**Lab 1 – Simple Obstacle Course (Instructor Manual)**

**Part 1:**

* Load in Part1 scene (just a blank world)
* Introduce students to the Unity interface and the basic rules
  + Scenes: contain environments and objects of your program (think of each scene as a level in a game)
  + GameObject: an object in your environment that can interact with other objects and be referred to by other objects
  + Components: properties that can be assigned to a GameObject
  + The environment should follow basic laws of physics
* Show how to create the ground and player object
  + Create the cubes
  + Adjust dimensions and positions
  + Assign pre-made component for ground and player
* Put player box above ground and start play simulation
  + QUESTION: why does the box not fall as expected?
  + Answer: the box is not recognized as an object with mass → need to add Rigidbody component to player object
* Show how to add a C# script to the player object to control movement
  + Explain how C# script is just a customized component
  + Add P1PlayerMovement script to Player object
  + Explain the Start() and Update() functions
* TODO: play around with the different properties for the player object (e.g. size, color, etc.) and get the Player object to continuously move forward in a straight line when the simulation starts
  + Give the hint of the rb variable referring to the player itself and built-in functions for Rigidbody for applying forces to objects
  + If the function is correct but the box looks like it is being kicked → hint at Slippery component (need to add to Player and Ground objects to get it to move smoothly)
* BONUS: figure out how to get the camera to move with the player
  + Assign FollowPlayer script to the Main Camera and adjust the Z value under the FollowPlayer component of the Main Camera

**Part 2:**

* Load in Part2 scene (the simple obstacle course)
* Explain the task: finish the code relating to the movement logic for the player object so it can navigate past all the obstacles
  + The logic for maneuvering is whenever the player object collides with an obstacle, it will dodge to the right or left depending on which way is open, otherwise it will determine there is no path (i.e. it has reached the finishing line) and stop
* Navigate to the P2PlayerMovement script
* TODO: implement the following functions:
  + Update(): for forward movement (hint: it will be similar but not exactly the same as Part 1)
  + OnCollisonEnter(): only the relevant section relating to the decision to dodge or stop
  + Dodge()
  + DodgeLeft()
  + DodgeRight()
* BONUS: implement ChangeColor() function to change the color of player object based on the input
  + E.g. if ChangeColor(“red”) is called, the box should turn red
  + If successful, try calling function at different parts of the decision-making process to visualize what the code is doing

**Post-Lab Answers:**

1. What type of logic was used to determine whether the player object should dodge?

→ OR logic

1. Can you think of another simpler solution to get the player object to move past the obstacles?

→ Add force to the Y axis to get it to jump over the obstacle (1 possible answer)