**Topic: Simple Movement**

**Overview:**  Like physics, movement in 2D is seductive, but necessary for the first phase of GenderlessInGermany. Its worth knowing because it is the one place where GetComponent is unnecessary. This tutorial illustrates old-fashioned keyboard movement, and includes a fancier technique where the camera remains on the player and the scene moves around. Movement based on velocity can be found in the Unity tutorials and extends the basic ideas here. The code is reasonably self-explanatory and is intentionally sparsely commented, asking you to research the details in the Unity documentation.

**Your Turn (Please do the following):**

1. Open the project FetchTheBall. Notice that the assets are organized into folders, Scenes scripts and Sprites. Its worth getting into the habit in order to keep things organized. Select the scene Camera Static. Run it and then use the arrow keys to move around. Which sprite moves? Open the script for OliverControl (which you will notice in the inspector is attached to Oliver.) Notice that in the Start metho, the transform attached to Oliver can be referenced without a GetComponent call. This is because the transform component is a bound variable in all components that can be rendered. This makes sense because updating position on the screen needs to happen as quickly as it can.
2. Notice that a transform’s position is a Vector3 object, even in 2D where the ‘z’ position should be set to zero. Try changing the initial position of Oliver.
3. Update uses object syntax. The ‘Input’ object (there is only one, its called a static class), has a GetKey function that evaluates whether the current key is the parameter. KeyCode is another static class that has pubic variables for each possible key on the keyboard. Therefore Input.GetKey(KeyCode.UpArrow) asks the question ‘Was the last input from the keyboard the up arrow key?”. Experiment with changing the keys that are used to control Oliver.
4. The transform a has a method ‘Translate’ that takes three arguments, the x, y, and z components respectively. Explore the documentation for other options for translating a game object. You might be tempted to play with rotation, but leave that for now. There’s some geometry that is necessary for rotation that gets nasty quickly.
5. What is that Time.deltaTime thing? Time is also a static class. Its attribute deltaTime is the amount of time that lapses between updates. Using this value as a multiplier for movement helps smooth the movement. Again, try changing this value to see the impact.
6. Now switch to the ‘Follow Ball’ scene, run it, press the arrow keys and then study the ‘Main Camera’ in the inspector. Notice that it has a script that has a public variable that is linked to the ball. What’s going on? At the start the camera calculates an offset of the player object – in this case the ball. Look up ‘LateUpdate’ in the documentation. Do the math to convince yourself you see how the camera keeps its position relative to the ball, and Oliver appear to move around the scene.
7. Now switch to the ‘Follow Oliver’ scene, and run it. This one is counter-intuitive. Oliver listens to the event handlers and moves, but the camera follows him! So the ball appears to move in the scene. This technique is used in the 2D UFO tutorial to keep the player at center stage. Have some fun with this technique.
8. A challenge… what happens if the both Oliver and the Ball can move based on keystrokes? What happens when the camera is trained on one of them?

**Exploring More in Unity:** check out Velocity which produces an entirely different kind of movment.

**Object Oriented Wrap Up:** Two subtle ideas were introduced here: (1) public variables like transform, (2) Static classes that you simply call upon for information, but that you cannot instantiate.