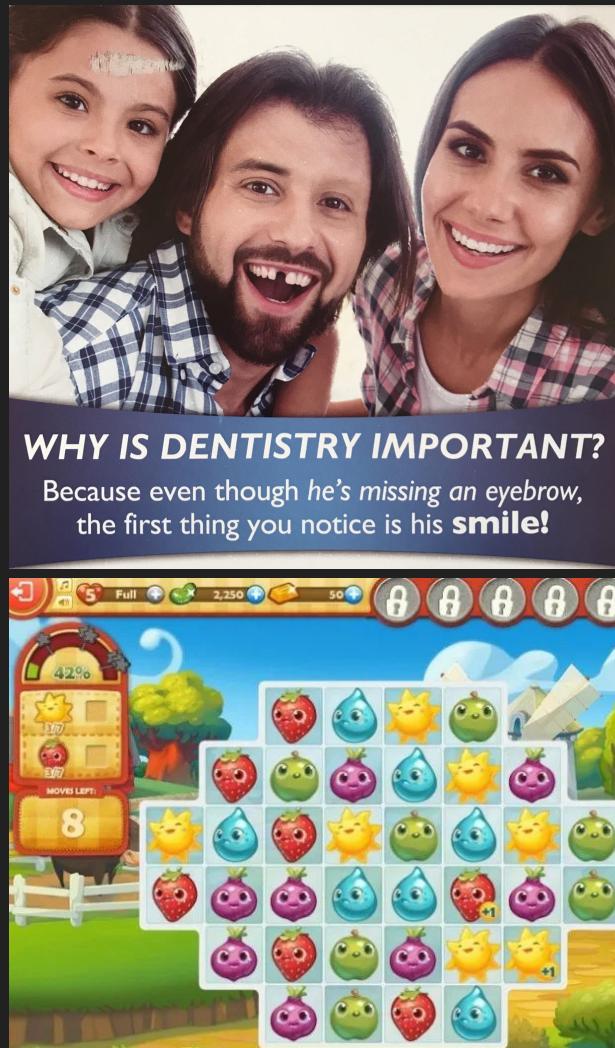


06 - Menus and User Interface

CS 3160 - Game Programming
Max Gilson

Menus and UI

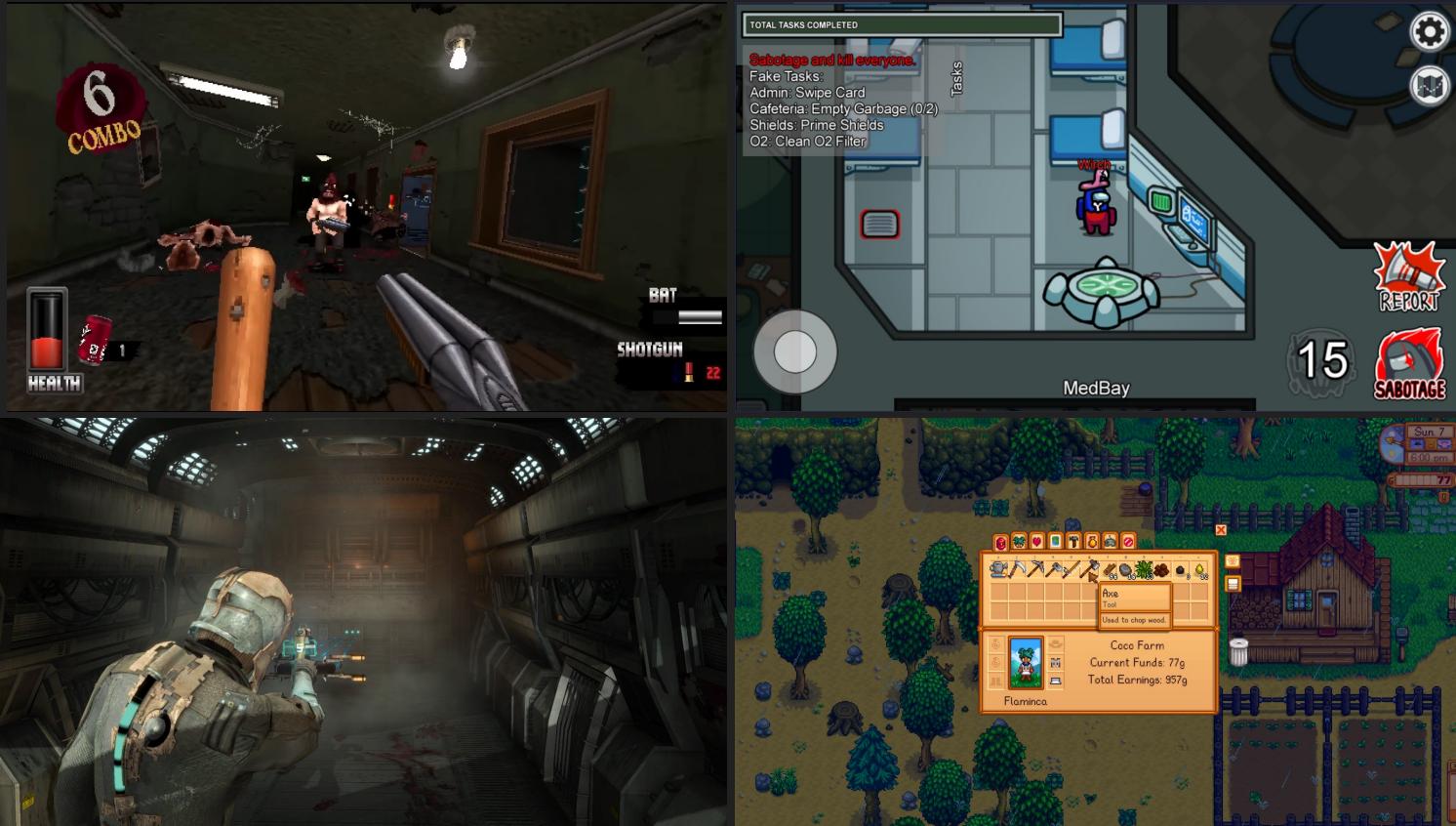
- Menus and UI are the first things your player sees and players set their expectations immediately upon seeing them
- A nice menu and UI system is just as important as adding movement, puzzles, characters, levels, items, etc. to your game
- Bad or clunky UI will:
 - Ruin the player's experience with your game
 - Convince prospective players that your game is low quality



“Bad” UI



“Good” UI



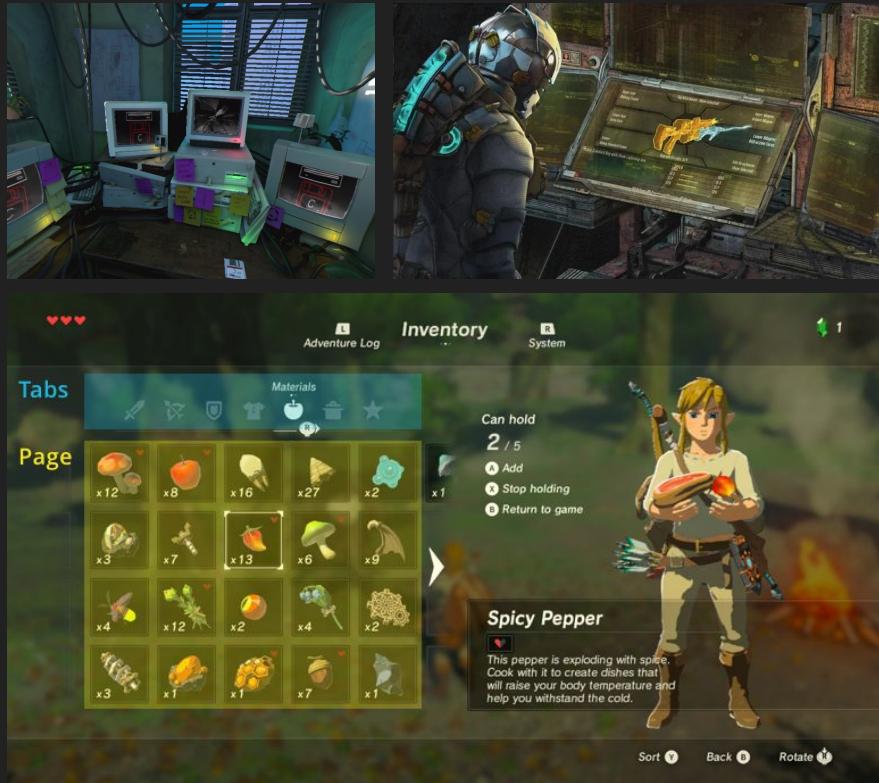
Types of UI

- Non-diegetic UI
 - Not visible to the characters in the world
 - Diegetic UI
 - A part of the game world
 - Meta UI
 - Loading screens, title screen, controls menu
 - Spatial UI
 - Fixed in 3D space

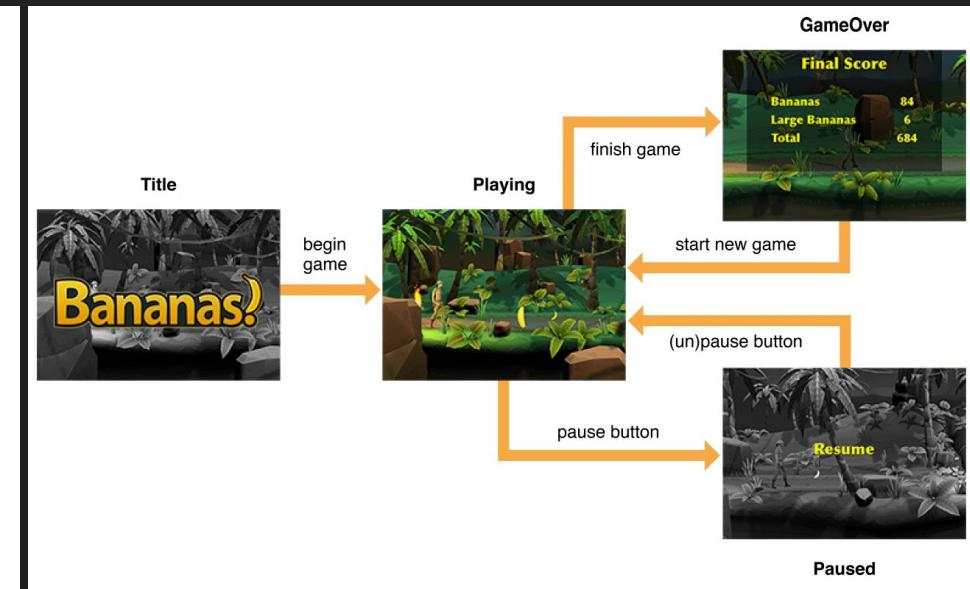
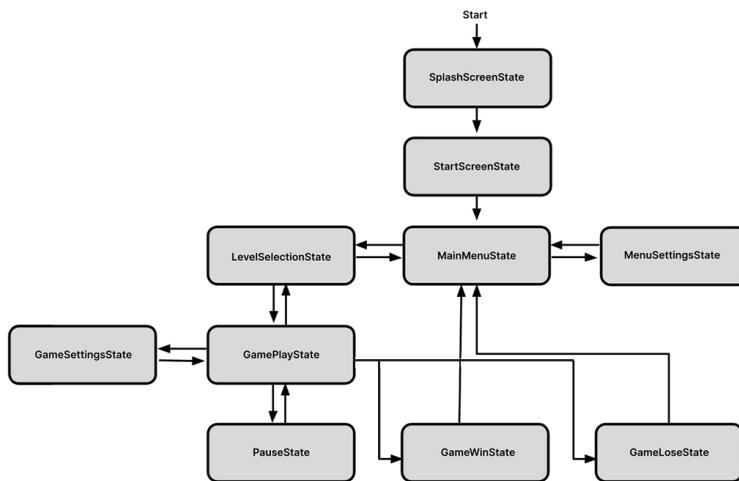


Menus

- A menu is a UI element that allows the player to make selections and control aspects of the game
 - Main menu
 - Pause menu
 - Controls menu
 - Save/load menu
 - Inventory menu
- Menus can be diegetic, non-diegetic, meta, or spatial
- You want menus to be as friction-less as possible
 - Friction unintentionally complicates or slows down the player's experience
 - May require implementing customized state machines or other techniques

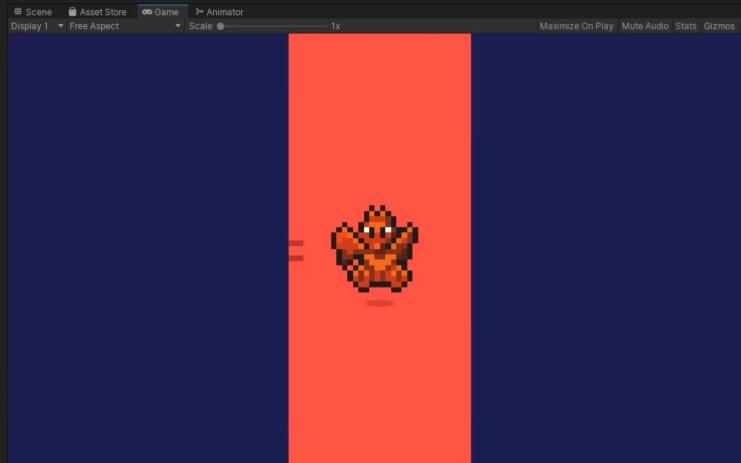
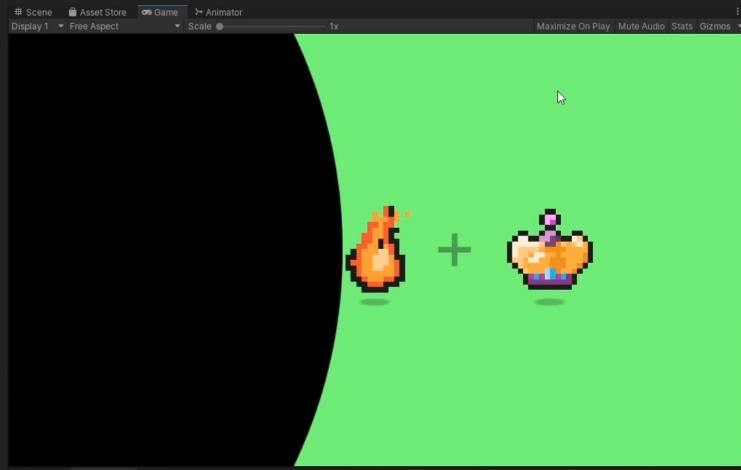


Menu State Machine Example



Menu Transitions and Animations

- Transitioning between menus could either be instantaneous or involve some animations:
 - Fade In/Out
 - Slide In/Swiping
 - Stacking/Layering
- https://youtu.be/CE9VOZivb3I?si=vtP1nSH30ib_U0S



UI Elements

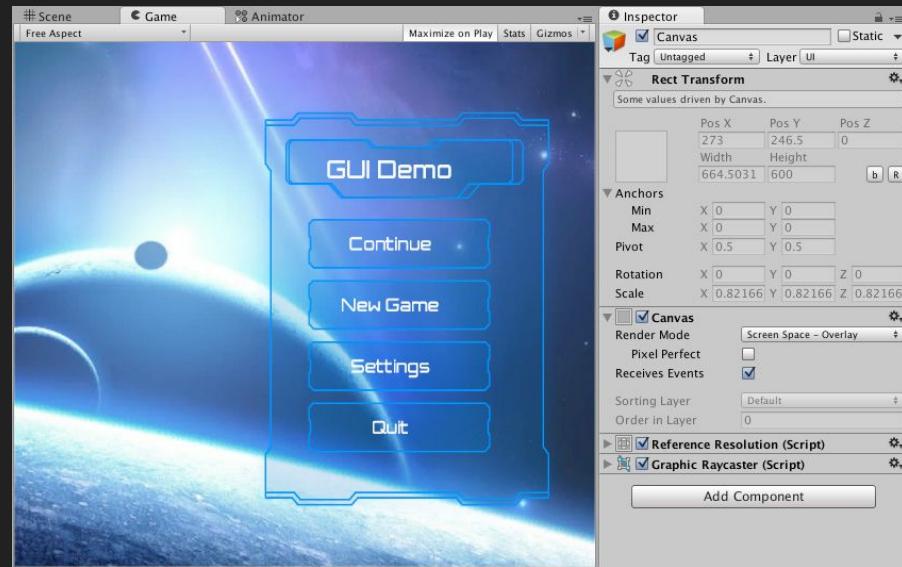
- UI elements can either be directly interactable or non-interactable
- Interactable:
 - Buttons, sliders, checkboxes, windows
- Non-interactable:
 - Indicators, health bar, score number, ammo count
- The player can interact with the UI either by selecting/dragging with a mouse or joystick, tapping a button prompt, etc.



Best Hit 4%
Hit Percent 74%
Hit Notes 179/239
Avg. NPS 12.49 nps
Fret Ghosts 0

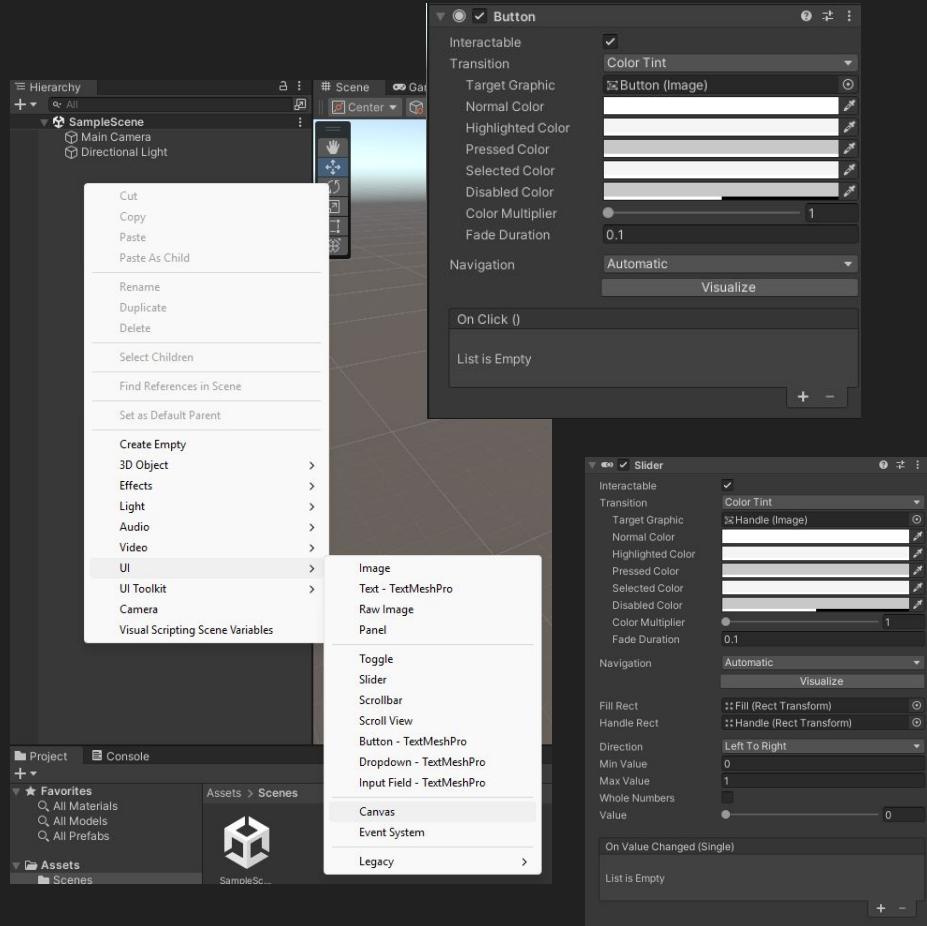
UI Elements in Unity

- Unity provides tools for creating UI elements:
 - Canvas - holds other UI elements
 - TextMeshPro - a text label
 - Image - displays an image
 - Button - allows for calling functions inside of scripts when clicked
 - Slider - slides a float value across a line i.e. 0.0 to 1.0
 - Input field - a text box
- Put your canvas and all relevant UI inside of its own GameObject and enable it or disable it when needed
 - <https://assetstore.unity.com/packages/essentials/ui-samples-25468>



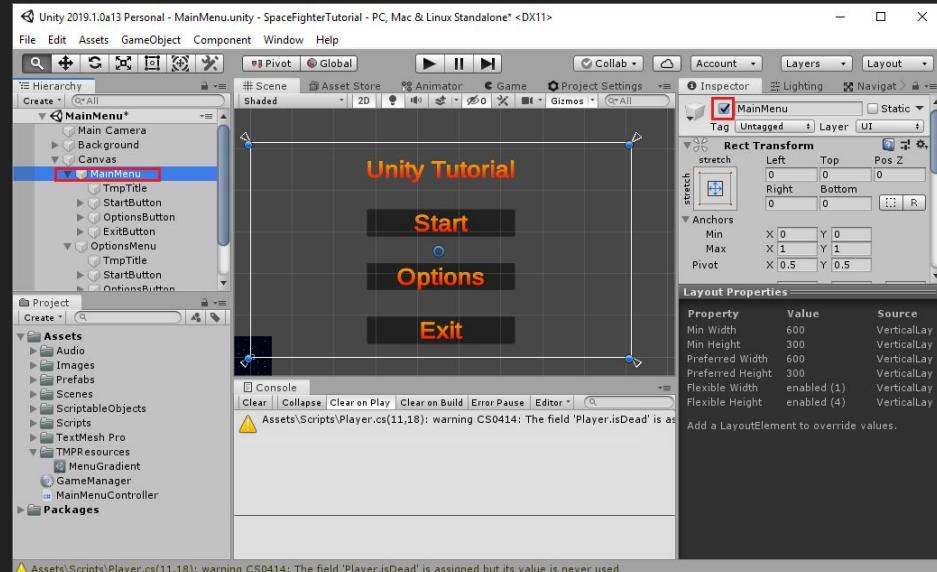
UI Elements in Unity (cont.)

- Create a new Canvas in the Scene
- Create new child GameObjects for different “screens” or menus
- Add more child GameObjects for each UI element
- Add a new “Button - TextMeshPro”
- In the Button Component, add an On Click () callback for any script in your Scene
 - Can also be used for adding sound effects
- Add a new “Slider”
- In the Slider Component, add an On Value Changed (Single) callback
 - You can also modify the slider’s value via a script:
 - `mySlider.value = 0.75f;`
- As you can see, making UI look good and work properly is a lot of work



UI Elements in Unity (cont.)

- A common way to approach menus in Unity is to use a child for each sub menu
 - Example: an options menu might have multiple sub menus for audio, video, controls, etc.
- Enable/disable each of the children when they are needed via a dedicated “UIController” script
 - Use “menu.SetActive(true)”
- Multiple UI elements as children are sorted by the order in which they appear in the hierarchy
 - This can be overridden if necessary
- For a simple pause menu use Time.timeScale
 - Time.timeScale = 0f; // Pauses
 - Time.timeScale = 1f; // Resumes
 - Can also be used for a slow motion effect
- For an options menu, use PlayerPrefs to save the player’s settings



UI Elements in Unity (cont.)

- Layout Groups make it easy to organize UI horizontally, vertically, or in a grid
 - Horizontal Layout Group
 - Vertical Layout Group
 - Grid Layout Group
 - Layout Element (for overrides)



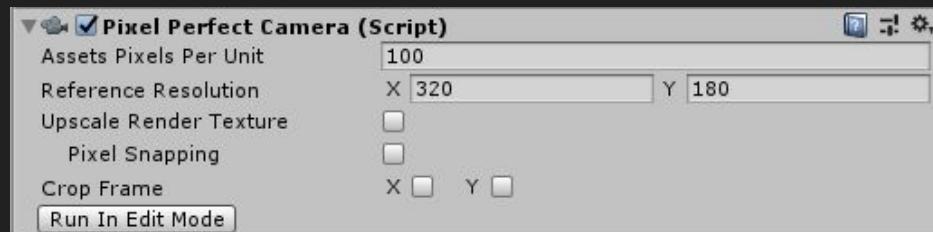
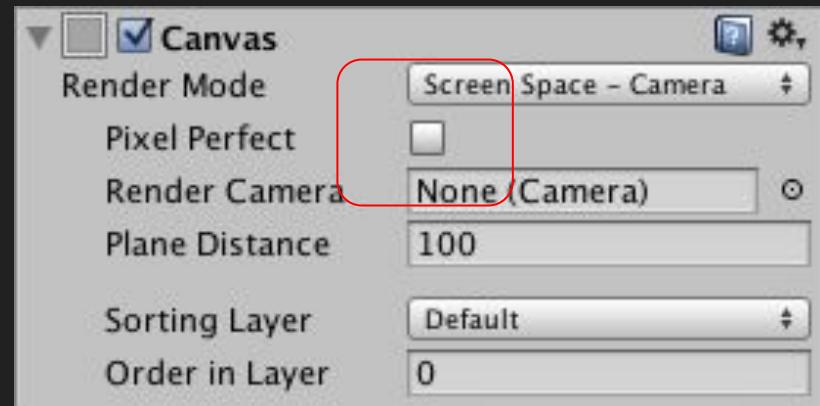
UI Elements in Unity (cont.)

- It is often recommended that you use TextMeshPro for UI in Unity
 - Requires installing an extra package
- Uses the TextMeshProUGUI type



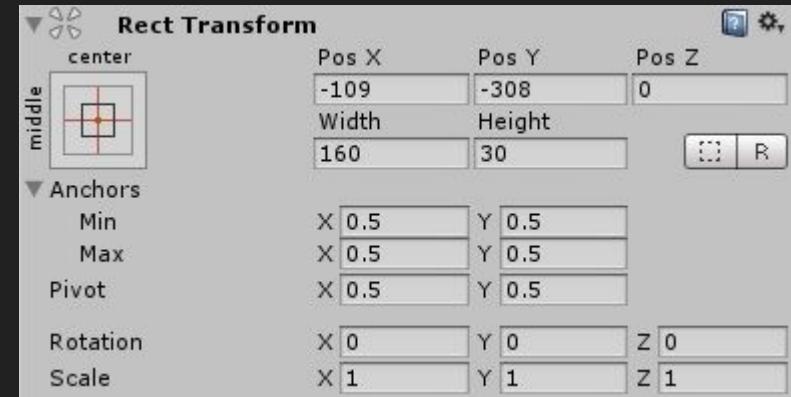
UI Elements in Unity (cont.)

- If using pixelated UI, you may want to use a pixel perfect canvas
- May also require using a pixel perfect UI render camera
 - Attach the “Pixel Perfect Camera” component to an additional Camera in your scene



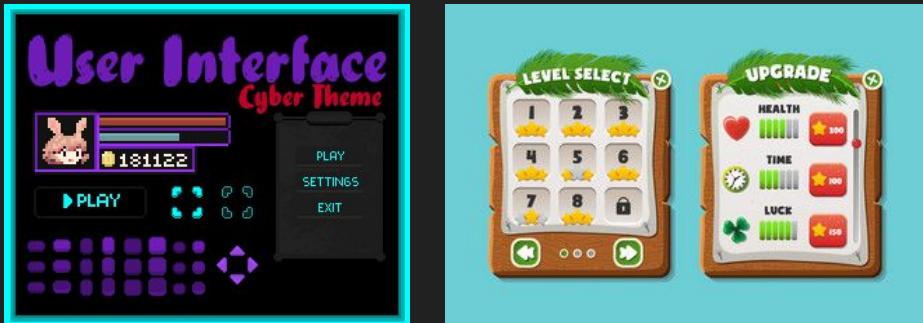
UI Elements for Multiple Screen Sizes in Unity

- You could create unique UI menus for every platform sized perfectly
 - This is time consuming
- Instead, Unity allows for specifying anchors and allows for dynamic resizing based on screen size
 - By default, UI elements size is specified in pixels
 - This can result in UI looking too small on high resolutions or too small on low resolutions
- This is all done via the Rect Transform and Canvas Scaler Components



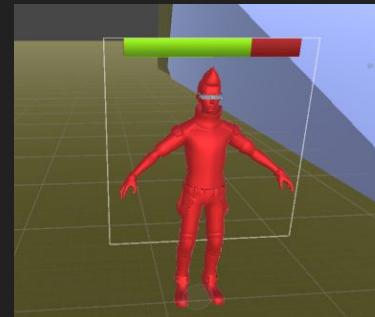
Custom UI in Unity

- You can overwrite any of the default graphics used for UI in Unity
 - Sliders, buttons, text, etc.
- <https://itch.io/game-assets/free/tag-user-interface>



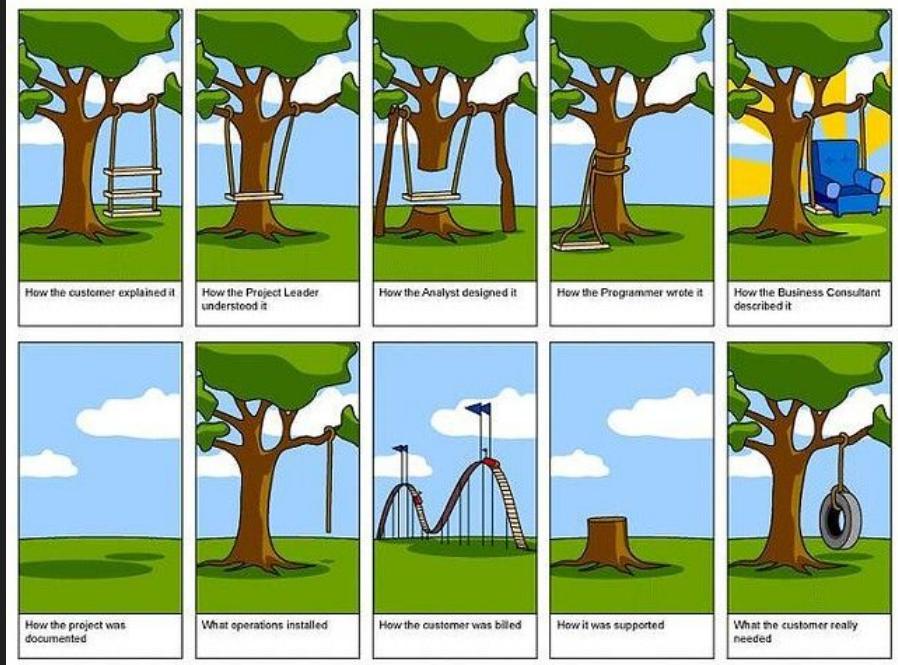
World Space UI in Unity

- In Unity, you can create “World Space” Canvas which will have UI elements inside of the game world
- This can be useful if you need UI to be present in the world
 - Example: a hologram or bulletin board
- Will look small at a distance
 - Use when appropriate



Making a Good UI

- As a game programmer, you are often implementing the vision of a graphic designer or being the graphic designer yourself
 - If the font, size, color, text, look, feel, etc. is wrong, you are often the one to blame
- You should have a strict attention to detail when implementing UI to avoid mistakes



Designing UI

Coding UI



Making a Good UI (cont.)

- Clarity - UI should be easy to read and understand at any distance
 - Even if you squint, you should generally be able to see what is most important (colors help!)
- Consistency - UI should maintain consistent look and controls
 - If you right-click to equip an item that is on the floor, it should work the same in inventory or shop menu, etc.
- Intuitive - UI should be easy to navigate and require minimal learning curve
 - Follow existing design patterns and avoid friction when possible
- Focused - UI should be focused, clearly showing only what is absolutely necessary
 - Too much information will stress out the player and prevent them from making informed decisions
- Responsive Feedback - UI should provide audio/visual feedback and feel responsive
 - If there is no indication that a user can interact with the UI it can be confusing
- Screen Scaling - UI should be sized properly for the screen and aspect ratio
 - Mobile UI element sizes are much larger than an ultra-wide screen PC monitor
- Accessibility - UI should provide options for making it easier to understand for all users
 - Color blind settings, scalable text and UI, audio cues, etc.

Good UI is Game Dependent

- Different games have different UI requirements
- Implementing good UI requires a deep understanding of your genre and audience's expectations
 - Strategy gamers want more info, not less
 - Horror gamers want more immersion, fewer distractions

