Graphical User Interfaces

Interfacing with a virtual world

Programming – Computer Graphics

SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Contents

SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

User Interfaces

- User Interfaces refer to the ways the user interacts and interfaces with your game world
  - Through interaction with the game via some control method
    - i.e. Keyboard, Mouse, Controller, Touch
  - Through visual feedback from the game
    - i.e. Game world, Heads-Up-Display (HUD) and Graphical User Interface (GUI), Character animation
  - Through audio feedback from the game
    - i.e Sound cues, NPC “barks”

- We will mostly be discussing the 2nd definition
  - In particular the HUD and Graphical User Interfaces (GUI)
  - Used to display information that does not make sense in the game world

- Can add to or distract from the game world

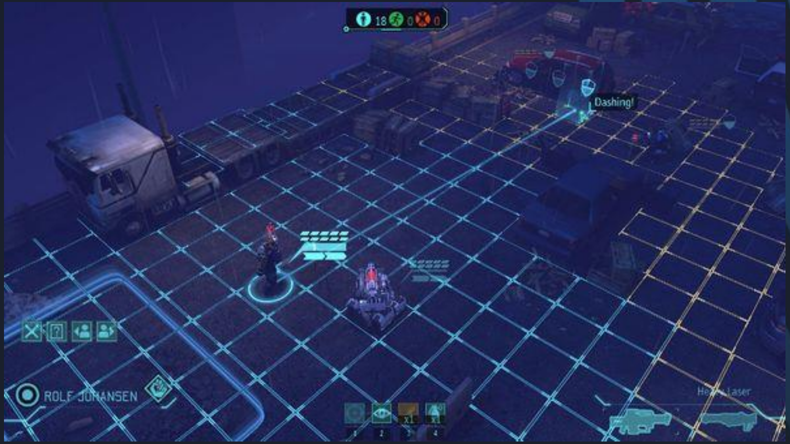
GUI Design

- A good GUI generally has the following features:
  - Simple to navigate
    - No more then 3 button presses to set/get the information you need
  - Important information should be immediately visible and accessible
    - You don’t want to have to go through 3 menu items to see your current health
  - If information can be displayed in game, it is preferable to do so rather then as part of the GUI
    - It still needs to be displayed clearly

SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

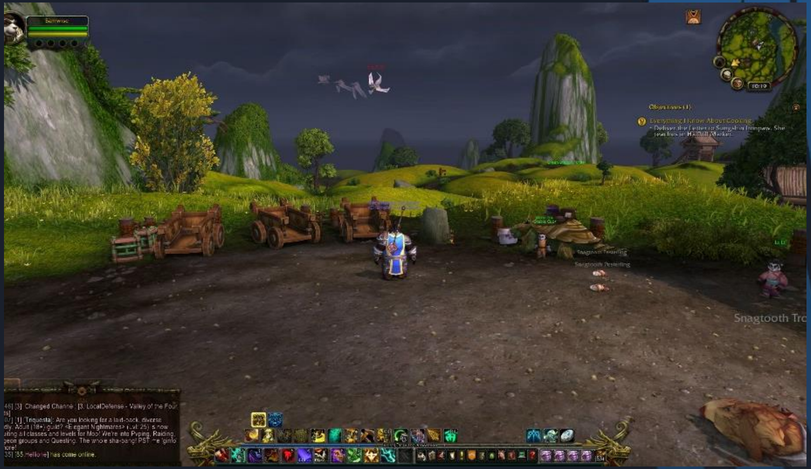
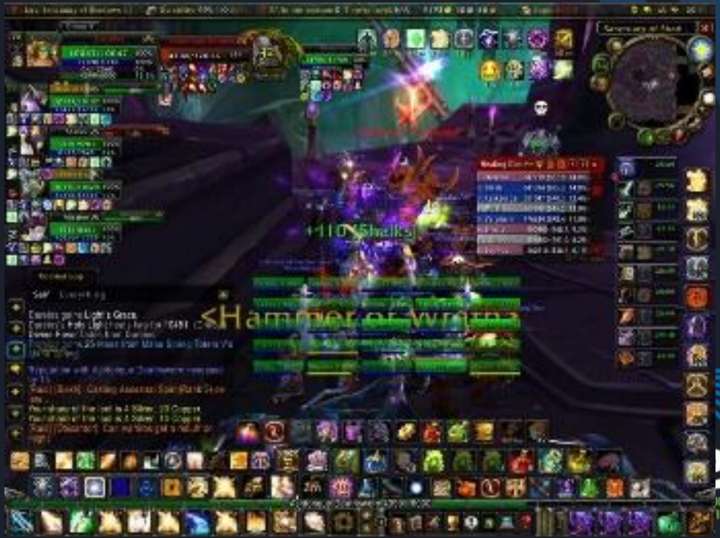
GUI Design – An example

- XCOM: Enemy Unknown has the following GUI
- Much of the GUI information is incorporated into the game world
  - The grid overlay for example
- HUD elements are clean, show only the information that is required, and require no menus



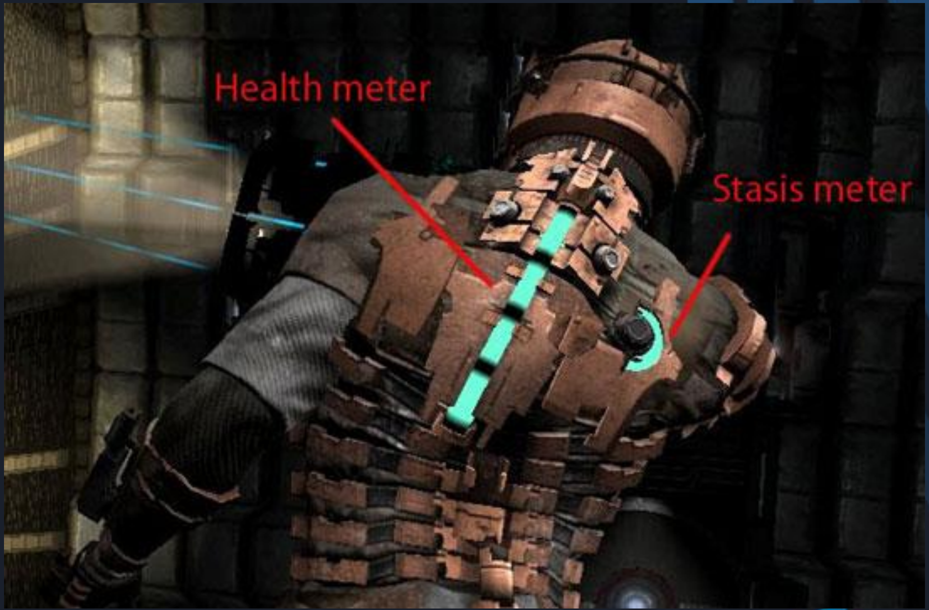
GUI Design

- It is often useful in complex games to allow the user to modify the GUI to their liking
- This is an example of the "vanilla" World of Warcraft GUI, vs an advanced player's customised version
  - You never want to introduce your players to a game with a GUI like that!

GUI Design

- Some games attempt to incorporate the GUI into the game world, rather than as a 2-D overlay
  - Dead Space is a prime example
- Games take abstract items, such as Health, and incorporate it onto the character or into the world in some way
  - A character with more cuts and bruises
  - The world appearing darker as you're about to be consumed by evil
- Other more tangible items as well
  - Ammo display on a gun, rather than on a HUD



GUI Design

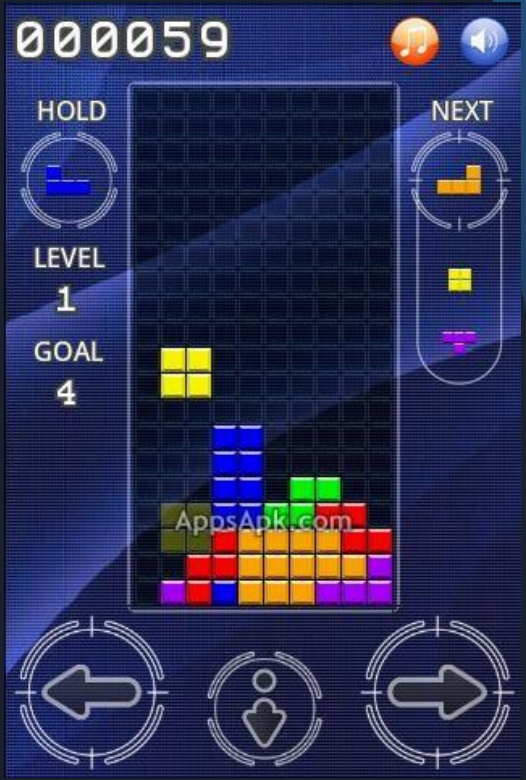
- It can be difficult to create a “perfect” GUI

- Questions to ask yourself
  - Who is my target audience and what do they like?
  - How does my target audience play this kind of game?
  - What would let them play better?
  - What frustrates them the most about interacting with this type of game?
  - What do they enjoy the most about interacting with this type of game?

Genre Specific GUI Design

- Casual 2-D Games
  - Simple or no UI
  - Generally made large and inviting
  - Generally requires no interaction from user
    - Purely there to display information

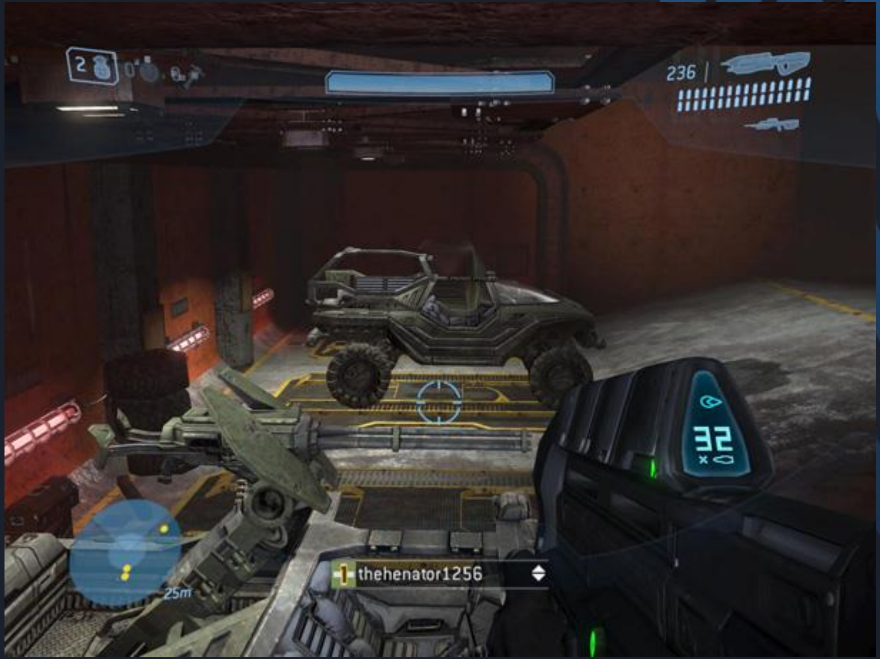
- The example on the right only allows the user to interact with 3-5 icon images, but the rest displays standard casual game information
  - Score, level, goal



Genre Specific GUI Design

- Shooters
  - Very clean
  - Generally only targeting reticule, health bar, ammo
  - Tries to be as immersive as possible

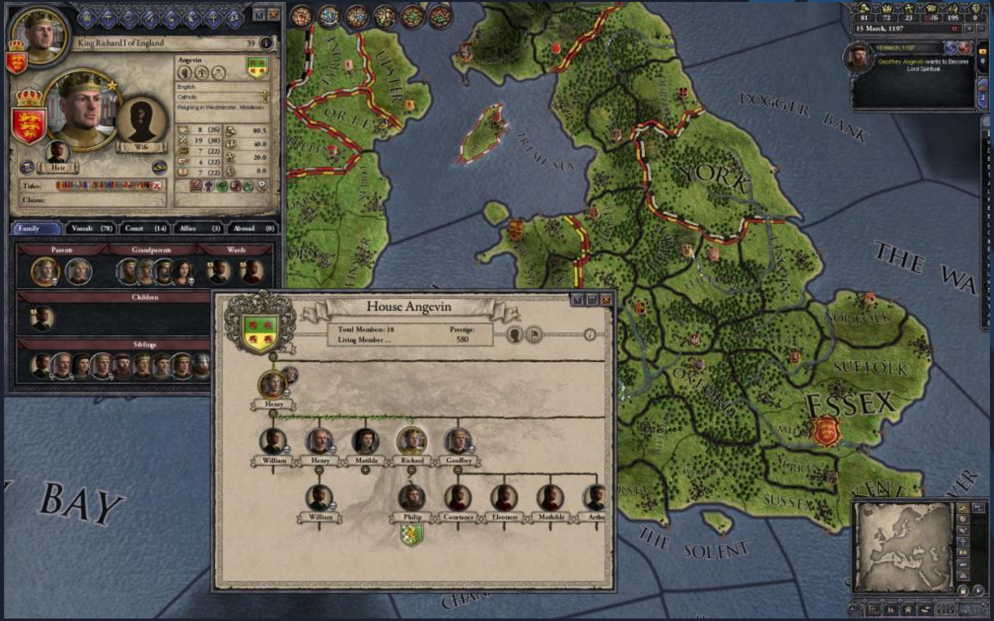
- Over the years more dynamic graphics features have allowed GUI elements to be moved into the game world
  - Ammo counters



Genre Specific GUI Design

- Strategy Games
  - Often complex GUI
  - Most of the game play is done through manipulating the GUI
  - Important that the GUI is clear to read, and taught to the player

- Certain games, like Strategy Games, attract players who like being able to go through lots of information about their carefully crafted game worlds
  - Care must be taken to ensure the menus and ways that the player can get this information is clean, quick and responsive



GUI Design

- The most important aspect of GUI design is communicating the correct information to the player
  - When designing your GUI, this should be your first consideration

- Test your GUI early and often, and make sure that people completely new to the game are able to navigate and understand your GUI

GUI Implementation

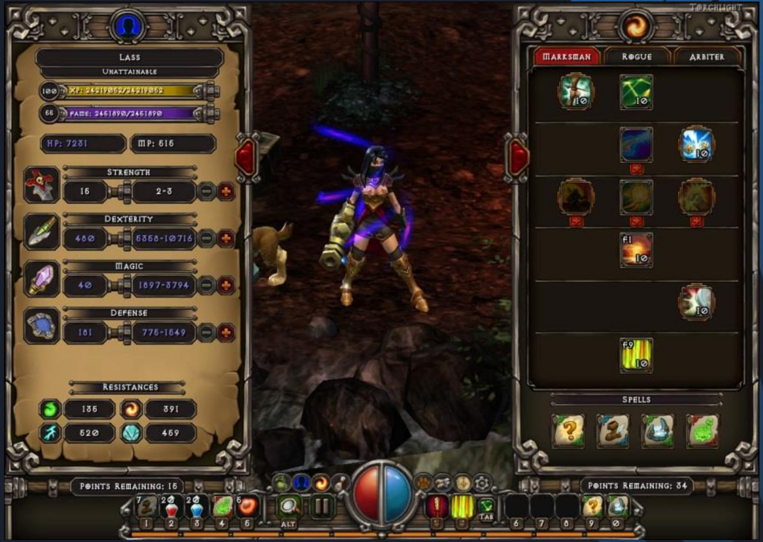
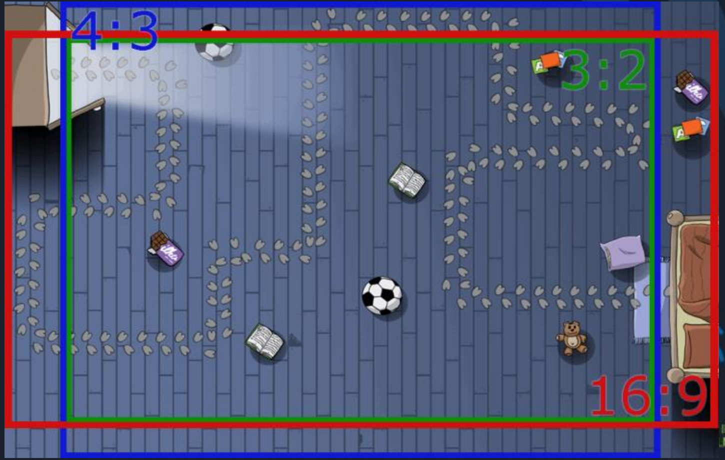
- Implementing a GUI can be as simple or complex as needs be

- Considerations must be made for
  - What information is to be displayed
  - How does a player interact with the information
  - What platforms will this game be played on
  - How do similar games display their information
  - Will the GUI be a 2D HUD or in-world displayed

2-D Rendering

- Rendering 2-D HUD images usually consists of
  - Render a textured quad or a polygon of multiple triangles
  - Use an orthographic projection of the same width and height of the screen, allow exact pixel positions

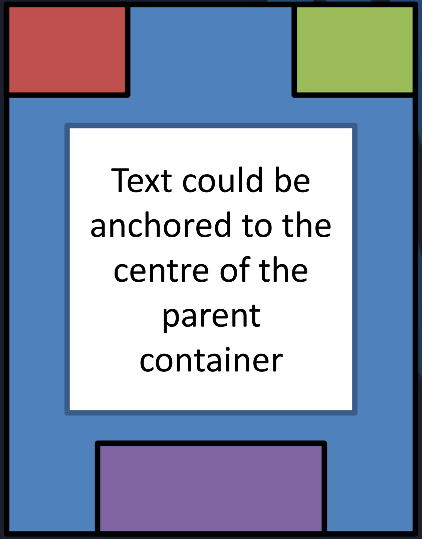
- In most cases HUD elements are designed to fit a certain aspect ratio rather than a set resolution
  - Positions and dimensions are treated as a scale based off the screen resolution
  - Textures need to be designed for the correct aspect ratio
  - Many GUIs have a safe zone, which means elements can't be drawn close to the edge of the screen to account for slight inconsistencies between monitors and screens

2-D Rendering

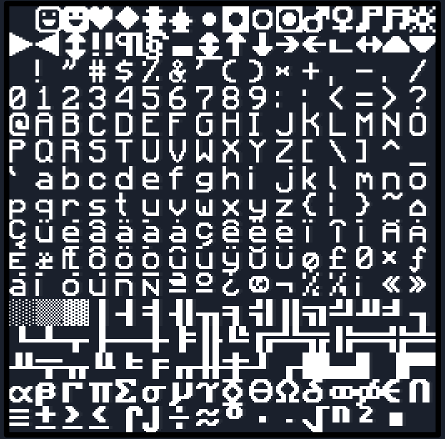
- A hierarchy structure is typically used to allow grouping of elements
  - A dialog box may group some text and buttons, in addition to border images
  - Transforms can propagate through the hierarchy allowing the scaling parent containers that also scale their child elements

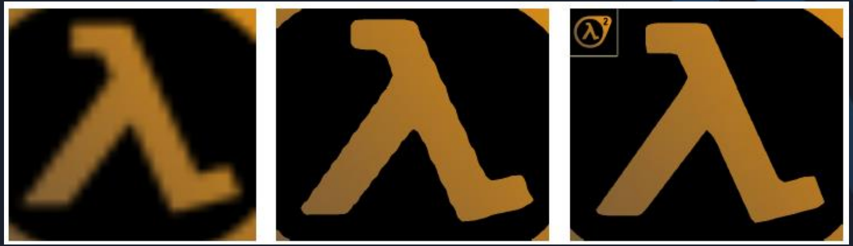
- Elements can be attached to each other
  - Usually allow for anchor points
    - The Top-Left of a container element could be set as the anchor point for the Top-Left of a "back" button
    - The Top-Right of the same container element could be set as the anchor point for the Top-Right of a "next" button
    - The Bottom-Middle could be set to an "accept" button
  - The anchor points allow the elements to use that as their (0,0) position so that they move and scale correctly with each other


Text could be anchored to the centre of the parent container

2-D Rendering Text

- Text can be rendered in multiple ways
  - A sprite sheet of letters can be used, with the text rendered as many 2-D quads, one for each letter in the text
  - Text could be rendered as vector-based graphics
    - Letters consist of many triangles, with the vertices making up the edge of a letter
  - Newer methods use Distance Fields
    - Similar to sprite sheets, except they store distance of each texel to the edge of the text and create smooth edges during rendering at any resolution





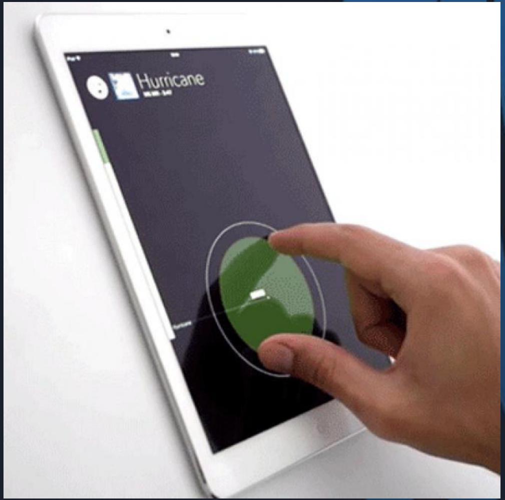
Alpha-blended Sprite

Alpha-tested Sprite

Distance Field

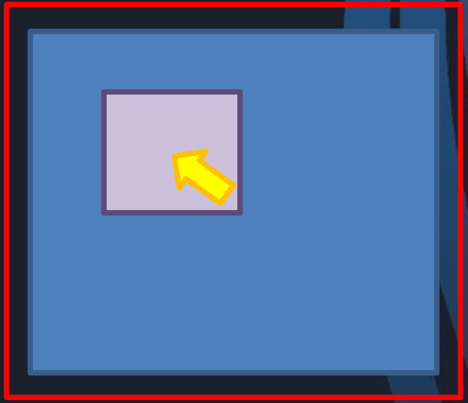
Interacting with a 2-D GUI

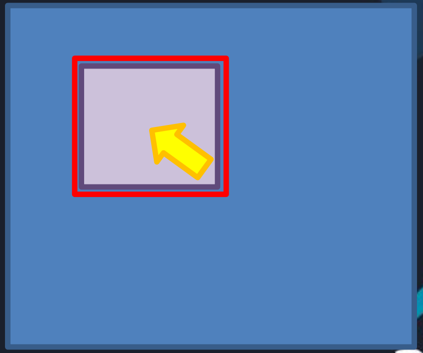
- Interacting with a HUD via mouse or touch clicks can involve
  - Testing if the cursor location is within the dimensions of the root element of the HUD
    - If not, then the touch event can be discarded
    - If it is then test that element's child elements until the top level HUD element has been selected, which can handle the touch event



Interacting with a 2-D GUI

- Interacting with a HUD via mouse or touch clicks can involve
  - Testing if the cursor location is within the dimensions of the root element of the HUD
    - If not, then the touch event can be discarded
    - If it is then test that element's child elements until the top level HUD element has been selected, which can handle the touch event

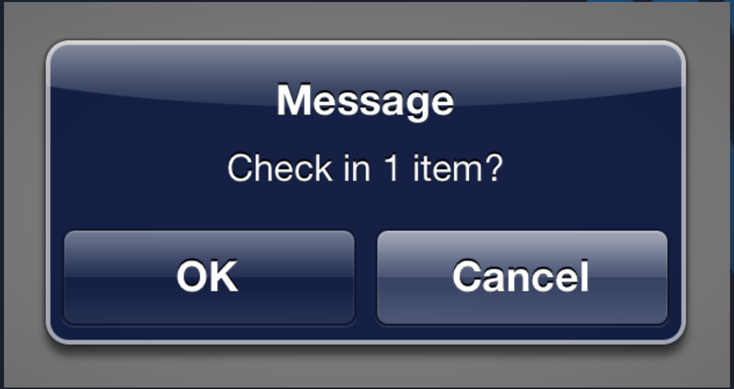




Interacting with a 2-D GUI

- Typically mouse-over events can occur
  - Menu icons highlighting when the cursor hovers over them

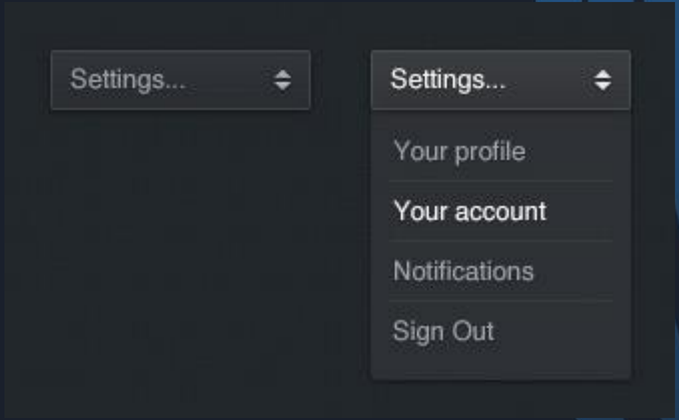
- These events can either be:
  - The input system notifies the UI of changes, which then determines if the mouse has entered / left an element
  - The elements can test for the cursor location during their own update, changing their state if the cursor has entered / left their area

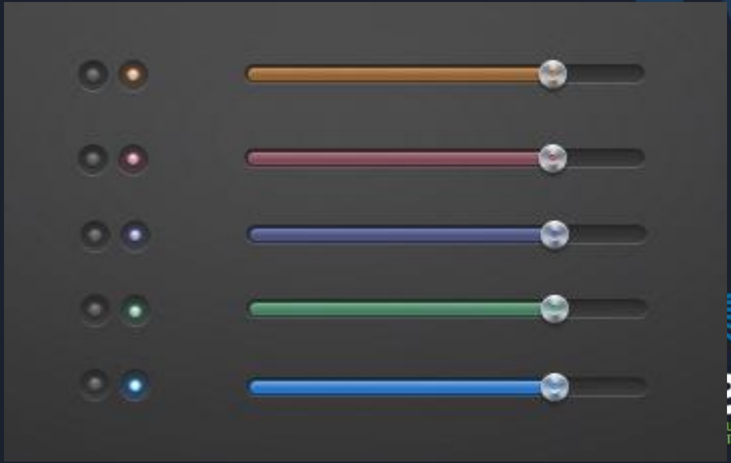


Interacting with a 2-D GUI

- More complex interact able elements exist
  - Sliders
  - Drop-down menus

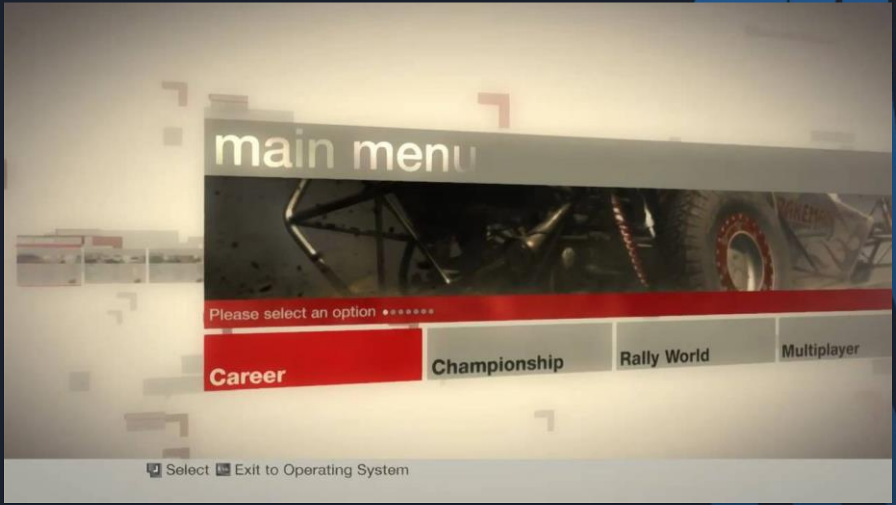
- These can use a complex combination of events
  - i.e. in the case of a slider it needs to update its position on mouse-move events if the cursor is over it and the mouse button has been held
  - The slider button would move but is confined to the area of its parent element, the slider bar

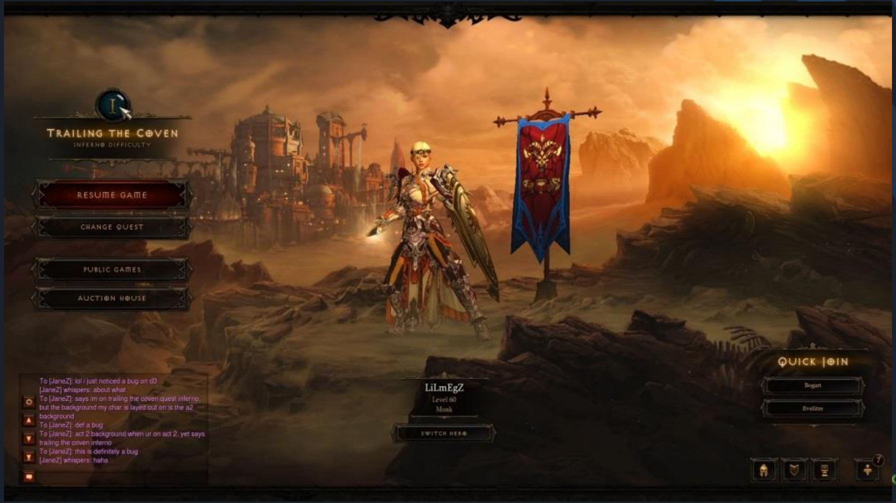




Dynamic GUI

- During a menu not much is typically happening
  - Spare processing power to make it fancy!
- Many games now have highly dynamic menus
  - Collin McRae Dirt for example
- Many incorporate a 3-D scene
  - Diablo 3 for example

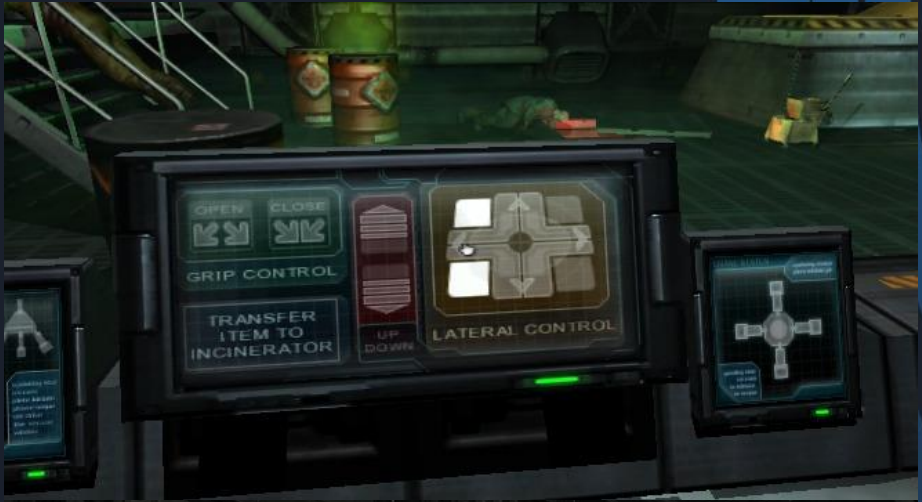


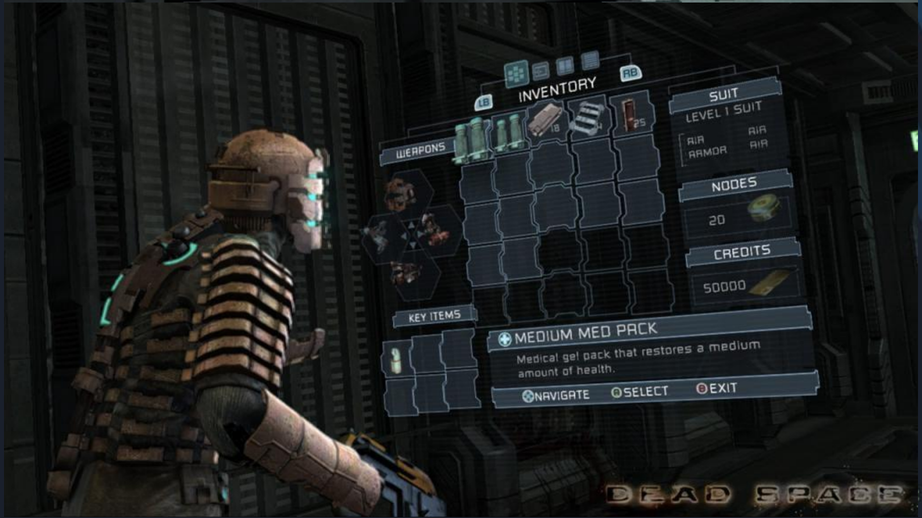


To [JaneZ]: lol i just noticed a bug on d3
[JaneZ] whispers: about what
To [JaneZ]: says im on trailing the coven quest inferno, but the background my char is layed out on is the a2 background
To [JaneZ]: def a bug
To [JaneZ]: act 2 background when ur on act 2, yet says trailing the coven inferno
To [JaneZ]: this is definitely a bug
[JaneZ] whispers: haha

In-World Display

- In-world displays can be simple
  - Health bar as part of the character
    - Masked areas of the model's texture can be set to different colours
- And in-world displays can be complex
  - Menu rendered to a texture which is then applied to an in-world mesh
    - Doom 3's interact able computer screens
    - Dead Spaces holographic menus
  - Input must be transformed into the coordinate space of the surface

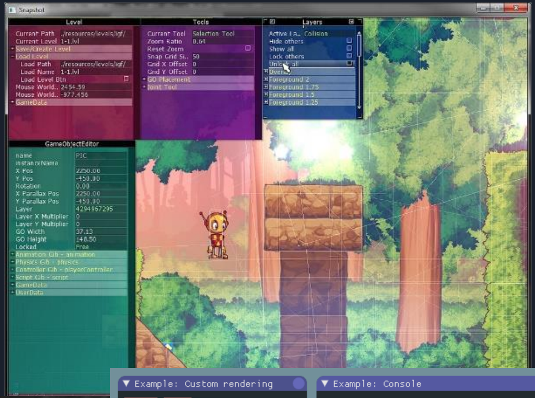


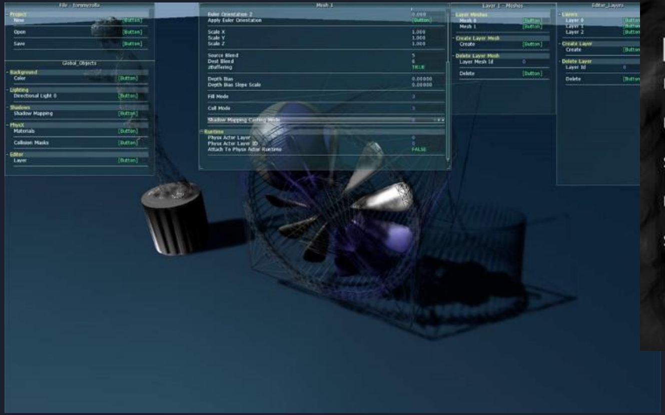


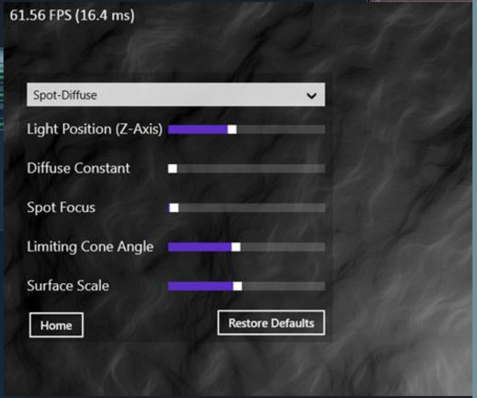
Development GUI Tools

- GUIs aren't just for in-game information

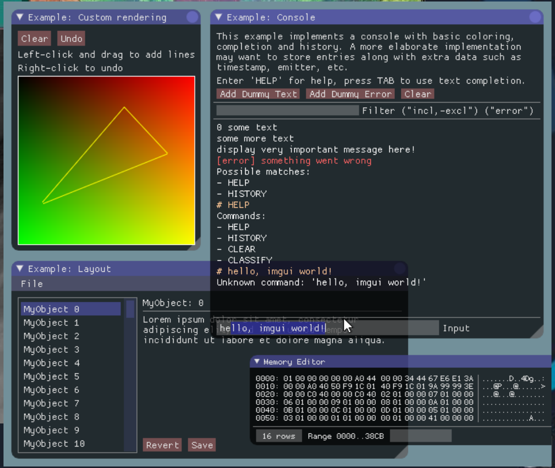
- During development, developers can use GUIs that allow them to tweak game values and properties while the game is running







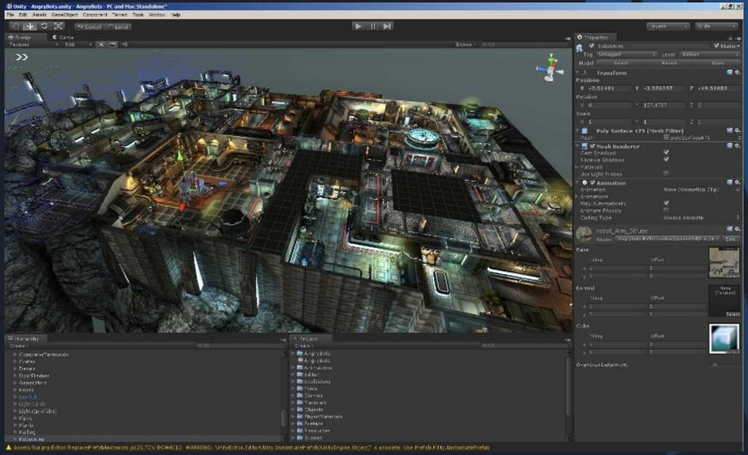
```
61.56 FPS (16.4 ms)
Spot-Diffuse
Light Position (Z-Axis)
Diffuse Constant
Spot Focus
Limiting Cone Angle
Surface Scale
```

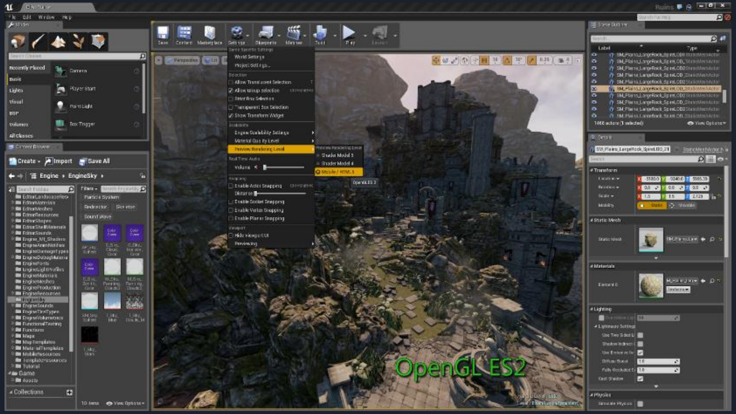


```
▼ Example: Custom rendering
Clear  Undo
Left-click and drag to add lines
Right-click to undo

▼ Example: Console
This example implements a console with basic coloring,
completion and history. A more elaborate implementation
may want to store entries along with extra data such as
timestamp, emitter, etc.
Enter 'HELP' for help, press TAB to use text completion.
Add Dummy Text  Add Dummy Error  Clear
                         Filter ("incl,-excl") ("error")
0 some text
some more text
display very important message here!
[error] something went wrong
Possible matches:
- HELP
- HISTORY
# HELP
Commands:
- HELP
- HISTORY
- CLEAR
- CLASSIFY
# hello, imgui world!
Unknown command: 'hello, imgui world!'
hello, imgui world!                    Input

▼ Example: Layout
File
MyObject 0     MyObject: 0
MyObject 1     Lorem ipsum
MyObject 2     adipiscing e
MyObject 3     incididunt ut labore et dolore magna aliqua.
MyObject 4
MyObject 5
MyObject 6
MyObject 7
MyObject 8
MyObject 9
MyObject 10    Revert  Save

▼ Memory Editor
0000: 01 00 00 00 00 00 A0 44  00 00 34 44 67 E6 E1 3A  ......D..4Dg..:
0010: 00 00 A0 40 50 F9 1C 01  40 F9 1C 01 9A 99 99 3E  ...@P...@......>
0020: 00 00 C0 40 00 00 C0 40  02 01 00 00 07 01 00 00  ...@...@........
0030: 06 01 00 00 09 01 00 00  08 01 00 00 0A 01 00 00  ................
0040: 0B 01 00 00 0C 01 00 00  0D 01 00 00 05 01 00 00  ................
0050: 03 01 00 00 01 01 00 00  00 01 00 00 41 00 00 00  ............A...
16 rows  Range 0000..38CB
```

Development GUI Tools

- Entire game engines can be built around GUI editors
  - Allows artists and designers to easily modify the game without needing a programmer's assistance





Summary

- The HUD should only hold as much information as is needed for the game to be playable
  - Display as much information in-world as you can
- Always aim for user accessibility as your first priority when designing a GUI
  - If your target audience can’t understand it then they’ll just play what they do understand
- Think about the design considerations when implementing UI systems

Further Reading

- Gregory, J, 2014, *Game Engine Architecture*, 2nd Edition, CRC Press

- Graham, D, McShaffry, M, 2013, *Game Coding Complete*, 4th Edition, Cengage Learning