CSYE 7270 – Building Virtual Environments
Quiz

Student Name: _____
Professor: Nik Bear Brown

Rules:
1. The quiz will be online via HackerRank
2. Ask the TA questions via Slack
3. The TA will post the password on Slack 5 minutes before the quiz

Q1 (6 Points)

What is the difference between sprites and tiles?

Solution:

A sprite is an image. It can be used in variety of ways in a computer game. A tile, on the other hand, is a bundle of data that might refer to a sprite.

A tile-based video game is a type of video or video game where the playing area consists of small square (or, much less often, rectangular, parallelogram, or hexagonal) graphic images referred to as tiles laid out in a grid.

Q2 (18 Points) Answer the following questions about the script below. Must explain answers. Yes/no answers given no credit.

```
public class Bullet : MonoBehaviour {

	public float speed = 20f;
	public int damage = 40;
	public Rigidbody2D rb;
	public GameObject impactEffect;

	// Use this for initialization
	void Start () {
		rb.velocity = transform.right * speed;
	}

	void OnTriggerEnter2D (Collider2D hitInfo)
	{
		Enemy enemy = hitInfo.GetComponent<Enemy>();
		if (enemy != null)
		{
			enemy.TakeDamage(damage);
		}

		Instantiate(impactEffect, transform.position, transform.rotation);

		Destroy(gameObject);
	}}
```

1. Which, if any variables are exposed to the inspector?
2. What direction does the bullet move when shot?
3. How does the player aim with this script?
4. How does a bullet know whether it hits an enemy?
5. If one wanted to change the script to destroy an enemy in one shot how would one do that?
6. Is there an animation or particle effect that plays when an enemy is hit?

Solution:

1. Which, if any variables are exposed to the inspector?

```
public float speed = 20f;
public int damage = 40;
public Rigidbody2D rb;
public GameObject impactEffect;
```

2. What direction does the bullet move when shot?

```
rb.velocity = transform.right * speed;
```

3. How does the player aim with this script?

Doesn't really. Can only aim by moving entire body as it uses the players transform

4. How does a bullet know whether it hits an enemy?

```
Enemy enemy = hitInfo.GetComponent<Enemy>();
Enemy script attached
```

5. If one wanted to change the script to destroy an enemy in one shot how would one do that?

```
enemy.TakeDamage(damage);   Increase the damage to more than the enemy starts with
```

6. Is there an animation or particle effect that plays when an enemy is hit?

Yes, something called `impactEffec`

```
Instantiate(impactEffect, transform.position, transform.rotation);
```

Q3 (10 Points) The CPU Usage Profiler module's chart tracks the time spent on which of the following categories?

1. Scripts
2. Environment X
3. Rendering
4. Physics
5. Animation
6. VSync
7. UI
8. Tiles X
9. Game Loop X
10. Input X

Solution:

It doesn't track
Environment, Tiles, Game Loop, Input

It tracks
Rendering, Scripts, Physics, Animation, GarbageCollector, Global Illumination, UI, Others

The CPU Usage Profiler module's chart tracks the time spent on which of the following categories?

Rendering - How much time your application spends on rendering graphics.
Scripts How much time your application spends on running scripts.
Physics How much time your application spends on the physics engine
.
Animation How much time your application spends on animating SkinnedMeshRenderers, GameObjects and other components in your application. This also includes the time spent on some calculations for systems the Animation and Animator components
utilize.
GarbageCollector - How much time your application spends on running the Garbage Collector.

VSync
How much time is spent in a frame waiting for the targetFrameRate or the next VBlank to sync with. This is according to the QualitySettings.vSyncCount value, or the target framerate, or the VSync setting that is the default or enforced maximum of the platform your application is running on. For more information about VSync, see the section in this documentation on Rendering and VSync samples.
Global Illumination How much time is spent on lighting in your application.
UI
How much time is spent on displaying your application's UI.
Others How much time is spent in code that does not fall in any of the other categories, for instance the entire EditorLoop, or the Profiling overhead when profiling Playmode in the Editor.

Q4 (18 Points) Answer the following questions about the script below. Must explain answers. Yes/no answers given no credit.

```
 public class PlayerMovement : MonoBehaviour {

        public CharacterController2D controller;

        public float runSpeed = 40f;

        float horizontalMove = 0f;
        bool jump = false;
        bool crouch = false;
        static bool ok = false;

        // Update is called once per frame
        void Update () {

                horizontalMove = Input.GetAxisRaw("Horizontal") * runSpeed;

                if (Input.GetButtonDown("Jump"))
                {
                        jump = true;
                }

                if (Input.GetButtonDown("Crouch"))
                {
                        crouch = true;
                } else if (Input.GetButtonUp("Crouch"))
                {
                        crouch = false;
                }
}

        void FixedUpdate ()
        {
                // Move our character
                controller.Move(horizontalMove * Time.fixedDeltaTime, crouch, jump);
                jump = false;
        }}
```

1. Why is `FixedUpdate ()` being used rather than everything in `Update ()`?
2. If one wanted to know the position of the game object that has this script, how would one determine that?
3. What is an "Axis"?
4. What is the purpose of multiplying horizontalMove by `Time.fixedDeltaTime?` What does Time.fixedDeltaTime do?
5. What data type is CharacterController2D?
6. What is a static variable?

Solution:

1. Why is `FixedUpdate ()` being used rather than everything in `Update ()`?

`FixedUpdate () is used for physics`

2. If one wanted to know the position of the game object that has this script, how would one determine that?

Its `transform.position`

3. What is an “Axis”?

It is a mapping of keys to a name, like horizontal or vertical.

4. What is the purpose of multiplying horizontalMove by `Time.fixedDeltaTime?` What does Time.fixedDeltaTime do?

`Time.fixedDeltaTime` is ihe interval in seconds at which physics and other fixed frame rate updates (like MonoBehaviour's FixedUpdate) are performed. It is used to adjust for the speed of different computers.

5. What data type is CharacterController2D?

A class called `CharacterController2D`

6. What is a static variable?

Static variables have a property of preserving their value even after they are out of their scope. Explaining how they are used is fine for full value.

Q5 (15 Points) The iPhone, iPad and iPod Touch devices are capable of tracking the following:
fingerId , deltaPosition, deltaTime, tapCount, and phase.

Explain what fingerId, deltaPosition, deltaTime, tapCount, and phase measure.

Solution:

fingerId The unique index for a touch.
position The screen position of the touch.
deltaPosition The screen position change since the last frame.
deltaTime Amount of time that has passed since the last state change.
tapCount The iPhone/iPad screen is able to distinguish quick finger taps by the user. This counter will let you know how many times the user has tapped the screen without moving a finger to the sides. Android devices do not count number of taps, this field is always 1.
phase Describes the state of the touch, which can help you determine whether the user has just started to touch screen, just moved their finger or just lifted their finger.
Began A finger just touched the screen.
Moved A finger moved on the screen.

Stationary A finger is touching the screen but hasn't moved since the last frame.
Ended A finger was lifted from the screen. This is the final phase of a touch.
Canceled The system cancelled tracking for the touch, as when (for example) the user puts the device to their face or more than five touches happened simultaneously. This is the final phase of a touch.

Q6 (18 Points) Answer the following questions about the script below. Must explain answers. Yes/no answers given no credit. Note the animator class has functions like *SetFloat* and *SetBool* that allows one to update the animators state using the following syntax.

animator.SetBool("variable_name", variable_name)

```
public class PlayerMovement : MonoBehaviour {

	public CharacterController2D controller;
	public Animator animator;

	public float runSpeed = 40f;

	float horizontalMove = 0f;
	bool jump = false;
	bool crouch = false;

	// Update is called once per frame
	void Update () {

		horizontalMove = Input.GetAxisRaw("Horizontal") * runSpeed;

    // 1. Write a line set an animator variable called Speed to the horizontal move speed.


		if (Input.GetButtonDown("Jump"))
		{
			jump = true;
			// 2. Write a line set an animator variable called IsJumping to true.
		}

		if (Input.GetButtonDown("Crouch"))
		{
			crouch = true;
		} else if (Input.GetButtonUp("Crouch"))
		{
			crouch = false;
		}

	}

	public void OnLanding ()
	{
		// 3. Write a line set an animator variable called IsJumping to false.
	}

	public void OnCrouching (bool isCrouching)
	{
		// 4. Write a line set an animator variable called IsCrouching to parameters
passed to this function.
	}

	void FixedUpdate ()
	{
		// Move our character
		controller.Move(horizontalMove * Time.fixedDeltaTime, crouch, jump);
		// 5. Reset the jump variable
	}
```

```
}
```

1. Write a line set an animator variable called Speed to the horizontal move speed below comment 1.
2. Write a line set an animator variable called IsJumping to true below comment 2.
3. Write a line set an animator variable called IsJumping to false below comment 3.
4. Write a line set an animator variable called IsCrouching to parameters passed to this function below comment 4.
5. Why is the function `OnLanding ()` public?
6. Reset the jump variable function below comment 5.

Solution:

```
public class PlayerMovement : MonoBehaviour {

        public CharacterController2D controller;
        public Animator animator;

        public float runSpeed = 40f;

        float horizontalMove = 0f;
        bool jump = false;
        bool crouch = false;

        // Update is called once per frame
        void Update () {

                horizontalMove = Input.GetAxisRaw("Horizontal") * runSpeed;

    // 1. Write a line set an animator variable called Speed to the horizontal move speed.

                animator.SetFloat("Speed", Mathf.Abs(horizontalMove));

                if (Input.GetButtonDown("Jump"))
                {
                        jump = true;
// 2. Write a line set an animator variable called IsJumping to true.

                        animator.SetBool("IsJumping", true);
                }

                if (Input.GetButtonDown("Crouch"))
                {
                        crouch = true;
                } else if (Input.GetButtonUp("Crouch"))
                {
                        crouch = false;
                }

        }

        public void OnLanding ()
        {
        // 3. Write a line set an animator variable called IsJumping to false.
                animator.SetBool("IsJumping", false);
        }

        public void OnCrouching (bool isCrouching)
        {
// 4. Write a line set an animator variable called IsCrouching to parameters passed to this
function.
                animator.SetBool("IsCrouching", isCrouching);
        }
```

```
        void FixedUpdate ()
        {
                // Move our character
                controller.Move(horizontalMove * Time.fixedDeltaTime, crouch, jump);
// 6. Reset the jump variable
                jump = false;
        }
}
```

Why is the function `OnLanding ()` public?

So it can be called by another class.

Q7 (15 Points) Unity Analytics allows one to measure and visualize funnels. What is a funnel?

Solution:

Unity Analytics

Funnels help you track whether players make it through linear sequences in your game. A funnel reveals what percentage of your players make it through each step of the sequence as well as the sequence as a whole.