “overdrawn” mode helps the user to profile the number of pixels being rendered in the same “area”. Yellow to white areas are “hot” areas where too many pixels are being rendered.

Developers can use this information to adjust their materials and make better use of the *z-Test* and optimize the rendering.



Explain why `Time.deltaTime` should be used to make things that depend on time operate correctly.

View the answer → Hide answer



Real time applications, such as games, have a variable FPS. They sometimes run at 60FPS, or when suffering slowdowns, they will run on 40FPS or less.

If you want to change a value from *A* to *B* in 1.0 seconds you can't simply increase *A* by *B-A* between two frames because frames can run fast or slow, so one frame can have different durations.

The way to correct this is to measure the time taken from frame *x* to *x+1* and increment *A*, leveraging this change with the frame duration `deltaTime` by doing `A += (B-A) * DeltaTime`.

When the accumulated `DeltaTime` reaches 1.0 second, *A* will have assumed *B* value.





Explain why vectors should be normalized when used to move an object.

View the answer → Hide answer



Normalization makes the vector unit length. It means, for instance, that if you want to move with speed 20.0, multiplying `speed * vector` will result in a precise 20.0 units per step. If the vector had a random length, the step would be different than 20.0 units.



Consider the following code snippet below:

```
class Mover : MonoBehaviour
{
    Vector3 target;
    float speed;

    void Update()
    {
    }
}
```

Finish this code so the `GameObject` containing this script moves with constant `speed` towards `target`, and stop moving once it reaches 1.0, or less, units of distance.

View the answer → Hide answer



```
class Mover : MonoBehaviour
{
    Vector3 target;
    float speed;

    void Update()
    {
        float distance = Vector3.Distance(target, transform.position);

        // will only move while the distance is bigger than 1.0 units
        if (distance > 1.0f)
        {
            Vector3 dir = target - transform.position;
            dir.Normalize(); // normalization is obligatory
        }
    }
}
```



```
    transform.position += dir * speed * Time.deltaTime; // using deltaTime and speed is obligatory
  }
}
```



Can two `GameObjects`, each with only an `SphereCollider`, both set as trigger and raise `OnTrigger` events? Explain your answer.

View the answer → Hide answer



No. Collision events between two objects can only be raised when one of them has a `Rigidbody` attached to it. This is a common error when implementing applications that use “physics.”



Which of the following examples will run faster?

1. 1000 `GameObjects`, each with a `MonoBehaviour` implementing the `Update` callback.
2. One `GameObject` with one `MonoBehaviour` with an Array of 1000 classes, each implementing a custom `Update()` callback.

Explain your answer.

View the answer → Hide answer



The correct answer is 2.

The `Update` callback is called using a C# Reflection, which is significantly slower than calling a function directly. In our example, 1000 `GameObjects` each with a `MonoBehaviour` means 1000 Reflection calls per frame.

Creating one `MonoBehaviour` with one `Update`, and using this single callback to `Update` a given number of elements, is a lot faster, due to the direct access to the method.



Explain, in a few words, what roles the inspector, project and hierarchy panels in the Unity editor have. Which is responsible for referencing the content that will be included in the build process?

View the answer →Hide answer



The inspector panel allows users to modify numeric values (such as position, rotation and scale), drag and drop references of scene objects (like Prefabs, Materials and Game Objects), and others. Also it can show a custom-made UI, created by the user, by using Editor scripts.

The project panel contains files from the file system of the assets folder in the project's root folder. It shows all the available scripts, textures, materials and shaders available for use in the project.

The hierarchy panel shows the current scene structure, with its GameObjects and its children. It also helps users organize them by name and order relative to the GameObject's siblings. Order dependent features, such as UI, make use of this categorization.

The panel responsible for referencing content in the build process is the hierarchy panel. The panel contains references to the objects that exist, or will exist, when the application is executed. When building the project, Unity searches for them in the project panel, and adds them to the bundle.



Arrange the event functions listed below in the order in which they will be invoked when an application is closed:

```
Update()
OnGUI()
Awake()
OnDisable()
Start()
LateUpdate()
OnEnable()
OnApplicationQuit()
OnDestroy()
```

View the answer →Hide answer



The [correct execution order](#) of these event functions when an application closes is as follows:

```
Awake()
OnEnable()
Start()
Update()
LateUpdate()
OnGUI()
OnApplicationQuit()
OnDisable()
OnDestroy()
```

Note: You might be tempted to disagree with the placement of `OnApplicationQuit()` in the above list, but it is correct which can be verified by logging the order in which call occurs when your application closes.





Explain the issue with the code below and provide an alternative implementation that would correct the problem.

```
using UnityEngine;
using System.Collections;

public class TEST : MonoBehaviour {
    void Start () {
        transform.position.x = 10;
    }
}
```

View the answer →Hide answer



The issue is that *you can't modify the position from a transform directly*. This is because the position is actually a *property* (not a field). Therefore, when a getter is called, it invokes a method which returns a `Vector3` **copy** which it places into the stack.

So basically what you are doing in the code above is assigning a member of the struct a value that is in the stack and that is later removed.

Instead, the proper solution is to replace the whole property; e.g.:

```
using UnityEngine;
using System.Collections;

public class TEST : MonoBehaviour {
    void Start () {
        Vector3 newPos = new Vector3(10, transform.position.y, transform.position.z);
        transform.position = newPos;
    }
}
```

* There is more to interviewing than tricky technical questions, so these are intended merely as a guide. Not every “A” candidate worth hiring will be able to answer them all, nor does answering them all guarantee an “A” candidate. At the end of the day, [hiring remains an art, a science — and a lot of work](#).

Submit an interview question

Submitted questions and answers are subject to review and editing, and may or may not be selected for posting, at the sole discretion of Toptal, LLC.

Name	
Email	
Enter your question here	
Enter your answer here	
All fields are required	
<input type="checkbox"/> I agree with the Terms and Conditions of Toptal, LLC's Privacy Policy	
<input type="button" value="Submit a Question"/>	

Thanks for submitting your question.

Our editorial staff will review it shortly. Please note that submitted questions and answers are subject to review and editing, and may or may not be selected for posting, at the sole discretion of Toptal, LLC.

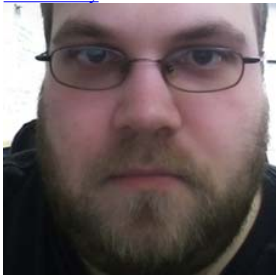
Looking for Unity or Unity3D experts? Check out Toptal's [Unity or Unity3D developers](#).



[View full profile »](#)[Alexey Pelykh](#)

Ukraine

Alexey is a seasoned software engineer with past work in 3D and computer game development, as well as extensive R&D experience. Currently, he works mostly with mobile and embedded platforms. He excels in the development of low-level software but does not shy away from UI design.

[Unity or Unity3DC++](#)[Hire Alexey](#)[View full profile »](#)[Cory Bechtel](#)

United States

Cory is a Game Developer who has created addictive, engaging games for the iOS, Android, and Amazon app stores. Highly experienced in C# and the Unity engine, his goal is to create great games that users want to play over and over again.

[Unity or Unity3DC++CJavaC#+5 more](#)[Hire Cory](#)[View full profile »](#)[Luko Gjenero](#)

Croatia

Luko is a seasoned developer with contributions to numerous large-scale projects currently in use by Croatian national organizations. He is schooled in electrotechnics, and has published several important research papers on a variety of topics. He enjoys being faced with challenging problems.

[Unity or Unity3DC++JavaC++9 more](#)[Hire Luko](#)

Toptal connects the [top 3%](#) of freelance talent all over the world.

Join the Toptal community.[Hire a developer](#)

or



[Apply as a developer](#)**Highest In-Demand Talent**

- [iOS Developers](#)
- [Front-End Developers](#)
- [UX Designers](#)
- [UI Designers](#)
- [Financial Modeling Consultants](#)
- [Interim CFOs](#)
- [Digital Project Managers](#)

About

- [Top 3%](#)
- [Clients](#)
- [Freelance Developers](#)
- [Freelance Designers](#)
- [Freelance Finance Experts](#)
- [Freelance Project Managers](#)
- [About Us](#)

Contact

- [Contact Us](#)
- [Press Center](#)
- [Careers](#)
- [FAQ](#)

Social

- [Facebook](#)
- [Twitter](#)
- [Google+](#)
- [LinkedIn](#)



Hire the top 3% of freelance talent™

- © Copyright 2010 - 2018 Toptal, LLC
- [Privacy Policy](#)
- [Website Terms](#)

By continuing to use this site you agree to our [Cookie Policy](#).
Got it

