# Notes

These instructions are meant as a guide, and may be modified as desired or required. You may run this at your own pace. These are intended for instructors who know the concepts already and assume that the content provided in the slides associated with this document has also been presented.

**Code Blocks are in the Appendix and are numbered for reference.**

# UI Walkthrough

Begin by resetting your layout to the default:

Window > Layouts > Default

Walk through each section of the UI:

* Hierarchy
  + GameObject create, select
* Inspector
  + Edit transform for an object
  + Show camera as example of components and their editors
* Scene
  + Use tools to move the Main camera object around
  + MMB for Pan
  + RMB for Pivot (rotate)
  + LMB for select and drag select
  + WSAD for movement
* Game
  + Triggered on pressing play
* Project
  + Filters
  + Create options
  + Searching
* Console
* Menus
* Main/Top Toolbar
  + Hand, Translate, Rotate, Scale
  + Global/Local Transform Tools
  + Play/Pause

Demo that the UI can be dragged around to adjust as desired and saved in the Window > Layouts menu, or using the top rightmost drop-down (which should display Default at the moment unless customized).

# Demo #1

1. Create a Plane
   * Pos 0,0,0
   * Scale 10,10,10
   * Name “Ground”
2. Add a directional light
   * 0,0,0
3. Add a sphere
   * Name “PlayerSphere”
   * Position it above the ground, in front of the camera
4. Select the light and change shadow type to Hard Shadows
5. Add a folder to assets named Materials
6. Add a material to that folder named GroundMat
   * Colour 73, 206, 129
7. Add a new material named PlayerMat
   * Colour 206, 82, 73
8. Drag material onto sphere in scene view
9. Select the player sphere and add a rigid body component
10. Edit > Project Settings > Input
11. Expand Axes, change size to 2
    * Show and explain Horizontal/Vertical options
12. Change Horizontal to Jump
    * Change buttons, set a single button to space
    * Clear other options
13. Add a scripts folder
14. Add a C# script named PlayerMovement
15. Assets > Sync MonoDevelop project
16. Open PlayerMovement.cs in the generated solution
17. At the top of the script add “public float Power =40;”
18. Inside Update() add Code Block 1
19. In the editor drag the script onto PlayerSphere
20. Change Angular Drag of sphere to 1.0
21. Add “public float JumpPower = 300;” to the top of PlayerMovement
22. Add Code Block 2 to Update()
23. Create a new Physics Material in Assets, name it BouncyMaterial
24. Drag the material onto the ground plane