BUILDING WORLDS: MIDTERM (Spring 2013) Name: _____

- This midterm is open-computer / open-book.
- **Write legibly.** I have to be able to read what you wrote.
- Ask me if you have a question about a question.
- Please don't share answers. Please don't cheat.
- You'll get partial credit if you were only kinda wrong, but on the right track.
- You have 60 minutes.

10 PRINT : DID YOU READ IT?

What are "platform studies" and "software studies", and what is the general difference between them?

Does 10PRINT produce a unicursal maze? Why or why not?

Explain how **10 PRINT CHR$(205.5+RND(1)); : GOTO 10** makes a pattern.

When coding in BASIC, why did coders usually add new line numbers in increments of 10?

What is the main difference between the original 10 PRINT and the Apple II port of 10 PRINT?

CODE COMPREHENSION

Here's a code sample written for Processing, but it's pretty similar to what we've been doing in Unity with C#. Remember that an RGB color value is made of 3 floats, 0-1.

```
1   for (int x = 0; x < image.width; x++) {
2    for (int y = 0; y < image.height; y++) {
3
4     int index = x + y*image.width;
5
6     float r = image.pixels[index].r;
7     float g = image.pixels[index].g;
8     float b = image.pixels[index].b;
9
10    float d = distance(x,y,mouseX,mouseY);
11    float adj = (50-d)/50;
12    r *= adj;
13    g *= adj;
14    b *= adj;
15
16    r = constrain(r,0,255);
17    g = constrain(g,0,255);
18    b = constrain(b,0,255);
19
20    color c = color(r,g,b);
21    pixels[index] = c;
22   }
23  }
```

constrain() is like a clamping function. Why is it necessary here? (lines 16-18)

if the **image** is 256 pixels wide and 128 tall, then how many times will the innermost for() loop execute?

If a white pixel (1, 1, 1) is 25 units away from the mouse cursor, then what color will that pixel be after this code runs?

(_____, _____, _____)

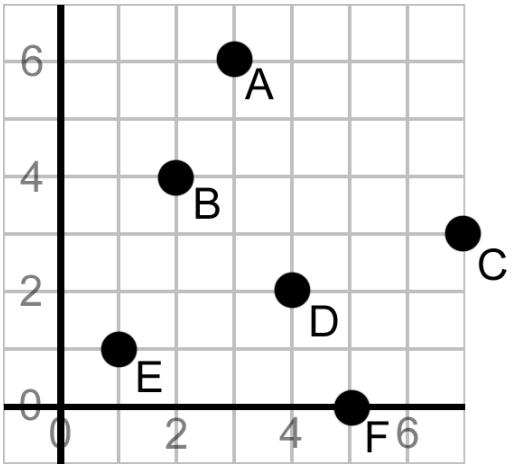
UNITY PROJECT : BUG-FIXING

For our purposes, a "bug" is an error that prevents Unity from compiling / entering play mode. For each bug you fix, write the filename, the line number, explain the bug / how you fixed it.

e.g. "CubeMove.cs: 20, 'Transform' should've been lowercase."

VECTOR MATH

You shouldn't need a calculator for any of these problems.
Leave any fractions as fractions. "SqrMagnitude" means don't take the square root.
Answers should contain no letters.



1. AB = (___, ___)
2. DC = (___, ___)
3. FA = (___, ___)
4. BC = (___, ___)
5. BC.sqrMagnitude = ____
6. AC.normalized = (___, ___)
7. DE + (6,-2) = (___, ___)
8. BD + (___, ___) = EB
9. CD + (DB * 10) = (____, ____)
10. FB.normalized * AC.magnitude = (___, ___)

UNITY REFERENCE

1. How often / when is **Update()** called?

2. Generally, what is the difference between an **array** and a **list**?

3. If we shoot a _____ for 1000 units and it hits a collider, it'll return a bool value of ____

4. When / why would you want to expose a variable or function as **"public"**?

5. When / where can you use a **"yield"** instruction?

6. How often / when is **Start()** called?

7. When and why would you multiply values with **Time.deltaTime**?

8. Match each transform property with its equivalent in local vector space:

___ transform.forward	a. (0, -1, 0)	f. (0, 1, 1)
___ transform.right	b. (-1, 0, 0)	g. (0, 0, 1)
___ -transform.right	c. (1, 0, 0)	h. (0, 0, -1)
___ transform.up	d. (1, 1, 1)	
	e. (0, 1, 0)	

UNITY CODE ARCHITECTURE : WHY IS THIS CODED THE WAY IT IS?

```
public class Flyer : MonoBehaviour {
	public float speed = 5f;
	Vector3 target;
	public float targetRange = 10f;

	void Update () {
	    transform.position += (target-transform.position).normalized * Time.deltaTime * speed;
	    if ( (target - transform.position).magnitude < 0.2f) {
	       SetNewTarget();
	    }
	}

	public void SetNewTarget () {
		Vector3 newTarget = new Vector3 (Random.Range(-targetRange, targetRange),
		                                 Random.Range(-targetRange, targetRange),
		                                 Random.Range(-targetRange, targetRange));
		SetNewTarget(newTarget);
	}

	public void SetNewTarget (Vector3 newTarget) {
		target = newTarget;
	}
}
```

1. In the first function overload of SetNewTarget(), which takes no parameters, it calls the second overload. Why would anyone do that, why not just put “target = newTarget” instead?

2. Why is the class member “target” left as a private variable?

3. In first line of Update(): what is the normalized vector, and why do we have to normalize it?

4. Given the Flyer.cs script above, finish this “FlyerStopAndGo.cs” script:

```
public class FlyerStopAndGo : MonoBehaviour {    // makes Flyer bounce between stopping and going
	Flyer flyer;
	void Start () {
	// grab reference to the Flyer script on the same GameObject
		flyer = _________________<_________>();
	}
	void Update () {
	// use sine wave to make flyer’s velocity “bounce” back and forth between 0-2 as game runs
		flyer._________ = ___ + _________._________(_______________);
	}
}
```

BUILDING WORLDS: MIDTERM (Sp 2013) Name: _____

- This midterm is open-computer / open-book.
- **Write legibly.** I have to be able to read what you wrote.
- Ask me if you have a question about a question.
- Please don't share answers. Please don't cheat.
- You'll get partial credit if you were only kinda wrong, but on the right track.
- You have 60 minutes.

CODE COMPREHENSION

Here's a code sample written for Processing, but it's pretty similar to what we've been doing in Unity with C#. Remember that an RGB color value is made of 3 floats, 0-1.

```
1   for (int x = 0; x < image.width; x++) {
2    for (int y = 0; y < image.height; y++) {
3
4     int index = x + y*image.width;
5
6     float r = image.pixels[index].r;
7     float g = image.pixels[index].g;
8     float b = image.pixels[index].b;
9
10    float d = distance(x,y,mouseX,mouseY);
11    float adj = (50-d)/50;
12    r *= adj;
13    g *= adj;
14    b *= adj;
15
16    r = constrain(r,0,255);
17    g = constrain(g,0,255);
18    b = constrain(b,0,255);
19
20    color c = color(r,g,b);
21    pixels[index] = c;
22   }
23  }
```

constrain() is like a clamping function. Why is it necessary here? (lines 16-18)

if the **image** is 64 pixels wide and 128 tall, then how many times will the innermost for() loop execute?

If a white pixel (1, 1, 1) is 20 units away from the mouse cursor, then what color will that pixel be after this code runs?

(, ,)

10 PRINT : DID YOU READ IT?

Explain how **10 PRINT CHR$(205.5+RND(1)); : GOTO 10** makes a pattern.

Why is the RND() function considered to be “pseudorandom”?

Explain how Claude Shannon’s 1950 mechanical mouse “solved” mazes.

Is the Apple II port of 10 PRINT better or worse than the original version? Why or why not?

Does 10PRINT produce a unicursal maze? Why or why not?

UNITY REFERENCE

1. How often / when is **Awake()** called?

2. How often / when is **Update()** called?

3. Generally, what is the difference between an **array** and a **list**?

4. What’s the difference between **Update()** and **FixedUpdate()**?

5. When / where can you use a **“yield”** instruction?

6. When / why would you want to expose a variable or function as **“public”**?

7. When and why would you multiply values with **Time.deltaTime**?

8. Match each static vector property with its world vector equivalent.

___ Vector3.right	a. (0, 1, 1)	f. (1, 1, 1)
___ Vector3.up	b. (-1, 0, 0)	g. (1, 0, 1)
___ Vector3.one	c. (1, 0, 0)	
___ Vector3.forward	d. (0, 0, 1)	
	e. (0, 1, 0)	

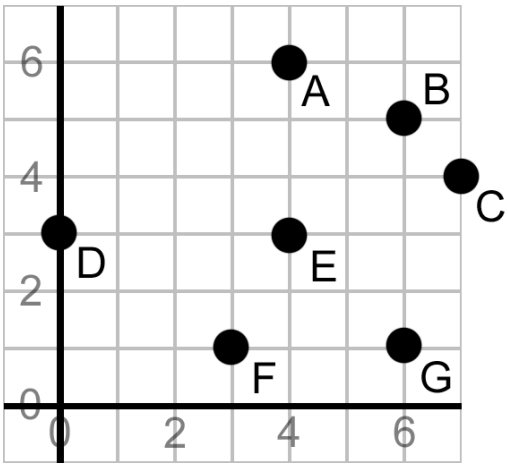
UNITY PROJECT : BUG-FIXING

For our purposes, a "bug" is an error that prevents Unity from compiling / entering play mode.
For each bug you fix, write the filename, the line number, explain the bug / how you fixed it.

e.g. "CubeMove.cs: 20, 'Transform' should've been lowercase."

VECTOR MATH

You shouldn't need a calculator for any of these problems.
Leave any fractions as fractions. "SqrMagnitude" means don't take the square root.
Answers should contain no letters.



1. AB = (___, ___)
2. DC = (___, ___)
3. FA = (___, ___)
4. BC = (___, ___)
5. BC.magnitude = ____
6. AD.normalized = (___, ___)
7. DE + (3,-2) = (___, ___)
8. BE + (___, ___) = GE
9. CG + (DB * 10) = (____, ____)
10. FB.normalized * DA.magnitude = (___, ___)

UNITY PROJECT DESIGN : WHY IS THIS CODED THE WAY IT IS?

```
public class Flyer : MonoBehaviour {
	public float speed = 5f;
	Vector3 target;
	public float targetRange = 10f;

	void Update () {
	    transform.position += (target-transform.position).normalized * Time.deltaTime * speed;
	    if ( (target - transform.position).magnitude < 0.2f) {
	       SetNewTarget();
	    }
	}

	public void SetNewTarget () {
		Vector3 newTarget = new Vector3 (Random.Range(-targetRange, targetRange),
		                                 Random.Range(-targetRange, targetRange),
		                                 Random.Range(-targetRange, targetRange));
		SetNewTarget(newTarget);
	}

	public void SetNewTarget (Vector3 newTarget) {
		target = newTarget;
	}
}
```

1. In the first function overload of SetNewTarget(), which takes no parameters, it calls the second overload. Why would anyone do that, why not just put “target = newTarget” instead?

2. Why is the class member “target” left as a private variable?

3. In first line of Update(): what is the normalized vector, and why do we have to normalize it?

4. Given the Flyer.cs script above, finish this “FlyerStopAndGo.cs” script:

```
public class FlyerStopAndGo : MonoBehaviour {    // makes Flyer bounce between stopping and going
	Flyer flyer;
	void Start () {
	// grab reference to the Flyer script on the same GameObject
		flyer = _________________<_________>();
	}
	void Update () {
	// use sine wave to make flyer's velocity "bounce" back and forth between 0-2 as game runs
		flyer._________ = ___ + _________._________(_____________);
	}
}
```