INFT3960 – Game Production

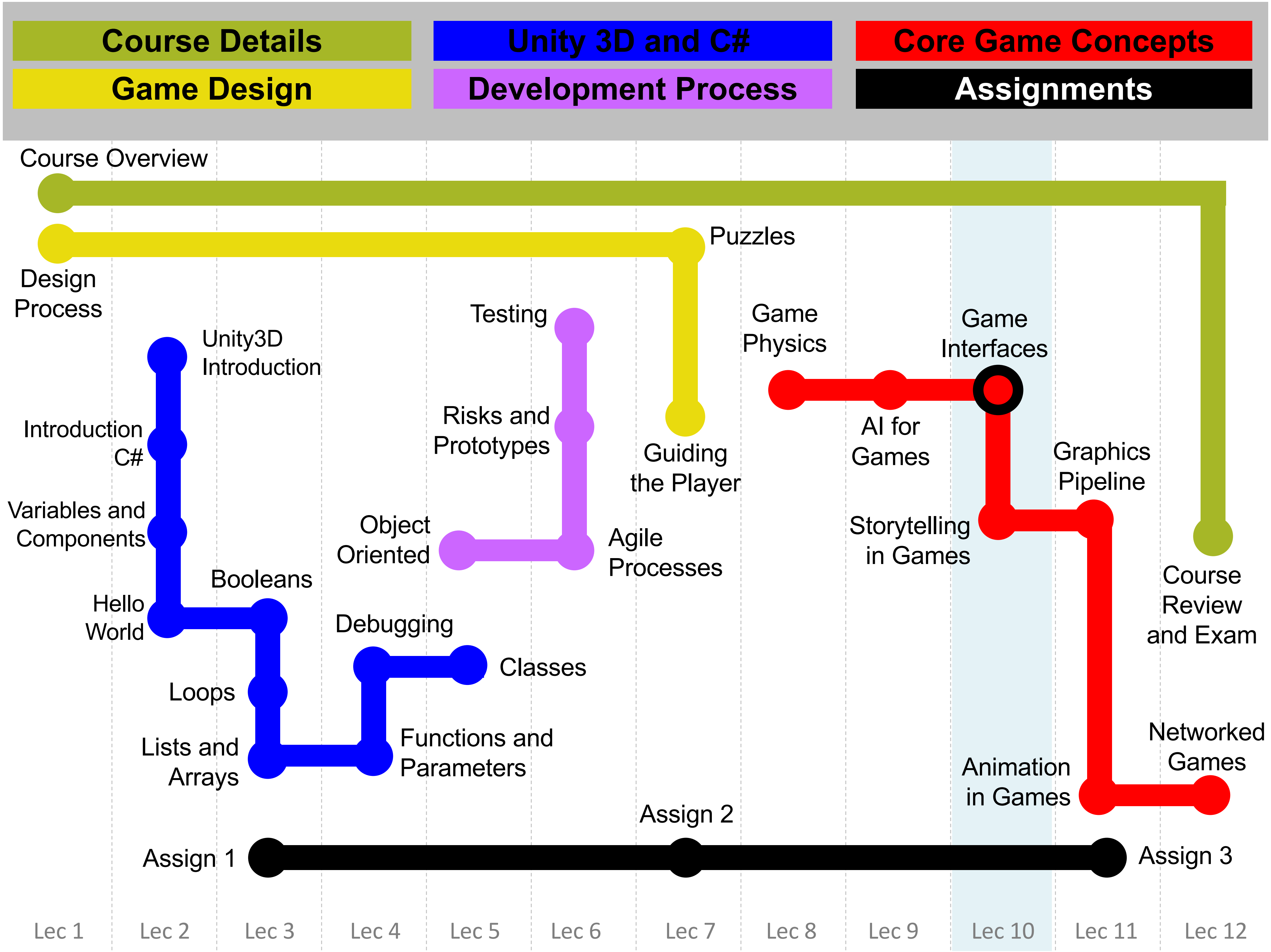
Week 10

Module 10.1

Game Interfaces

Course Overview

Lec	Start Week	Modules	Topics	Assignments
1	3 Aug	Mod 1.1, 1.2	Course Overview, Design Process	
2	10 Aug	Mod 2.1, 2.2, 2.3, 2.4	Unity3D Introduction, Introduction C#, Variables and Components, Hello World	
3	17 Aug	Mod 3.1, 3.2, 3.3	Booleans, Loops, Lists and Arrays	Assign 1 21 Aug, 11:00 pm
4	24 Aug	Mod 4.1, 4.2	Functions and Parameters, Debugging	
5	31 Aug	Mod 5.1, 5.2	Classes, Object Oriented	
6	7 Sep	Mod 6.1, 6.2, 6.3	Agile Processes, Risks and Prototypes, Testing	
7	14 Sep	Mod 7.1, 7.2	Puzzles, Guiding the Player	Assign 2 18 Sep, 11:00 pm
8	21 Sep	Mod 8.1	Game Physics	
9	12 Sep	Mod 9.1	AI for Games	
10	19 Oct	Mod 10.1, 10.2	Game Interface, Storytelling in Games	
11	26 Oct	Mod 11.1, 11.2	Graphics Pipeline, Animation in Games	Assign 3 1 Nov, 11:00pm
12	2 Nov	Mod 12.1, 12.2	Networked Games, Course Review	



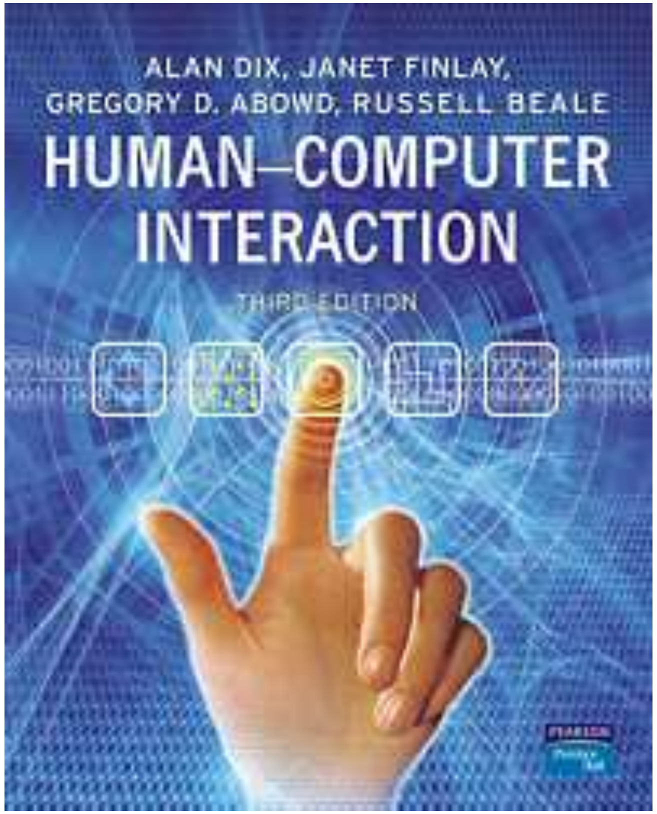
Course Details
Unity 3D and C#
Core Game Concepts
Game Design
Development Process
Assignments
Course Overview
Design Process
Puzzles
Guiding the Player
Unity3D Introduction
Introduction C#
Variables and Components
Hello World
Booleans
Loops
Lists and Arrays
Functions and Parameters
Debugging
Classes
Object Oriented
Testing
Risks and Prototypes
Agile Processes
Game Physics
AI for Games
Game Interfaces
Storytelling in Games
Graphics Pipeline
Animation in Games
Networked Games
Course Review and Exam
Assign 1
Assign 2
Assign 3
Lec 1
Lec 2
Lec 3
Lec 4
Lec 5
Lec 6
Lec 7
Lec 8
Lec 9
Lec 10
Lec 11
Lec 12

Game Interfaces – Topics

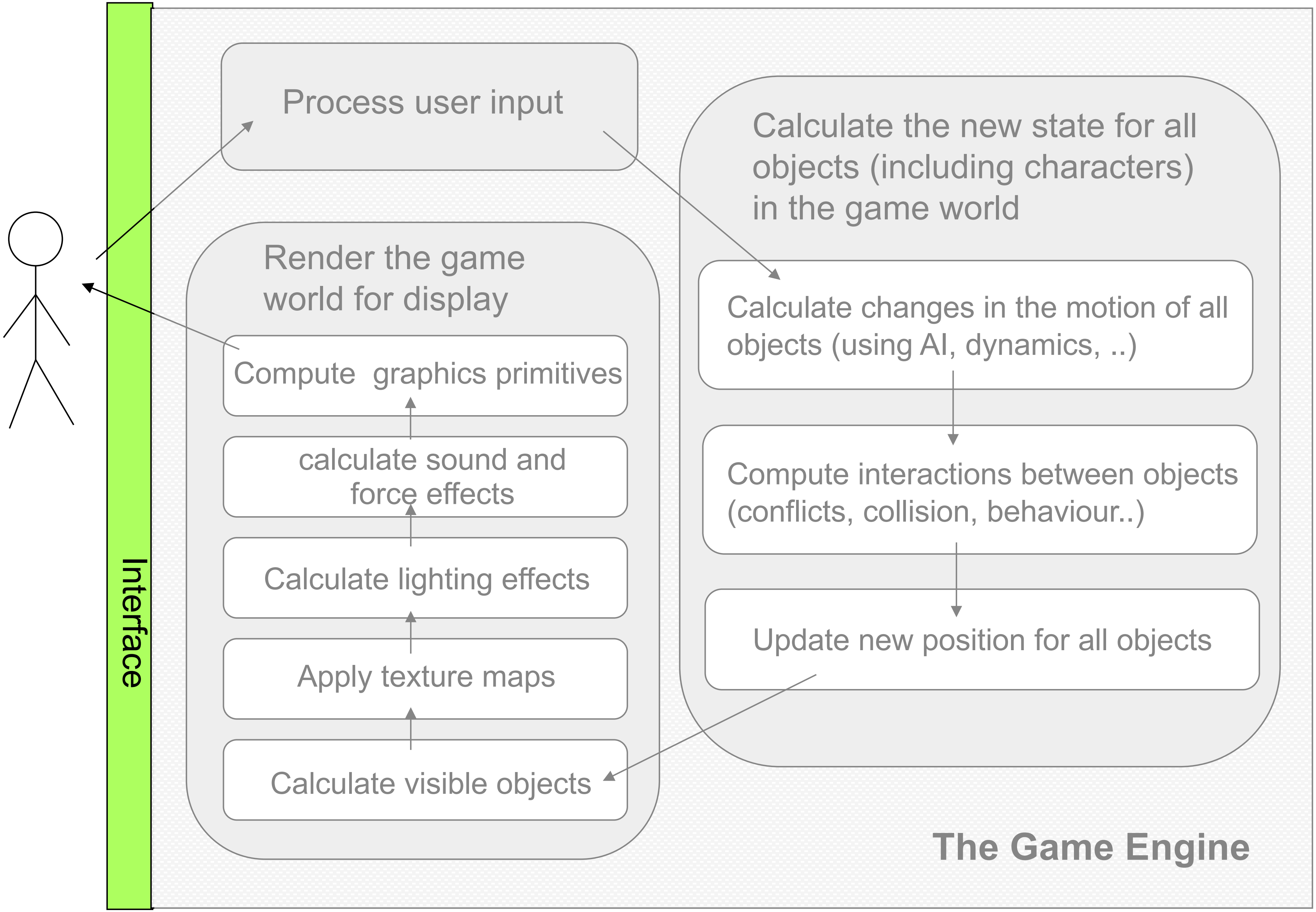
- User Interfaces
- Perception, Cognition, Action
- Design Principles

Useful References

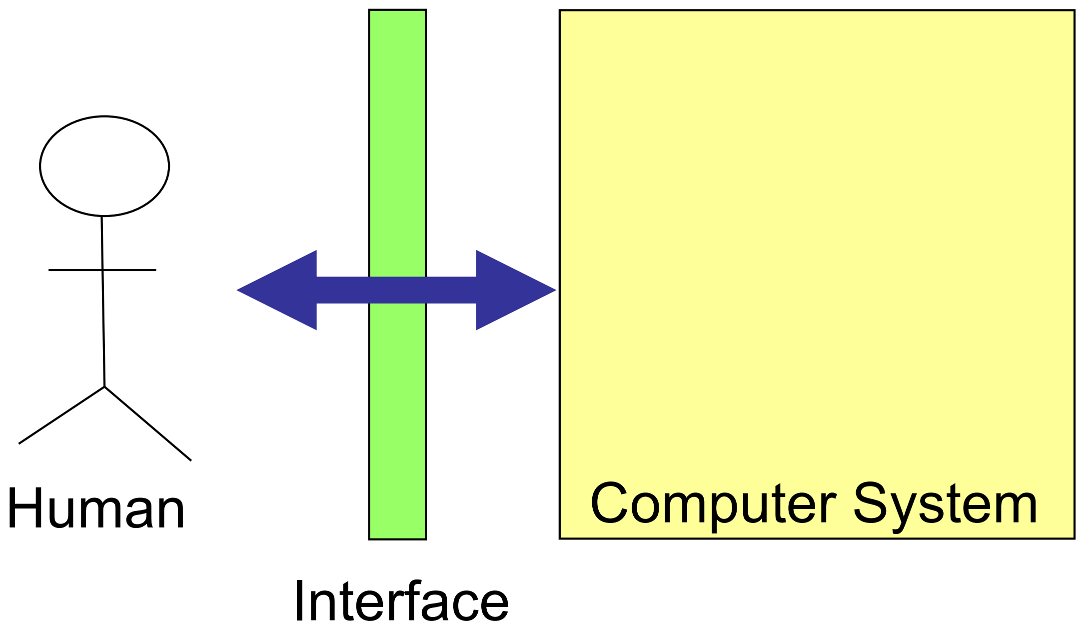
Dix, A., Finlay, J., Abowd G. Beale, R. (2004) Human Computer Interaction, Pearson.



Saunders, K. and Novak, J. (2007) Game Development Essentials: Game Interface Design. Delmar Cengage Learning.

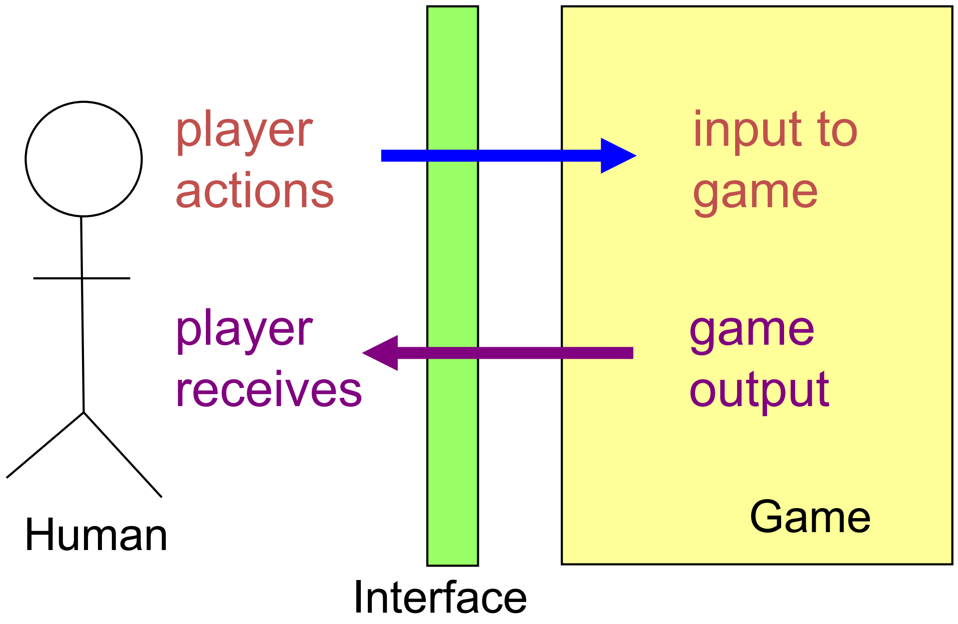

Interface
Process user input
Calculate the new state for all objects (including characters) in the game world
Render the game world for display
Compute graphics primitives
calculate sound and force effects
Calculate lighting effects
Apply texture maps
Calculate visible objects
Calculate changes in the motion of all objects (using AI, dynamics, ..)
Compute interactions between objects (conflicts, collision, behaviour..)
Update new position for all objects
The Game Engine

User Interfaces



The interface between computers and people (and more generally machines and people) has a long tradition of study. Fortunately, this work is also very relevant to understanding the interfaces between players and games.

Sensory Channels



All sensory channels can:

- Receive information from the computer (Game's output)
- Provide information to the computer (Player's input)

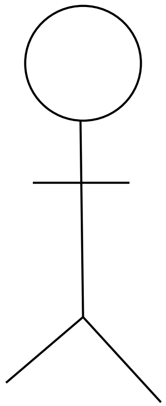
Input, Output

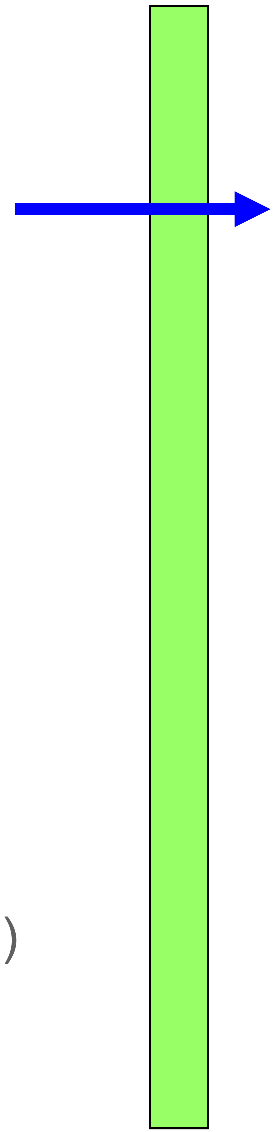
Some Input Modes

eye gaze direction
head position

sound
speech

keyboard
mouse (2D, 3D)
trackball
joystick
props (guitar, steering wheel, gun..)
gestures
body position





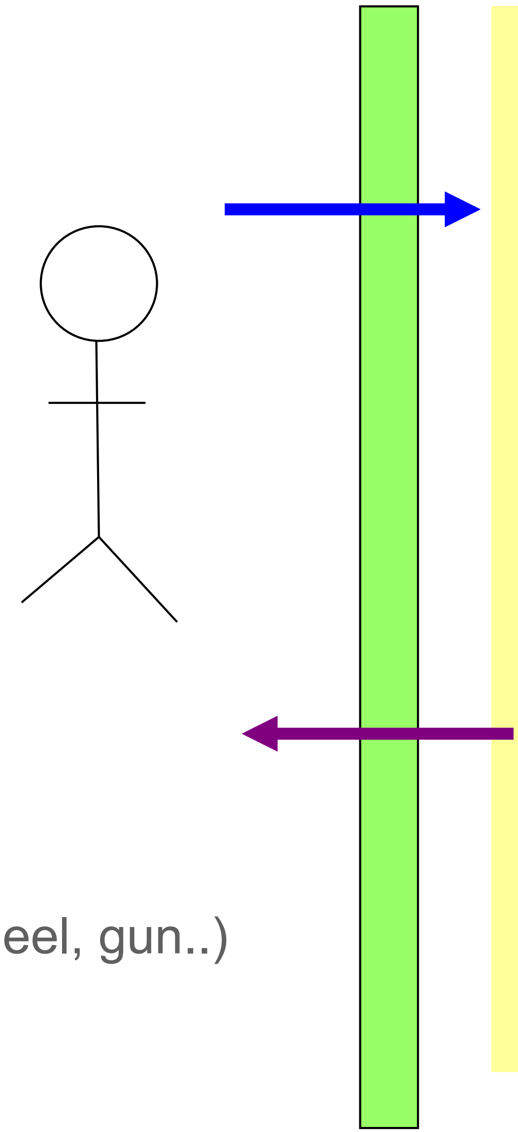
Input, Output

Some Input Modes

eye gaze direction
head position

sound
speech

keyboard
mouse (2D, 3D)
trackball
joystick
props (guitar, steering wheel, gun..)
gestures
body position



Some Output Modes

Text
Image
Motion
Position
Size

Colour
Shape
Lighting
Transparency
....

Speech
Pitch
Timbre
Loudness

Rhythm
Direction
Melody
....

Force
Vibration
Texture
Hardness

Movement
Temperature
Pain
....

Art or Science

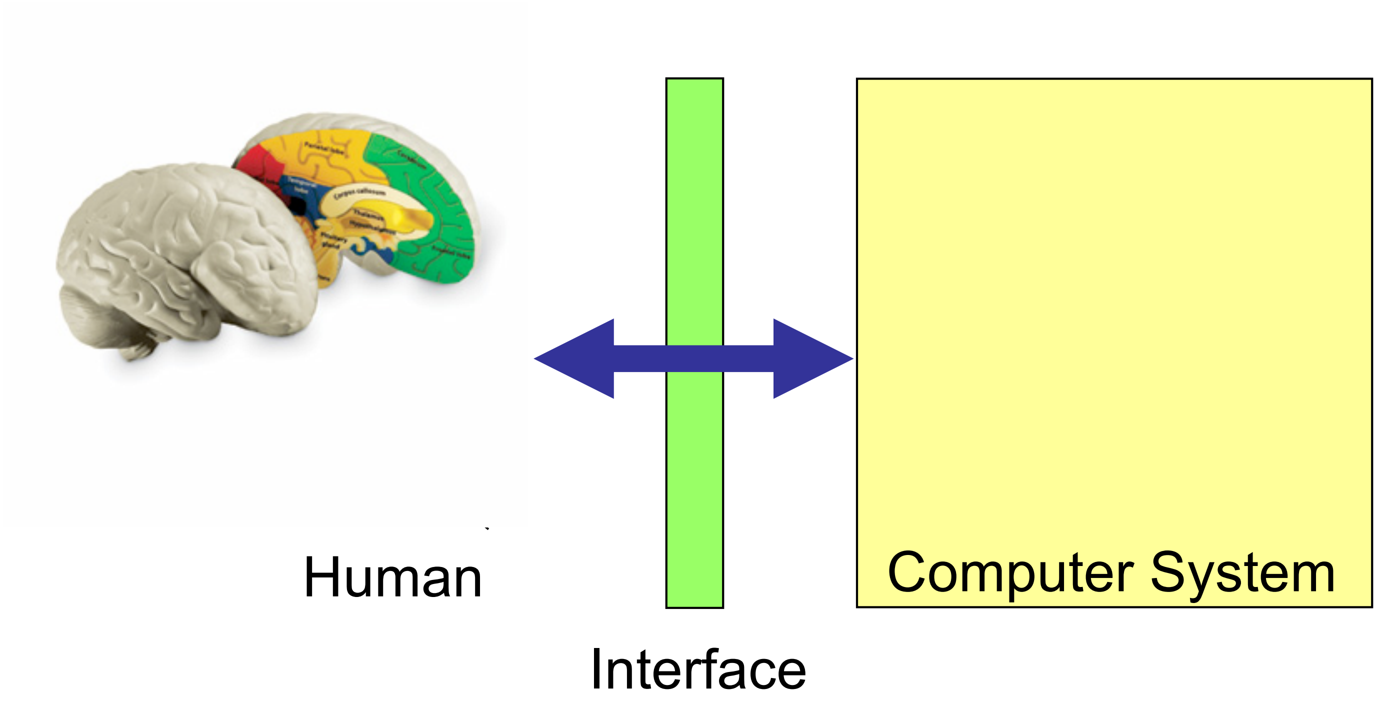
Some people will argue that good interface design, like other areas of design, may be more reliant on "art" than "science".

Probably both aspects are useful.

Art – intuitive design decisions (subjective)

Science – evidence-based design decisions (objective)

Art or Science

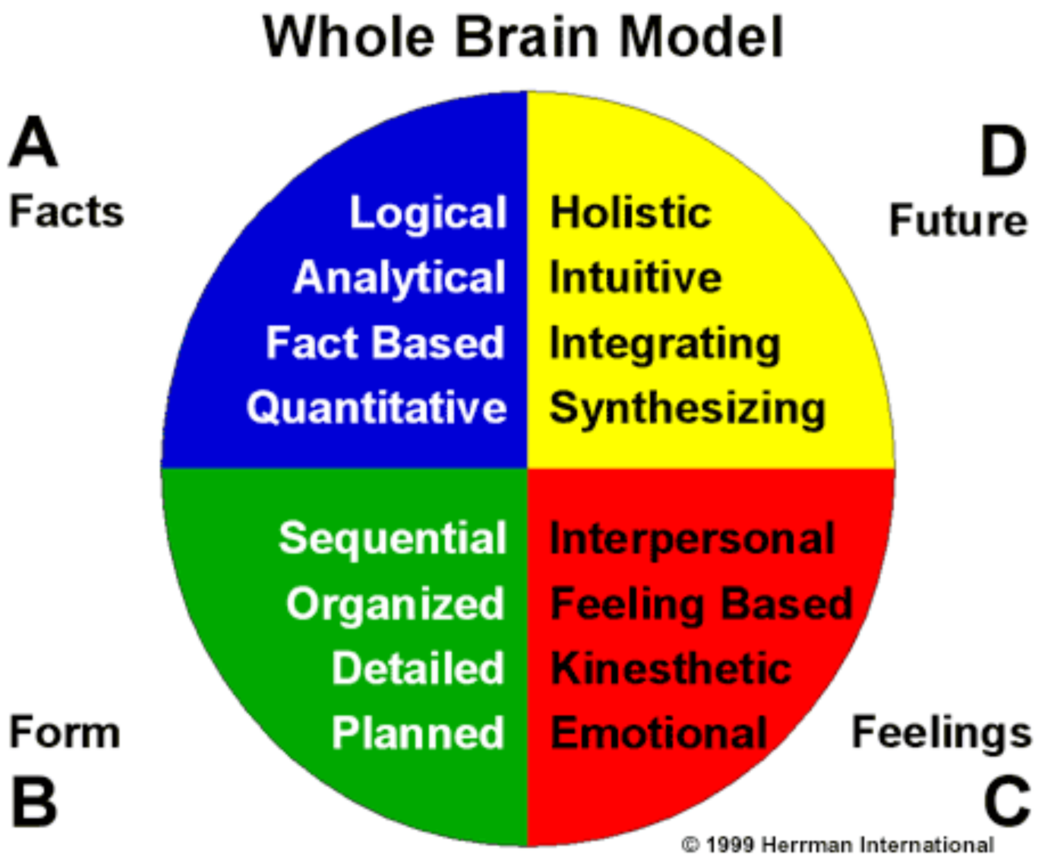


Simple psychological models may be used to help understand how to design better interfaces.

As they try to explain how a user takes in information, thinks about it and then reacts.

Psychological Models

Although such psychological models often use very general concepts and metaphors that are almost impossible to validate or invalidate. (Can still provide a lens for thinking about design)



Bartle Player Types

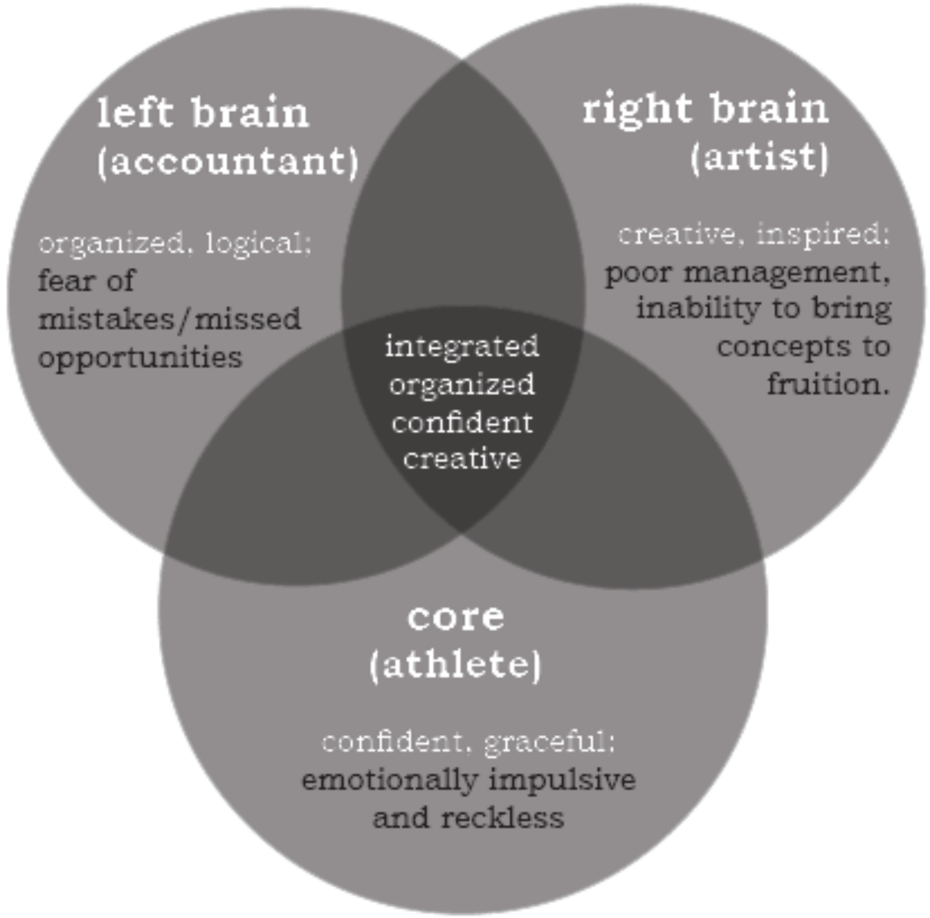
https://www.interaction-design.org/literature/article/bartle-s-player-types-for-gamification

Psychological Models

You should be mindful of the limited nature of most psychological models.

They may tend to ignore low level details such as how the brain really functions.

They may also be very dependent on the context - specific users and situations.



Human Information Processor

But even "pseudo-scientific" models can be useful in understanding some common interface issues and provide a different way of considering interaction in games.

They can also provide some helpful guidelines for designers.

Let's look at an example..

Human Information Processor Model

Card, Stuart; Thomas P. Moran and Allen Newell (1983). The Psychology of Human Computer Interaction. Lawrence Erlbaum Associates.

Human Information Processor

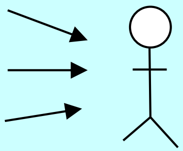
Perceptual System	Cognitive System	Motor System
input	process	output

We could consider human interaction to be like a computer where..

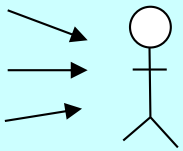
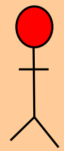
- the human take some input
- processes the information
- and produces some output (motor action)

Then we need to consider the human perceptual, cognitive and motor systems.

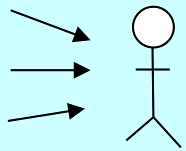
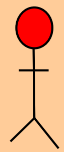
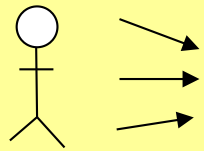
Human Information Processor

Perceptual System	Cognitive System	Motor System
Handles sensory stimulus from the outside world.		
game display 		

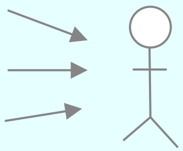
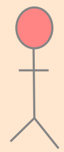
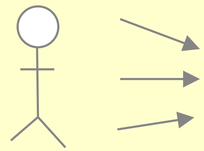
Human Information Processor

Perceptual System	Cognitive System	Motor System
Handles sensory stimulus from the outside world.	Connect the motor and perceptual systems.	
game display 	player processing 	

Human Information Processor

Perceptual System	Cognitive System	Motor System
Handles sensory stimulus from the outside world.	Connect the motor and perceptual systems.	Controls actions of the user.
game display 	player processing 	 game

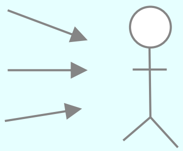
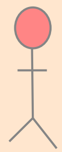
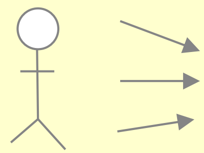
Human Information Processor

Perceptual System	Cognitive System	Motor System
Handles sensory stimulus from the outside world.	Connect the motor and perceptual systems.	Controls actions of the user.
game display 	player processing 	 game

In some games different parts become more critical to understand.

eg. the motor system for action games or cognitive processing in strategy or puzzle games.

Human Information Processor

Perceptual System	Cognitive System	Motor System
Handles sensory stimulus from the outside world.	Connect the motor and perceptual systems.	Controls actions of the user.
game display 	player processing 	 game

Next we will introduce a few issues from each of these areas and discuss some approaches used to design interfaces and how these approaches relate specifically to computer games.

Perception

There are many ways information can be presented to users.

Their perception will ultimately be an individual experience that depends on the things like their previous experiences and even dynamic things like individual biochemistry.

But there are some general principles that are useful to know when designing an interface. We will look at some guidelines later but for the moment we will just mention a few perceptual issues relevant to games.

Depth Perception

True 3D or Stereoscopic displays use two cameras (one for each eye) to create binocular depth cues.

These types of displays have made a small impact on the game market (e.g. Oculus Rift, HTC Vive)

There are still some doubts about health risks:

- accommodation (focus) - convergence conflict
- cybersickness (vestibular mismatch)

Depth Perception

Since there is usually no true 3D - Usually “3D” games are only 2.5D

Game worlds rely on creating the impression of a 3D world using 2D depth cues (depth cues that can be created on a single flat screen).

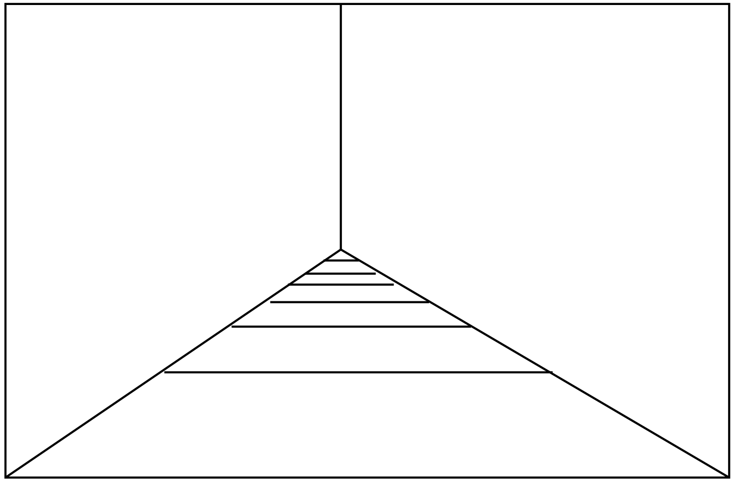
Like artists, computer game designers can do this by using what are sometimes called the "Painter's Algorithms”

Painters Algorithms

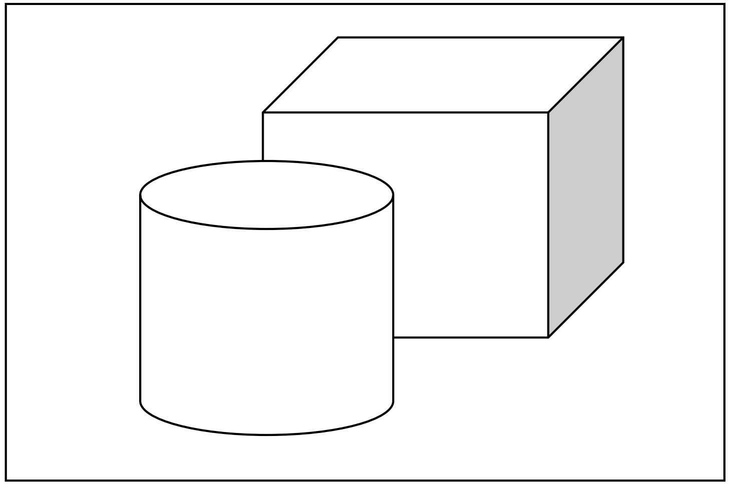
The painter's algorithms provide a mixture of 2D depth cues that can be used when modelling game worlds :

- linear perspective
- occlusion
- texture gradients
- brightness
- atmospheric attenuation
- relative height
- relative size
- familiar size
- shadows

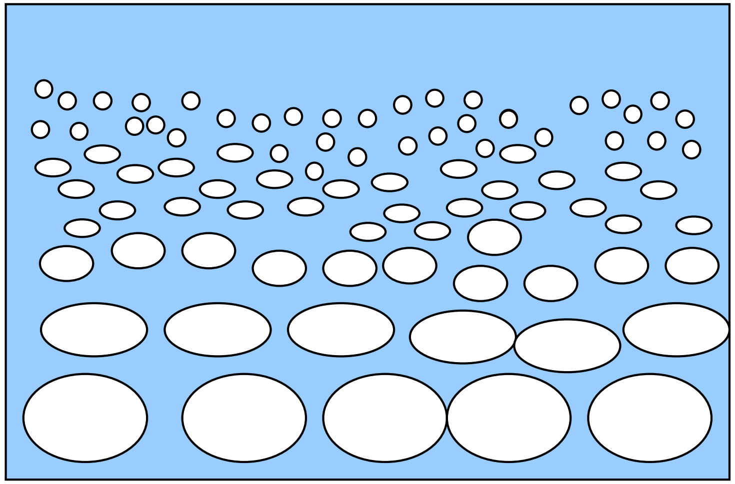
Painters Algorithms



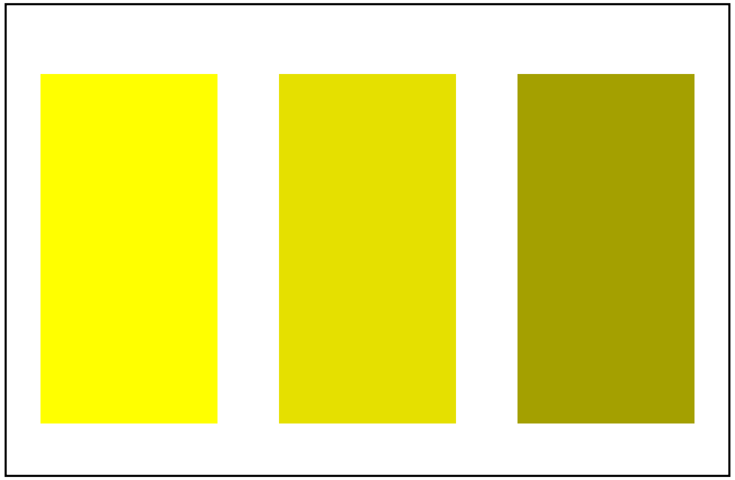
linear perspective



occlusion

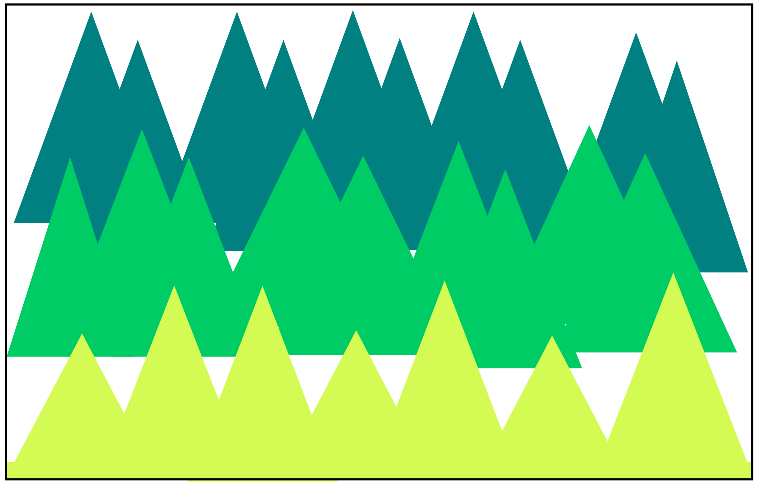


texture gradient

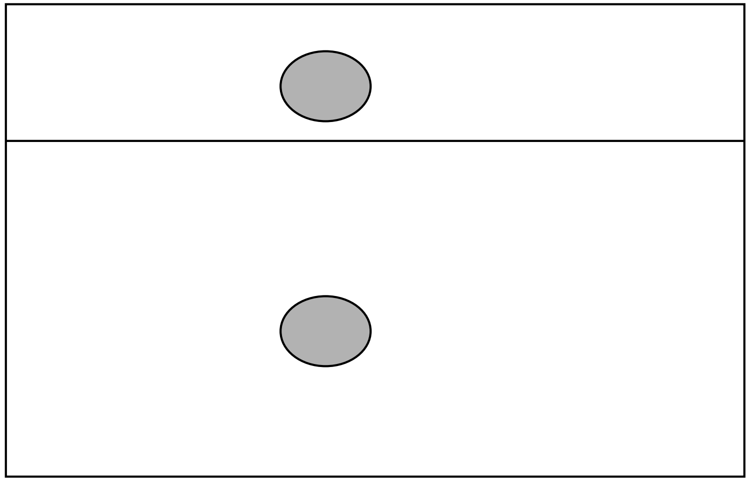


brightness

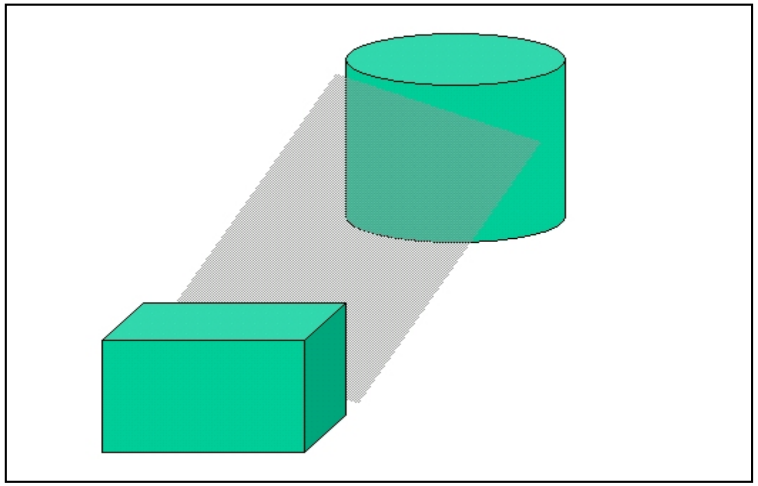
Painters Algorithms



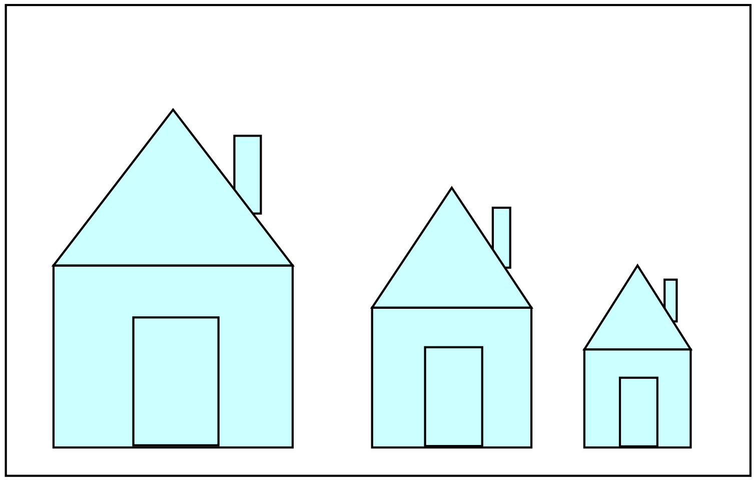
atmospheric attenuation



height in the visual field



shadows



size and past experience

Motor Actions

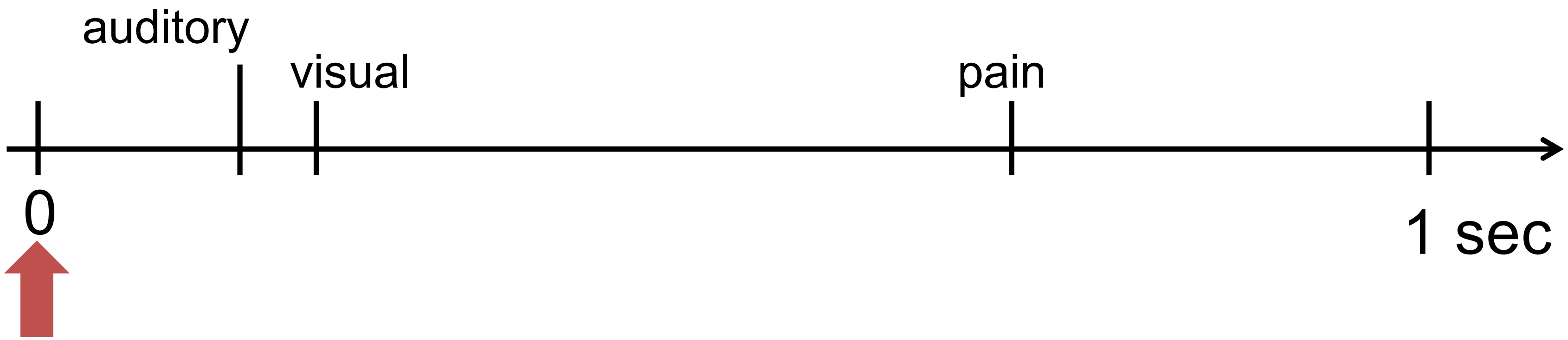
Motor skills are often critical in games, especially where speed and accuracy are important.

For example, in action games.

Motor Actions

Note that the reaction time of the player is dependent on stimulus type:

visual	~ 200ms
auditory	~ 150ms
pain	~ 700ms



Motor Actions

It is also known that…

if you require faster reaction times

it will decrease the accuracy of the unskilled player but not of a skilled player.

Fitt's Law

This is a useful law to know when designing games.

It provides a mathematical relationship that accurately describes how long in takes a player to move between targets

$$T = a + b\log_2\left(1 + \frac{D}{W}\right)$$

where:

- T is the average time taken to complete the movement. (Traditionally, researchers have used the symbol MT for this, to mean *movement time*.)
- a represents the start/stop time of the device and b stands for the inherent speed of the device. These constants can be determined experimentally by fitting a straight line to measured data.
- D is the distance from the starting point to the center of the target. (Traditionally, researchers have used the symbol A for this, to mean the *amplitude* of the movement.)
- W is the width of the target measured along the axis of motion. W can also be thought of as the allowed error tolerance in the final position, since the final point of the motion must fall within $\pm {}^{W}\!/_{2}$ of the target's center.

Fitt's Law

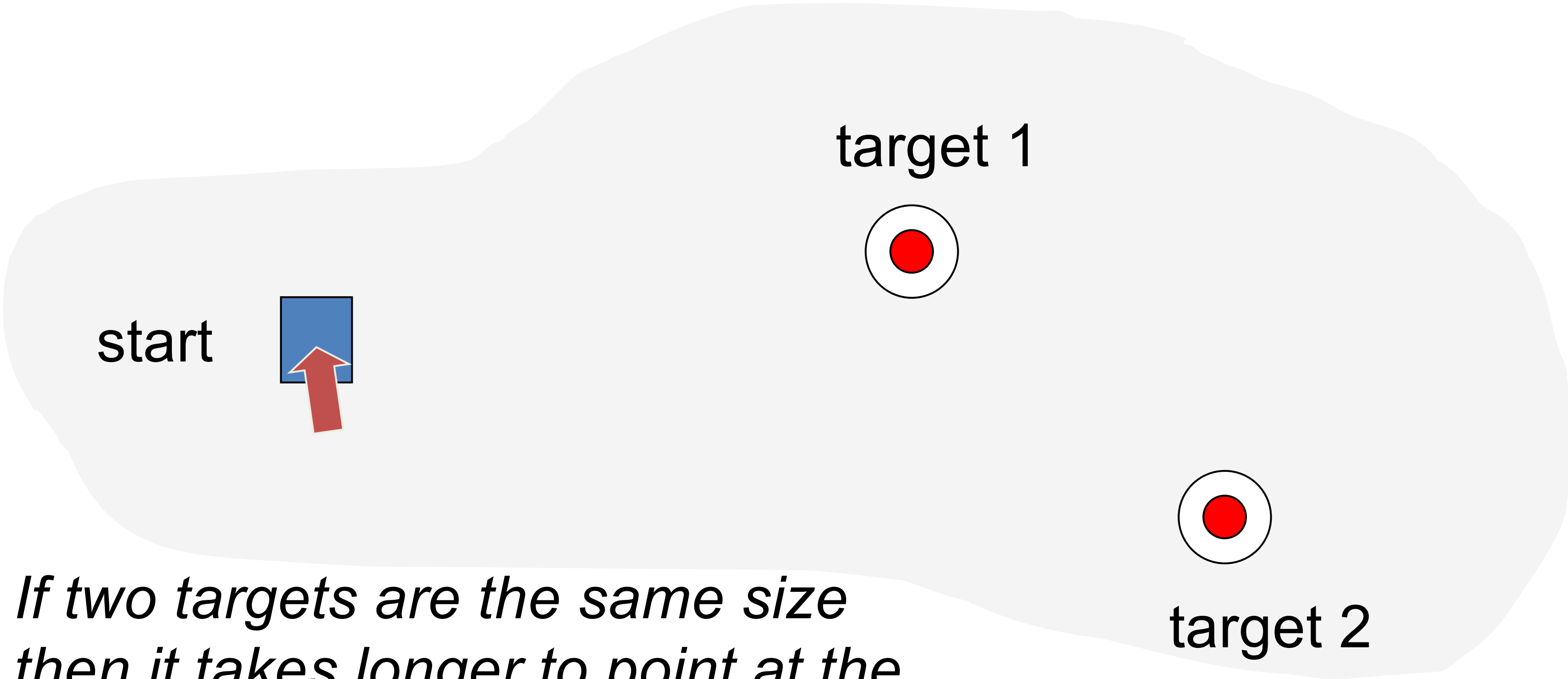
(Without the mathematics), it basically says

it takes longer for players to move their cursor to a target when it is further away

it takes longer for players to move their cursor to a target when the target is made smaller

Fitt's Law

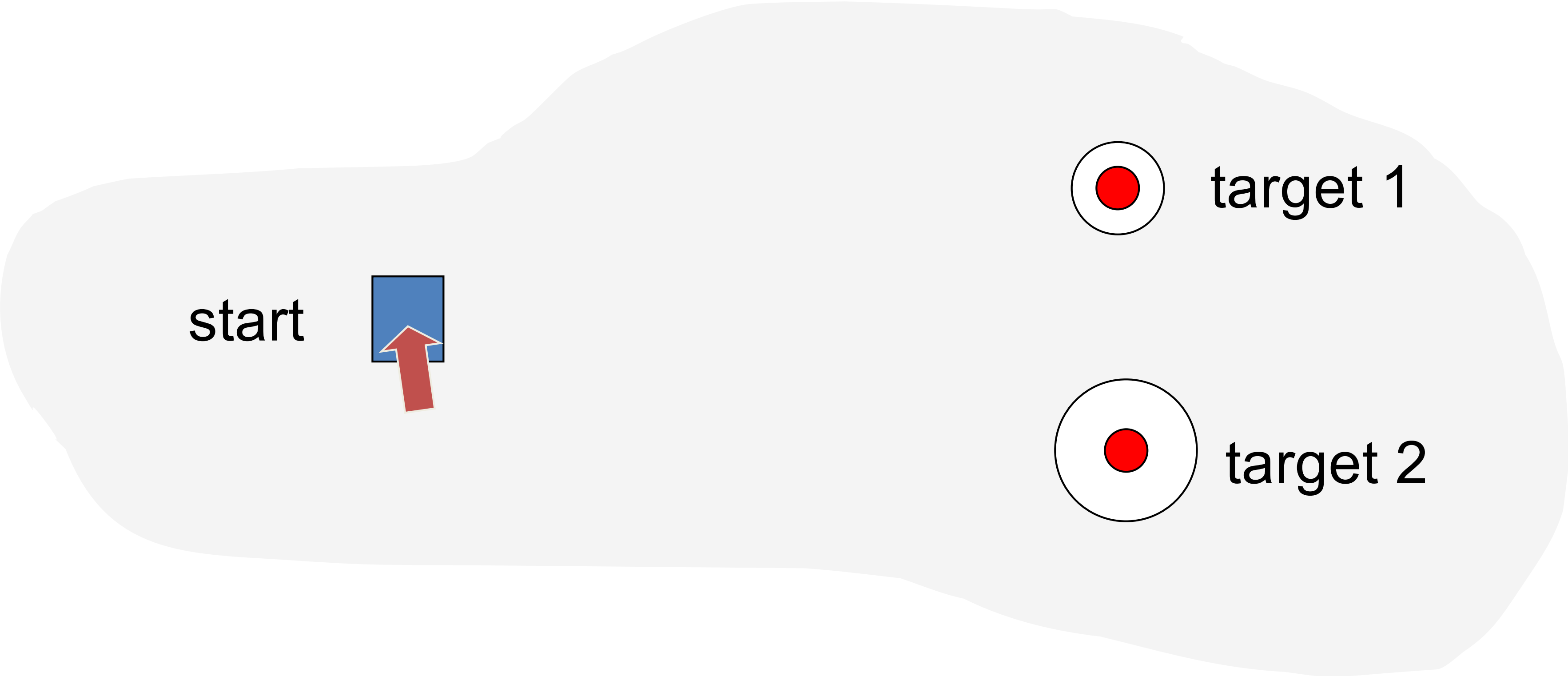
The further away & the smaller the object, the longer the time to locate it and point.



If two targets are the same size then it takes longer to point at the one furthest away

Fitt's Law

The further away & the smaller the object, the longer the time to locate it and point.



If two targets are the same distance away it takes longer to point at the smallest one

Cognition Errors

There are some typical errors that occur in interfaces.

They should considered in any game as well.

A basic one - "to err is human"

Try and avoid having to tell them - try and make sure they just can't (a common way is to use drop down menus rather than typing)

Cognition Errors

There are two situations where a player fails to complete a challenge...

Player has the right intention, but stuffs up

- Perhaps they have poor physical skill or are inattentive.
- The difficultly of the game may need to be reduced, or the player may need to practice the required skill.

Player has the wrong intention, and stuffs up

- Maybe they do not understand what to do
- The game world may need to provide better clues about how to proceed.

Emotional Errors

Games will often try to manipulate the emotional responses of the player. But emotion also influences how the player will respond to situations.

positive emotions $\rightarrow$ creative problem solving

negative emotions $\rightarrow$ narrow thinking

So if the player experience negative emotions it can make it harder to do even easy tasks.

While a positive affect can make it easier to do difficult tasks.

Designing Interfaces

The software user interface designer

- Wants to support users achieving their goals, by providing support for the tasks they must perform.
- Has to work within constraints and make tradeoffs due to technology and costs or conflicting requirements.
- Must understand both the computer and the human and also what happens when they interact.

Designing Interfaces

A game interface designer (likewise)

- Must support players perform their task (overcomes challenges) so they can try and meet the objective or goal of the game.
- Both technology and production issues can create many constraints and the game designer may need to make tradeoffs (e.g. between realistic graphics and interactive frame rates)

Designing Interfaces

Goals

Who is it for? Why do they want it?

Tasks

The steps required to achieve a goal

Constraints

Limitations of materials, platforms, environment, audience, license, time, money

Trade-offs

Decisions made by the designer because requirements, limitations etc are often contradictory

User Centered Design

Most game design processes are based around what is called user-centered (or player-centered) design.

That is, the player's needs should drive development. (The designer is NOT the player)

It is the player that is central – Not the programmer, manager, designer or artist

Usability

In interface design the interface is usually designed with only one or two key aims - these are called "usability goals".

e.g. efficiency, learnability, effectiveness

The usability goals in a game might depend on the expected market or genre.

Unlike most traditional systems, "fun" is usually also a key criteria of the system.

Usability Goals

Effectiveness	This is a general goal. How good is the system at doing what it is supposed to do? That is, is it an effective system to use? This could measure things like quality, quantity, completeness or accuracy of the tasks the system is designed for
Efficiency	This is a more specific goal. How does the system support the user in carrying out their tasks? That is, how efficient is the system to use? This could be measured but things like How much time is required to perform a task? What is the productivity (e.g. output/hour) of an expert? Does the system have unnecessary steps? Is there a bottleneck in the task.
Safety	This is also a specific goal. Does the system protect the user from dangerous conditions or from making serious errors. Does the system impact on the physical safety of the user. Does the system guard against user errors? Does the system allow for recovery from error?
Utility	Does the system provide the right kind of functionality? Does the system allow the user to perform all required tasks and achieve their goals. Are there missing functions and are these critical to the user's tasks?
Learnability	Is the system easy to learn? How much time is needed to learn to use the system? To get started. To become an expert. Is there support for learning? Does the system support different levels of expertise in the user (e.g. shortcuts for the expert).
Memorability	Once learned - is it easy to remember how to use the system? Is there support provided to help users remember how to do tasks?

What the Player Wants

When building the interface for the game.

Remember the player...

1. Needs to know where they are
2. Needs to know what they can do
3. Needs to know where they are going
4. Needs to know where they have been

Design Guidelines

Shneiderman's 8 Golden Rules

1. Strive for consistency
2. Enable frequent users to use shortcuts
3. Offer informative feedback
4. Design dialogs to yield closure
5. Offer error prevention and simple error handling
6. Permit easy reversal of actions
7. Support internal locus of control (player feels in charge)
8. Reduce short-term memory load (Miller's Magic number 7)

Design Guidelines

Game Interface	Controls should be customizable and default to industry standard settings
Game Interface	The interface should be as non-intrusive as possible
Game Interface	A player should always be able to identify their score/status in the game
Game Interface	Follow the trends set by the gaming community to shorten the learning curve
Game Interface	Interfaces should be consistent in control, color, typography, and dialog design
Game Interface	For PC games, consider hiding the main computer interface during game play
Game Interface	Minimize the menu layers of an interface
Game Interface	Minimize control options
Game Interface	Use sound to provide meaningful feedback
Game Interface	Do not expect the user to read a manual

Game Mechanics	Feedback should be given immediately to display user control
Game Mechanics	Get the player involved quickly and easily

Federoff, M. A. (2002) Heuristics And Usability Guidelines For The Creation And Evaluation Of Fun In Video Games. Master of Science in the Department of Telecommunications of Indiana University, USA.

Design Guidelines

Game Play	There should be a clear overriding goal of the game presented early
Game Play	There should be variable difficulty level
Game Play	There should be multiple goals on each level
Game Play	"A good game should be easy to learn and hard to master" (Nolan Bushnell)
Game Play	The game should have an unexpected outcome
Game Play	Artificial intelligence should be reasonable yet unpredictable
Game Play	Game play should be balanced so that there is no definite way to win
Game Play	The game must maintain an illusion of winnability
Game Play	Play should be fair
Game Play	The game should give hints, but not too many
Game Play	The game should give rewards
Game Play	Pace the game to apply pressure to, but not frustrate the player
Game Play	Provide an interesting and absorbing tutorial
Game Play	Allow players to build content
Game Play	Make the game replayable
Game Play	Create a great storyline
Game Play	There must not be any single optimal winning strategy
Game Play	Should use visual and audio effects to arouse interest

Federoff, M. A. (2002) Heuristics And Usability Guidelines For The Creation And Evaluation Of Fun In Video Games. Master of Science in the Department of Telecommunications of Indiana University, USA.

Design Patterns

Many of the interfaces used in computer games have developed over many iterations.

Of course very unique interfaces have also been developed e.g. Wii, EyeToy, Kinect

Myron Kruger
Video Place (1974)

http://www.youtube.com/watch?v=dqZyZrN3PI0

Design Patterns

However, often quite standard designs (patterns) have been adopted for specific genres.

This is not necessarily a lack of imagination but rather because the design works well and has become accepted by players.

Be aware of these common patterns and reuse them were appropriate.

e.g. HUDs in FPS, RTS, etc