
HCI + UI = UX

CS 4730 – Computer Game Design



Computer Science
at the UNIVERSITY of VIRGINIA

CS 4730

How do we interact with the game?

- Broken into two parts
 - 1. a device that can sense a player's physical motion (we're assuming we're not using mental devices just yet...)
 - 2. how the game interprets the movement into the virtual world
- Part 2 is related to mechanics/dynamics
- Part 1 can be decoupled from games a bit and discussed independently

Categories of Devices

- Touch-based
- Inertial-based
- Sound-based
- Camera-based
- “Advanced sensing”
- Each of these has subcategories as well



Touch-Based

- This category involves all devices that require physical contact to sense user input
- By far the largest/most common category
- Most are all electric circuit based, wherein a player completes a circuit by a motion, thus triggering a signal to the system



Touch-Based

- Binary Circuits
 - Button unpressed = low voltage
 - Button pressed = high voltage
- Angular Sensors (potentiometer)
 - Analog joysticks vary the voltage with position
- Electromagnetic Field
 - Surface with baseline electromagnetic state is disrupted by a finger touch
 - Resistive completes a circuit beneath

Controller Examples

- Keyboard
- Mouse
- Console Controller
- Specialized Console Controller



WASD and the Mouse

- The keyboard has been around for a LONG time (relatively speaking)
 - Devised in the early 1900s
 - With computing devices in the 60's
- The mouse... not as long, but still for a while
 - 1984
 - Xerox PARC and Apple and Microsoft

WASD and the Mouse

- What sort of games make sense for this control structure?
- Why?
- How does that affect the game play?



Evolution of the Console Controller



Inertial-Based

- Accelerometer
 - Senses the direction of acceleration and gravitational force on a single axis
- Magnetometer
 - Effectively a compass sensing the Earth's magnetic field
- Gyroscope
 - Senses angular velocity and change in force on a spinning mass



Accelerometers and the Wiimote

- Wiimote uses a 3-axis combo accelerometer
- Wii Motion Plus adds a gyroscope



Accelerometers and the iPhone

- How is this a different dynamic?



GPS as Input Device

- Seek ‘n Spell



Handling Noisy Input

- Uncertain input can make games frustrating
- Player intention has to match game interpretation
- This breaks down into a *state estimation problem*
- By increasing or decreasing sampling rate, we can have a better prediction as to what state the device is in



Light Guns

- Flashing on the screen is interpreted by the device
- A ray can be traced from the gun aperture to the screen to determine the angle of attack



Sound-Based

- Speech Commands
- Calculate the Euclidian distance from a normalized sound clip to training clips stored in a database
- Additional training models can be layered on top of this
- Or you could ignore all of that and just go for pitch (i.e. Rock Band)



Camera-Based

- Blobfinding – latching on to specific colored things in the environment
- Background subtraction – over n frames, what hasn't changed? That's the background
- And... let's just look at the Kinect and Child of Eden



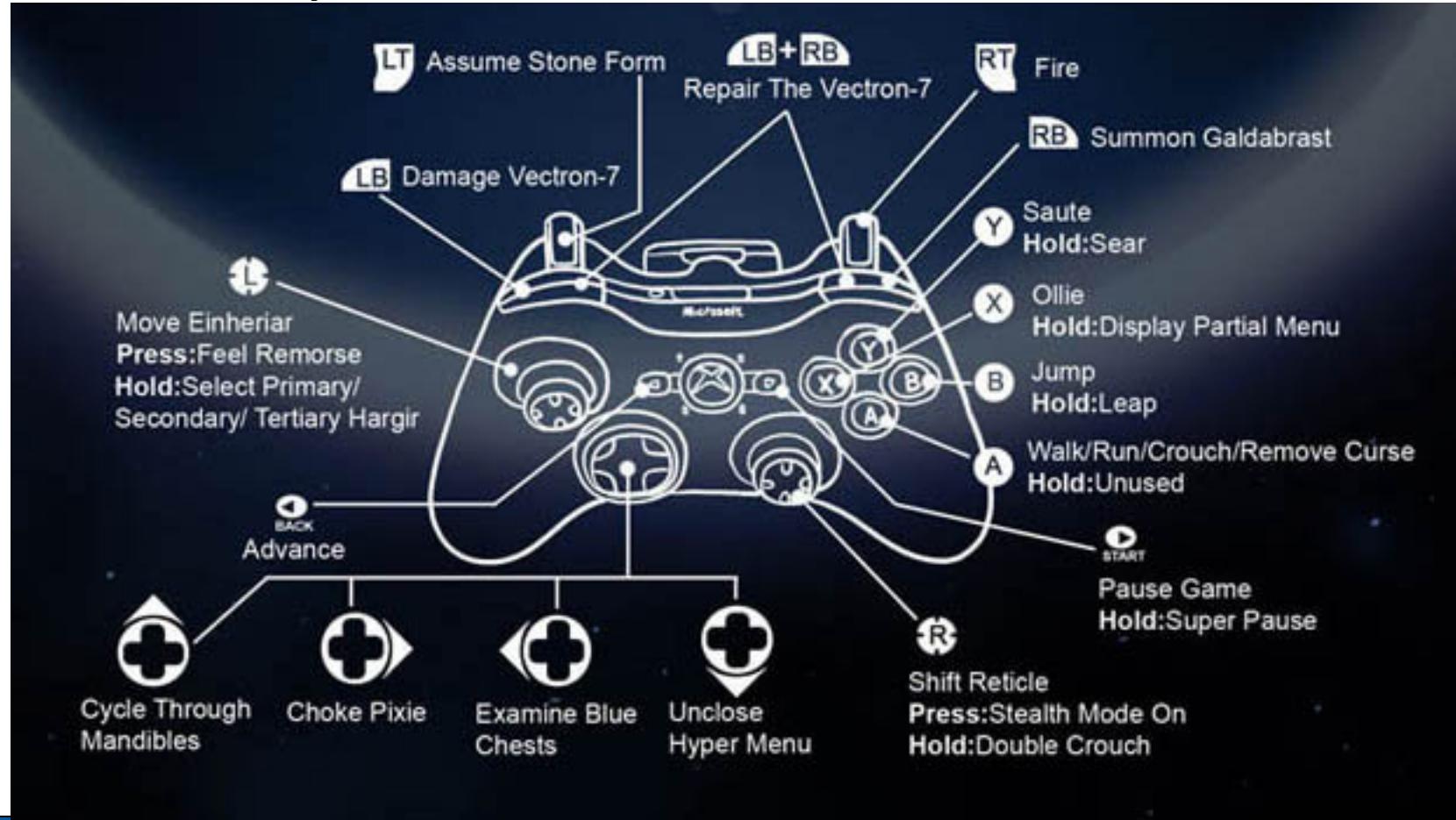
Input Schemes

- Step 1: Which controller
- Step 2: What buttons do what?
- Step 3: ????
- Step 4: Profit!



Input Schemes

- How do you know what button does what?



Input Schemes

- The bane of several genres
 - Sports
 - FPSs
- The bane of touch screen games
 - Virtual sticks
 - Touch-to-move
- The bane of keyboard games
 - ... that's a lot of buttons...



“Accepted” Input Schemes

- How are these communicated to users?
- How do we get over the problem of “training” new gamers?

Okay... so that's control

- Now how do we relay info back to the user on screen
- The User Interface is essential to the feedback construct of games



Let's Take This Out of Video

- How do you consider user interface for non-digital games?
- What provides feedback?
- How is it enforced?



How “Easy” is “Easy”?

- Ease-of-use vs. Ease-of-learning
- Ease-of-use
 - If you already know what you want to do, how quick / easy is it to pull it off?
- Ease-of-learning
 - If you are new to a game, how easy is it to figure out what you are allowed to do and how to do it?
- Usually, there is a linear trade off between these two concepts

How “Easy” is “Easy”?

- Consider any WASD game
 - Tons of hot keys
 - ... tons of keys period
- Very quick to do many different actions
- How do you know which button to press?



How “Easy” is “Easy”?

- Consider a complex board game
 - Usually, there are tons of charts, symbols, icons, etc.
 - Very easy for an experienced player to know exactly what's going on
 - Intimidating to novice players



How “Easy” is “Easy”?

- Sometimes you can do both
- The concept here is called *recognition over recall*



How “Easy” is “Easy”?

- Sometimes you can do both
- The concept here is called *recognition over recall*



Super Models

- The User Model
 - How the user “thinks” the system will behave
- The Program Model
 - How the game ACTUALLY will behave
- Program Model is always “right” with video games
 - This isn’t always true with board games!



Super Models

- You want these models to match as much as possible
 - Even if there is some “behind the scenes” shenanigans going on to make it look right
- Which is easier to change?
- How do you know if you need to change something?



Super Models

- Play testing!
- This is actually much more robust than in other software development
- You watch people play
 - See what buttons they press
 - When they press
 - How they press
 - How they interact
 - Watch lots of people!

If the Models are Wrong

- Frustration, embarrassment, anger are all reactions to an incorrect user model
- Probably not a central aesthetic to your game
 - (maybe...)



Focus on the Interactions!

- What is the *operational profile* of your game?
- Find the most common activities and make them the easiest to do
- Find the most common icons/symbols and put them up-front and center
- Don't rely on traditional colors/symbols for everything
 - Consider the mailbox icon...

Tips and Tricks

- Eliminate all uninteresting tasks – everything should have a purpose
- Use visual metaphors – make it obvious for what something represents
- Be consistent with similar/established games
- Keep HUDs appropriate to the aesthetics



How important is UI design?

- If you screw up your UI for a text editor
 - The user might get a little frustrated
 - Might read the manual
 - Might realize they are using EMACS and quit
- If you screw up your UI for a game
 - The user gets very frustrated
 - The user may get angry
 - The user will never buy your game again
 - The user will leave lots of nasty comments

What are you trying to convey?

- What are you trying to get across to the user with your UI design?

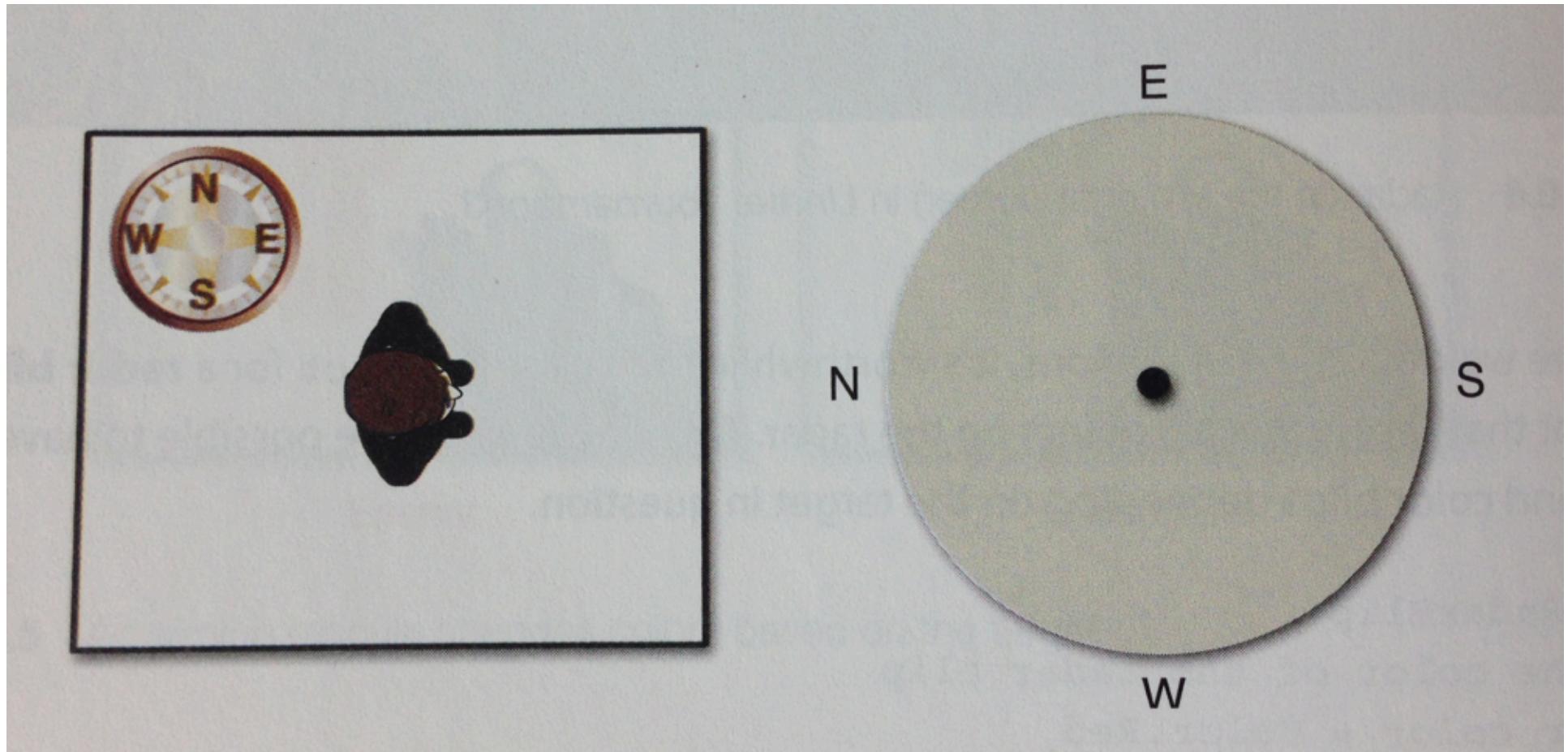


What are you trying to convey?

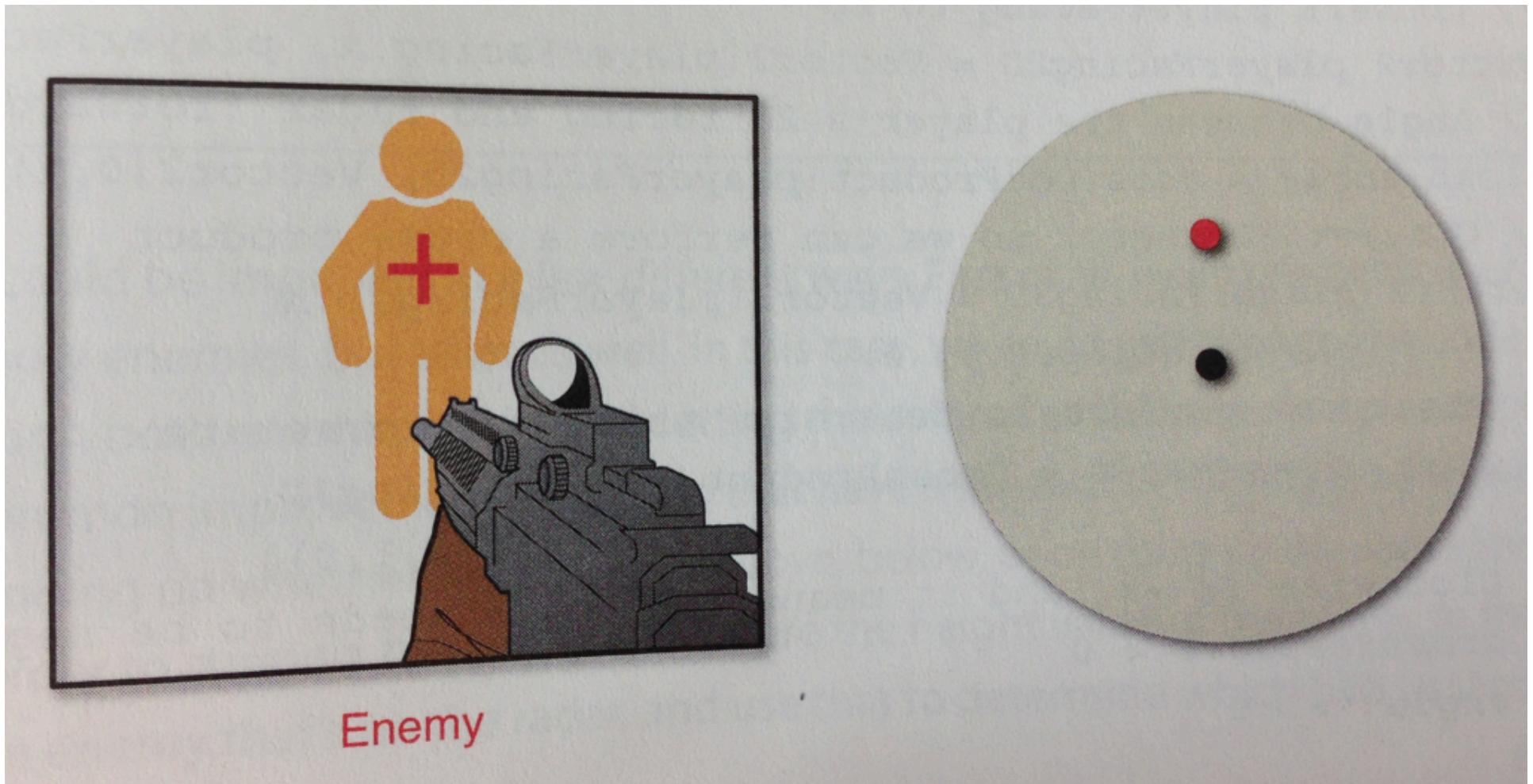
- The state of the world
- The rules
- The available actions/interactions
- How do you do this?
- How do you show the state while keeping gamers immersed?



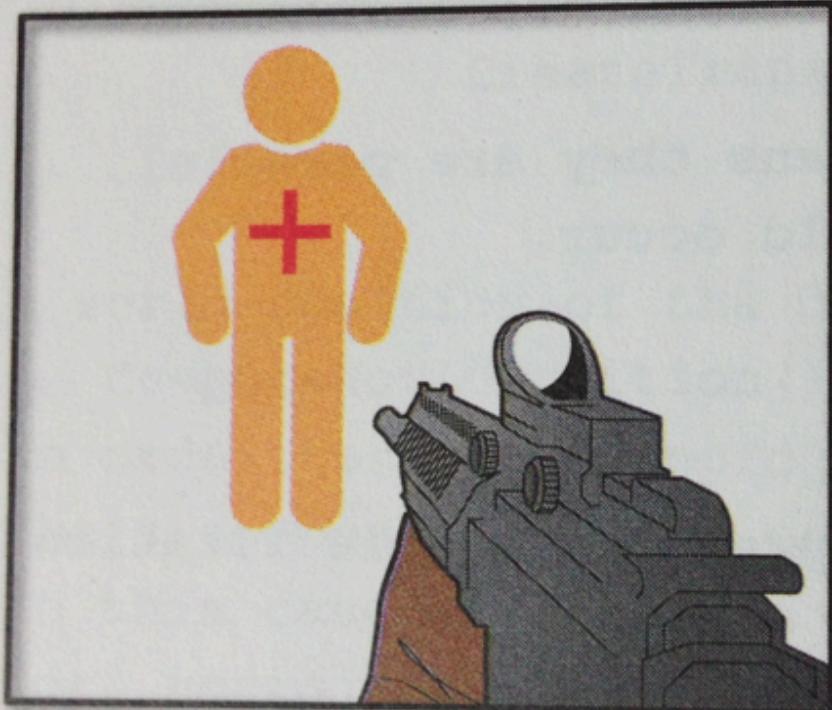
Providing Info



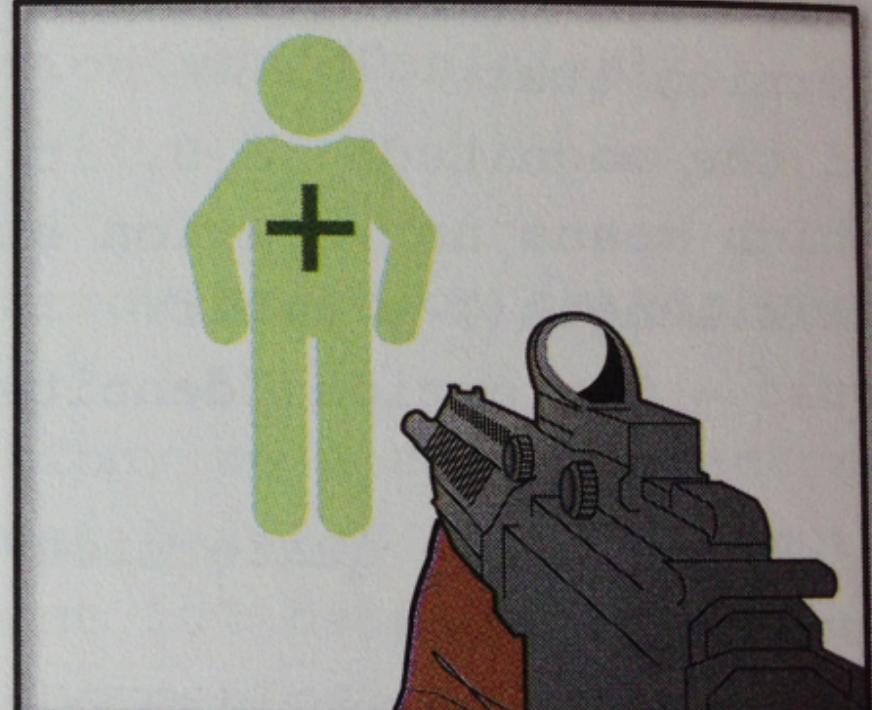
Providing Info



Providing Info



Enemy



Friendly



Examples



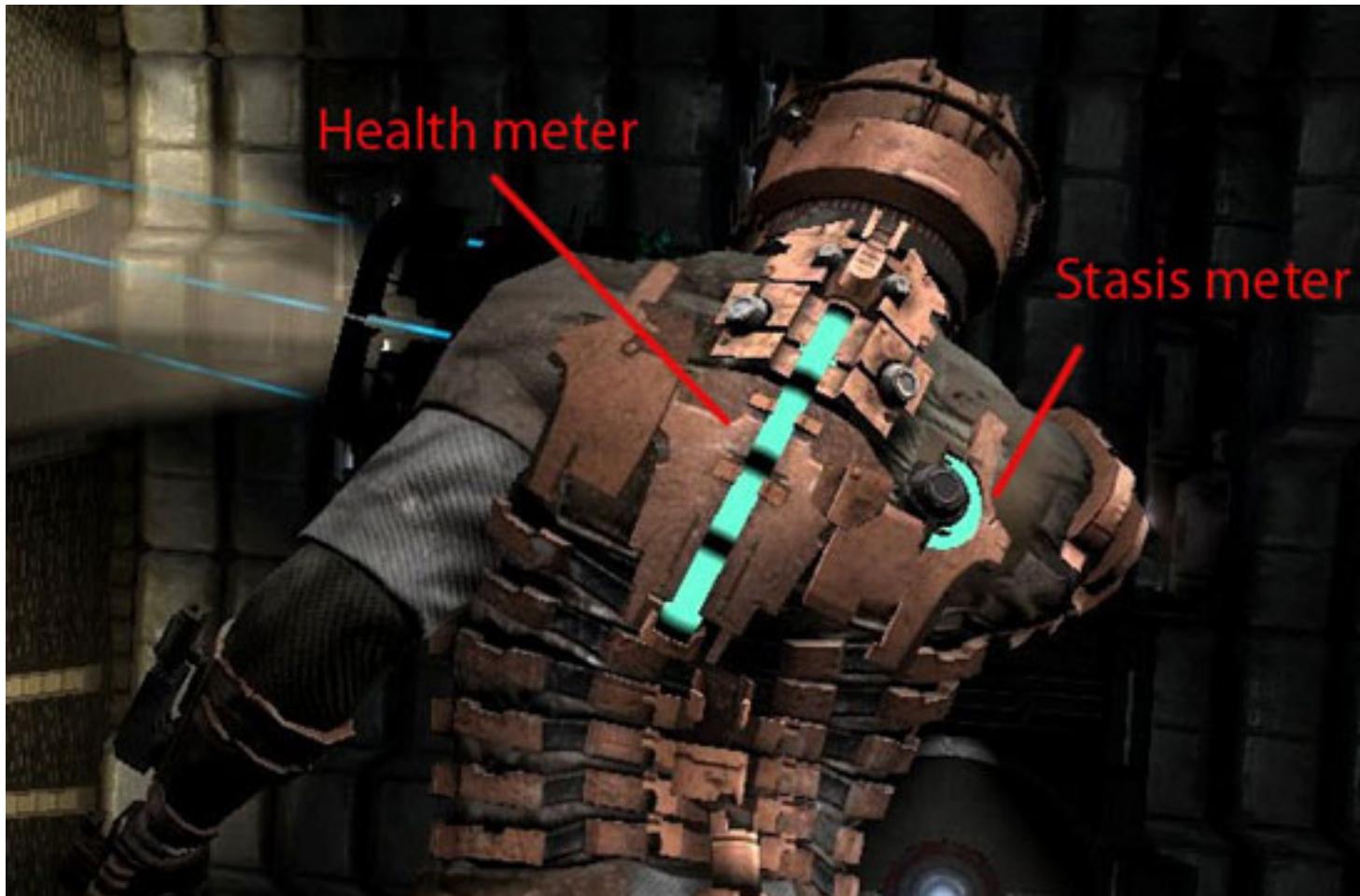
Examples







Examples



Examples



Examples

