HowTo: Set up Unity camera and canvas for 2D work

For Unity 5.4, but probably works with little change for Unity 4.x as well.

By Todd D. Vance

1. Make some decisions.

- Let's say, the game aspect ratio is Rx by Ry. For example, the standard definition ratio would be Rx=4, Ry=3. Standard def portrait mode would be Rx=3, Ry=4. High def is Rx=16, Ry=9. High def portrait mode is Rx=9, Ry=16. These are the most common choices. **Set your Rx = _____ and Ry = _____ here.**
- Let's say we want the game screen (including any borders) divided into game units so that it is w game units wide. To keep things an integer, w should be an exact multiple of Rx. Let r be that multiple, so that r*Rx = w, or r = Rx/w. For example, if using standard definition Rx=4 by Ry=3 and r is 3, w would be 12, so the screen is 12 game units wide (and r*Ry = 9 game units high). **Set your r = _____ here.**
- Set your target resolution X by Y. It is important that X/Rx = Y/Ry. Possible choices: if using high def, 1080p is a common resolution, so Y = 1080 and X = Y*Rx/Ry = 1080*16/9 = 1920. High def portrait would be the opposite aspect: Y=1920 and X=1080. Standard def, a common size is X=800 by y=600. Alternatively, set according to the size of the background image you intend to use, but still make sure the ratio is so that X = Y*Rx/Ry or equivalently, Y = X*Ry/Rx. **Set your X = _____ and Y = _____ here.**
- Decide on the padding around the game screen. The game screen minus padding we shall call "game space". The padding is measured in game units. This gives space for score, navigation buttons, whatever needed for the game. There are four padding values, for left, right, top, bottom, labeled Pl, Pr, Pt, and Pb, respectively. Possible padding values might be Pl=Pr=Pt=Pb=0.5 so if the game screen is r*Rx by r*Ry, then the game space is r*Rx-Pl-Pr by r*Ry-Pt-Pb. (alternatively, decide the game space dimensions first, and adjust Rx, Ry, r, and padding to make it work). **Set your Pl = _____, Pr = _____, Pt = _____, and Pb = _____ here.**

2. Make some calculations. Now that we have the basic numbers Rx, Ry, r, X, Y, Pl, Pr, Pt, and Pb, we can compute other numbers needed to set up the game space.

- Canvas scale: this is the size of the canvas in game units divided by the size in pixels, and is computed by r*Rx / X, which had better be the same value as r*Ry / Y. **Compute your canvas scale here: s = r*Rx/X = _____.**
- Camera size: in Unity, this is the camera half height in game units. Thus, it is r*Ry/2. **Compute your camera size here: c = r*Ry/2 = _____.**
- Camera and canvas offsets: We want the (0,0) position to be the bottom left of the game space. To do this, we set the x and y offsets of both the camera and canvas to the half width minus left padding by the half-height minus the bottom padding. Thus, Ox = r*Rx/2 - Pl and Oy = r*Ry/2 - Pb. **Compute your offsets here: Ox = r*Rx/2 - Pl = _____ and Oy = r*Ry/2 - Pb = _____.**
- Finally, this is not something to be set, but to be kept in mind when laying out and scripting the game. The game space has lower-left coordinates (0, 0) and upper-right coordinates (Gx, Gy) = (r*Rx – pl – pr, r*Ry – pt – pb). **Compute your game space limits here: Gx = r*Rx – pl – pr = _____ and Gy = r*Ry – pt – pb = _____.**

3. Set it all up in Unity

- Fire up Unity and start a new project in 2D mode and save the new scene with a reasonable name.
- Select the Game tab and set the aspect ratio to your Rx:Ry.
- Select the Main Camera in the Hierarchy window and set Camera.Size to your c and Transform.Position x and y to your Ox and Oy, respectively.
- Select your scene (root node) in the Hierarchy window, right click, select Game Object → UI → Canvas to create a new canvas that is not a child of any other game object.
- With Canvas selected, change Canvas.Render_Mode to World Space (not World Space Camera).
- With Canvas selected, change Canvas.Event Camera to the Main Camera (by dragging from the Hierarchy or selecting from the circle-dot to the right of the input box).
- Set Rect_Transform Width and Height to your X and Y, respectively.
- Set Rect_Transform.Scale X and Y both to your s.
- Set Rect_Transform PosX and PosY to your Ox and Oy, respectively.
- Test! Make sure the Scene tab is selected. In the Hierarchy, double-click the Main Camera and then double-click the Canvas. Both the Main Camera and Canvas should show as identical rectangles.
- Profit.

4. Optional: You can do more depending on the game being created.

- If the game piece is, say, 256x256 and you want the game piece to be 1x1 game unit in the game, then select the sprite for the game piece asset in the Project section, and set pixels per unit to 256. In general, if you want the piece to be axb in the game, and its resolution is AxB, set ppu to A/a, which should equal B/b. If not, you might need to either adjust the scale or edit the image for the sprite in an image editor (such as Gimp) to change the ratio.
- If you want a background that is the tiling of an image, set the PPU the same way based on number of game units wide and high you want each tile to be. Set the size to the same as the canvas size (X by Y) in pixels. Set the tiling dimensions to the number of tiles wide by number of tiles high. So if tiles are 1x1 game units, this would be r*Rx by r*Ry.
- To ensure the background tiling is behind everything, it might go on a separate canvas from foreground UI stuff. Set it up the same way as the previous canvas was set up, and give it Pos Z value something greater than zero. The UI canvas, which is to be on top of things, could then get a Poz Z value less than zero, but remember it should still be greater than the Camera's Z value (which is -10 by default).Then, your game stuff can go at Z=0 and be in the right places.