# Game Design Summary

## 1. Designers, Games, and Ideas

**Required Skills:** Animation, Anthropology, Archiecture, Brainstorming, Business, Cinematography, Communication, Creative writing, eocnomics, engineering, history, management, mathematics, music, psychology, public speaking, sound design, technical writing, visual arts, and so on!

**Most Important Skill:** Listening – to Team, Audience, Game, (Client), Self

**Game Designer's goal:** create an experience; games enable the experience, but are not *the* experience
Games themselves are simply artifacts

**Experience in all types of entertainment are linear (Books, movies, music, etc.)**
**In video games we have more interaction, leading to complex experiences**

**Creating the Experience:** Introspection is key (psychology, anthropology, and design)
Peril 1: false conclusion
Peril 2: subjectivity VS objectivity ("I only design for people like me" "Personal opinions can be trusted.")
→ Resolved through proper listening

**How to "Correctly" use Introspection?**
Dissect your feelings (why is this important to me, why does this make me feel a certain way)
Observer yourself during experiences:
analyze memories, two passes, short glances, continuos observation
→ Find the "essential experience" (what experience do I want the player to have? How to capture that?)
What is essential to that experience

**Impossible to define what a game is but easy to recognize a game in reality**

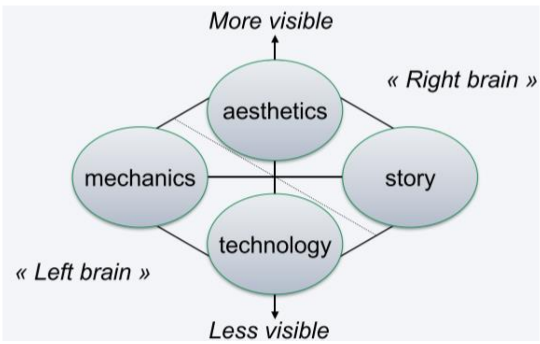
**Game Design:** Lack of terms, Game designers follow instincts, difficult to explicitly identify good and bad aspects in a design.

**What is a game:** Something you play with, are not toys, generate fun, surprise the players
surprises are crucial in entertainment, root of humour, strategy, problem solving, etc.

**Game characteristics:** entered willfully, have goals, conflicts, rules, win/lose conditions, interactive, have challenges, create own internal value, engage players, closed, formal systems

**Internal Values:** Points and money make sense in the game economy (videos, lives, secrets, etc.)

**Anatomy of a game:** Elemental tetrad
Components are related, influence each other
Components all of same importance




**Mechanics:** procedures and rules of the game
Goals! Opportunities to achieve them
Does not exist in movies, music, books, etc.

**Story:** Sequence of events
linear and pre-scripted or branching & emergent (part of your personal experience)
Interaction with the player: creates a game's story or through interaction with other players.

**Aesthetics:** How the game looks, sounds, feels, etc.
Directly and strictly related to player's experience
Sound effects or noises more important than the music, influence the music.

**Technology:** not only high technology; enables and prohibits things to do
medium in which the aesthetics take place

**Holographic Design:** Elements of the game make the experience enjoyable?
What elements of the game detract from the experience?
How can I change game elements to improve the experience? (Affordance)

**Importance of theme:** elements support a theme (Define it as soon as possible, reinforce it!)
Unifying themes = stronger experience; e.g. being a pirate → sense of freedom (reinforce each other)

**Creative Cycle:** think of an idea, try it out, keep changing and testing until it seems good enough
Infinite inspiration (look everywhere), listen to your subconscious

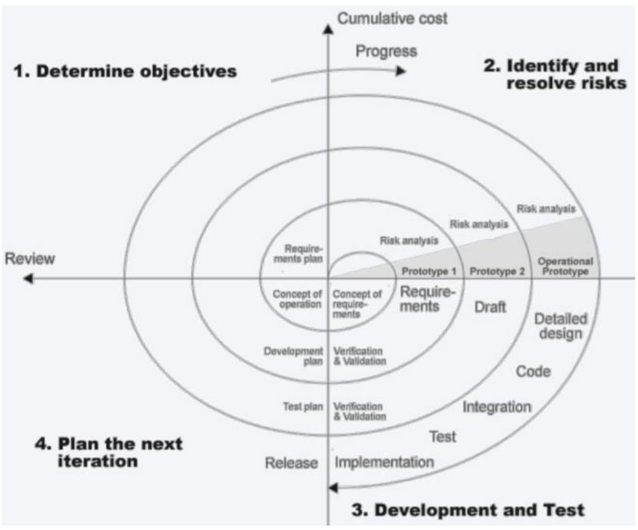
**Inspiration → Design:** state the problem
Advantages: broader creative space (look at problem, not solution)
clear measurement: how well ideas solve the problem?
Communication
The problem often constrains the 4 elements

**Choosing an idea:** do not fall in love with your ideas, be ready to reverse wrong decisions
**8 Filters to validate Ideas:**

| | |
|---|---|
| Does the game feel right? | Will the intended audience like the game enough? |
| Is the game well-designed? (Experience?) | Is this game novel enough? |
| Will this game sell? (Steamspy) | Is it technically possible to develop this game? |
| Does game meet social & community goals? | Do playtesters enjoy the game enough? |
| Your additional or alternate filters | |

**The Loop:** Test & Improve Expensive, not always applicable → how to loop as fast as possible?
How to make every loop count?

**Boehm's Model (Looping Model)**
Basic Design
Figure out greatest risks
Build prototype mitigating those risks
Test them
Come up with a more detailed design
Return to Step 2 with new detailed design




**Risk Mitigation:** Stop thinking positively;
What could keep this game from being great?
How can we stop this?

**Productive Prototyping:**
Answer a question; Forget quality; Don't get attached; Prioritize your prototypes (biggest risks); Parallelize prototypes!!!; Doesn't have to be digital; "Fast loop game engine" (Script > Code); Build toys first (Select fun ones)

**Summary:** State problem; Brainstorm solutions; List risks of using it; Build prototypes to mitigate risks
Test them. If they are good, stop. Else: State new problems, go back to brainstorming solutions

## 2. Player, mind, and mechanics

**Player:** Empathy with your audience
Target demography (talk with them, observe, imagine to be them)

**Biggest increase in demographic 25-35 (Twenties, Thirties)**
Family formation, most of them are casual game players
Hardcore players: important target market, influence their social network

**Can be difference between male & female:**

| Male | Female |
|---|---|
| Mastery/Challenges | Human emotions |
| Destruction | Nurturing |
| Spatial puzzles | Dialog and verbal puzzle |
| Trial and error | Learning by example |

**Psychographics: how players think on the inside**

**LeBlanc's taxnomy of pleasure:**
Sensation (aesthetics of the game)
Fantasy (imaginary worlds)
Narrative
Challenges (one of the core interests of gameplay)
Fellowship (e.g. cooperation)
Discovery
Expression (Design of user levels & heroes)
Submission (leaving the real world behind)

**Bartle's taxonomy:** Achievers: goal oriented
Explorers: pleasure of discovery
Socializers
Killers: imposing themselves on others (even healers)

**Taxonomies are not the end-all answer, so be mindful!**

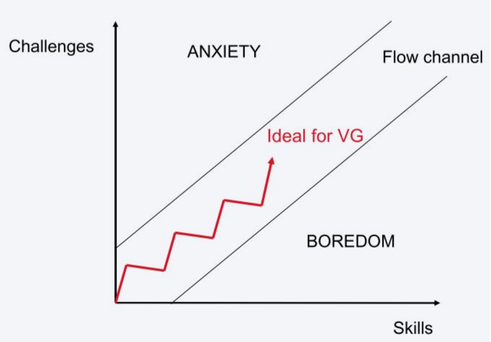
**Player's mind:** Mental ability for gameplay (modelling, focus, empathy, imagination)

**Modelling:** Reality amazingly complex, our mind simplifies realities
Models make things plausible (comics are models; easier to interpret)
Games are models of reality: find right model for your game!

**Focus:** more important than what actually exists
Sense of the world by focusing or ignoring things (filter non interesting information)
Game experience has to be interesting enough to capture player's attention

- clear goals (achievements, also as player statistics)
- no distractions
- direct feedbacks
- continuously challenging

**Flow Channel (Csikszentmihalyi)**



**Imagination:** everyday-use for communication, problem-solving
able to complete brief information
adapts itself very fast to new context
art of compromise: what you should show the player or not

**Mechanics:** core of what games truly are
Taxonomies incomplete: complex mechanics even for simple games, act as mental models (analysis?)
6 main categories: space, objects, actions, rules, skills, and chance

**M1: Space:** Where games take place
Characteristics: discrete or continuous, number of dimensions, bounded areas / connected
Aesthetics can give distorted perception of space (e.g. Grid)
Game can mix / next several space models

**M2: Objects:** Characters, props, scoreboards, etc.
Different objects imply different appearance
Objects have attributes (e.g. position, color) in a state (current value) [visible / hidden to player]
State machines useful to represent attributes & states

**M3: Actions:** What can the player do?
Operative actions ("verbs") "What I do"
Move, Jump, ...
Resultant actions (in the picture of the game) "What happens through use of standard actions)
 Most are not part of the rules
 make noise... to force the enemy to uncover, fix the enemy... flank him

**Emergent Gameplay:** Create interesting actions – elegant, but need care
Creating interesting resulting actions
 more operative actions (useful)
 verbs acting on many objects
 more ways to achieve a goal
 many subjects, e.g. the specialists of Commando
 Side effects changing constraints, see checkers
Innovative games: new actions!

**M4: Rules** Most fundamental mechanic, rules enable the other mechanics and add goals
**Parlett' Rule Analysis**
**1. Operation rules** – describe what player do to play
**2. Foundational rules –** underyling formal structure of the game (E.g. after action, buff / debuff)
**3. Behavioral rules "unwritten" –** sportsmanship
**4. Written rules –** The document, in modern videogames, replaced by more effective tutorials
**5. Laws or "tournament rules" –** competitive settings, clarify or modify standard written rules
**6. Official rules –** merge written rules and laws, over time they become written rules
**7. Advisory rules "rules of strategy" –** hints to paly beter
**8. House rules –** flexible player's rules

**Rules Integration – modes:** too many confuse players, they must know current mode
Traditional written / spoken rules become physical laws in videogame world
Most important rule: goal – concrete, achievable, rewarding

**M5: (Player's) Skills:** related with players and flow channel
Categories: physical (strength, dexterity, coordination, etc.)
mental (memory, observation, puzzle solving)
social ("Anticipation", fooling, coordination, etc.)
Different from visual skills (e.g. your avatar)
Feeling of power for the player
Player's skills can augment at the same time (or not)

**M6: Chance** = uncertainty = surprise
Probability: look at probability distribution curve
Practical probability often more useful than theoretical: Monte Carlo Method
Expected value to determine winning, balanced, and losing (sub-) games
use the real expected value to verify the real value of your actions
expected value does not perfectly predict human behaviour
take into account the perceived probabilities

**Skills and Chance are Coupled:** estimating chance is a skill
Skills have a probability of success
Estimating an opponent's skill is a skill
Predicting pure chance is an imagined skill (human looks for imaginary patterns)
Controlling pure chance is an imaginary skill (Superstition, rituals, etc.)

**Internal Economy:** System (resources produced / consumed / exchanged in quantifiable amounts)
In games, internal economy includes all sorts of resources, that are not part of a real-life economy
Elements of internal economies: Resources, entities

**Resources:** concept that can be measure numerically (almost anything in a game can be a resource)
All economies revolve around flow of resources
Anything the player can produce, gather, collect, or destroy is probably a resource
Not all resources under player's control

**Types of Resources:** tangible / intangible (physical properties, e.g. occupy space, can be moved)
abstract / concrete: abstract resources don't really exist in game, player cannot see them (Tactical advantage)
Computed from the current state of the game (internal logic)

**Entity:** an entity stores a specific amount of a resource (a variable)

**Four Economic Functions:**
Sources: mechanics that create new resources out of nothing,
triggered by events or operate continuously
produce resources at a certain production rate
Drains: take resources out of the game
reduce the amount stored in an entity
remove resources permanently
Converters: turn resources of one kind into another
Traders: move resource from one entity to another, and another resource back in opposite direction
based on exchange rules

# 3. Balance

**12 Types of Game Balance**

**Type 1: Fairness:**
Symmetrical games (equal resources / powers to all players, good to determine best player)
Asymmetrical games (simulation of real situation, different ways to explore a game, personalization, level opposing forces, interesting situations)

**Balance Asymmetrical Games:** Value to each resource (power, skill, etc.) and equal sum of the values)
Theoretical: verify by playtesting
Intransitive relationships ("Rock, paper, scissors")

**Type 2: Challenge Vs. Success** Cf. Flow channel
Techniques: Increase difficulty after each success, let players get through easy parts fast
Create layers of challenge, let players choose difficulty level, playtest with variety of players

**Type 3: Meaningful Choices:** choices must have real impact on the game
Dangers: dominant strategy, especially at the beginning of development
Triangularity: player has to choose between a low or high risk for al ow or high reward

**Type 4: Skill Vs. Chance**: Balance depends on players tastes (also age, gender, culture, etc.)
Alternate use of skills & chance (handing out a card is chance, how to play it is skill)

**Type 5: Head Vs. Hand:** alternate/involve simultaneous problem solving and dexterity (Castlevania)
Announce clearly this balance in your game

**Type 6: Competition vs. Cooperation:** Basic instincts, in video games more competition
Competition and cooperation can coexist (e.g. getting bonus, team competition)

**Type 7: Short vs. Long:** Too short games (players may not develop meaningful strategies)
Too Long: boring, demand too muc time
Altering win conditions should influence length, or change gameplay after some time

**Type 8: Rewards** – people want to be judged favourably, try to add points to economy of the game
Common Rewards (sometimes combined)
Praise, Points, Prolonged play, gateway, spectacle, Expression, Powers, Resources, Completion
How to balance them: Increase value of rewards as player progress, variable rewards

**Type 9: Punishment -** Games are supposed to be fun, but
Punishment create endogenous value, taking risks is exciting, punishments increase challenge
Common Punishments: shaming, loss of points, shortened play, Game over, setback, removal of powers, resource depeletion
→ Reward more effective than punishment

**Type 10: Freedom vs. Controlled experience:** control over everything boring for player, complex for designers
Where to give player freedom? How much? (Horror for us to give freedom to players – expensive!)

**Type 11: Simple vs. Complex:** Innate complexity (complex rules; simulations, "artificial balancing")
Emergent complexity – praised by everyone
"Elegance" (Simple systems performing well in complex situations) – remove elements with only 1-2 purposes
But Character can mitigate the elegance: tower of piece with no tilt would be of no interest

**Type 12: Detail vs. Imagination:** Only detail what you can do well, give details the imagination can use
Familiar worlds do not need much detail, use binocular effects (mostly look at actors in the beginning)
Give details that inspire imagination

**Game Balancing Methodologies:** general method does not exist
State your problem
Doubling and halving instead of fine tuning
train your intuition
Document your model
Tune your model (as you tune your game)
Plan to balance (even in real-time)
Let players do it (in general, to avoid)

**Balancing Game Economies:** How will player earn money? How will player spend it?
Money can also be skill points, similar to balance any other mechanic of the game

**Dynamic Game Balancing:** Dream: adapt game to player’s skills on the fly
Problems: Spoils the reality of the world, exploitable, players improve with practice

# 4. Puzzle and Interface

**Mechanics support puzzles**

**Puzzles:** Sometimes visible in games, something enmeshed (part of design)
Make player stop and think
Are puzzle games? Not repayable, afflicted by dominant strategies, puzzle is a game with dominant strategy

**Puzzle Evolution**: from explicit / incongruous, to integrate in environment in many games

**10 Design Principles to create good puzzles:**
**P1: Goal Easy to Understand:** Clearly define the goal
**P2: Make it easy to get started:** moves may not be obvious, but clear to start
**P3: Give Sense of Progress:** Difference between riddles and puzzles is progress
**P4: Give Sense of Solvability:** Suspect of non-solvability: players give up (Visible progress, show solution)
**P5: Increase Difficulty Gradually:** like for games
**P6: Parallelism – let player rest:** several related puzzles, allow player to switch among them, breaks
**P7: Pyramid Structure:** Series of small puzzles giving clues to a larger puzzle
**P8: Hints increase Interest:** Hints can renew the hope and interest of frustrated players (Hints can cost)
**P9: Give the answer!:** Answer probably more important than solving puzzle
**P10: Perceptual Shifts:** Great pleasure for players able to solve them, almost riddles

**The Interface:** Intermediary between player and game
Interface fails: Experience compromised
Good interface is robust, powerful, as invisible as possible
Good interface makes players feel in control of their experience

**Interface categories:** Physical input (joypad, mouse, joystick, etc.)
Physical output (Screen, audio devices, VR Systems, etc.
Virtual Interface: elements that don't belong to game's worlds
Input: menus, buttons, etc.
Output: Scoreboard, augmented info (name of players), etc.
**Mapping:** Input > Worlds (avatar jumps)
World > output (view of world)
Input > virtual interface (assign experience points)
World > virtual interface (Display stats during a battle)
Virtual Interface > Output (shown data)

**Loop of interaction:** From player to game to player to game to ...
Feedbacks: Influence what players do next, affect understanding adn enjoyment
Experience without feedbacks: Frustrating, confusing
Feedbacks have to be immediate (not only result, but also based on progress)

**Interface and Narrative:** 4 types of interfaces linked to narrative and game geometry
Diegetic, Meta, Spatial, non-diegetic

**Fagerholt & Lorentzon's Model:**

| | | Is the representation visualized in the 3D game space? | |
|---|---|---|---|
| | | no | yes |
| Is the representation existing in the fictional game world? | no | non-diegetic representations | spatial representations |
| | yes | meta representations | diegetic representations |

**Diegetic:** Interface elements exist within game world (player and avatar can interact with them)
enhance narrative experience for player
More immersive and integrated experience
Sometimes no HUD (Head Up Display)

**Meta:** Sometimes UI elements don't fit within the geometry of the game world
can maintain narrative (stay in context of game), often sit on 2D hub plane

**Spatial:** Need to break the narrative
More information than avatar should be aware of, sit within geometry of game's environment
Immerse player; prevent break to experience by jumping to menu screens

**Non-diegetic:** freedom to be removed from games' fiction and geometry
Can adopt their own visual treatment

**Channels of Information:** Interfaces main goal is to communicate information
Games contain huge amount of information to display at the ame time
How to present in an efficient way?

**S1: List and Prioritize Information:** game contains lots of information, differing importance
Need to know always: immediate surrounding
From time to time: currency, health, surrounding, current equipment
Occasionally: Other inventory

**S2: List Channels:** Channel: way of communicating a stream of data
different parts of the screen, avatar and enemies, music & sfx
e.g. main display area, dashboard of information at top of screen, additional modes

**S3: Map information to channels:** Complex task requiring experience, instinct, trial & error

**S4: Review use of dimensions:** channel has several dimensions: textual info, colors, font types & sizes
Using more dimensions reinforce information (e.g. bar of energy for fighting games)

**Modes:** change in one of the mapping arrows, e.g. changing functionality of a button
Add variety to the game, risk confusing the player

**How to avoid troubles:** use as few modes as possible, player has to understand & learn each mode
Avoid overlapping modes (same action in different modes? Keep same button?
Make different modes look as different as possible (camera perspective, visible changes, avatar, etc.)

**Tricks for Interface Design:** Steal good design & adapt (top-down approach) – improve concepts
Design interface from scratch (Bottom-up approach) – explore new ways
Theme the interface
Simulate touch with sounds – our mind associates touch and sound
Use metaphors (know context) – players understands something that has been seen before faster
Test, test, test – as early as possible, as often as possible; paper prototypes!
Break the rules to help your player: wrong ideas can becomes rules of thumb!

## 5. Interest curves and story

**Experience judged by interest curves**

**Interest curves:** useful tools to create entertaining experience → spot troubles
Observe players during the experience, compare their and your curves
Different demographics can have different curves

**Pattern inside patterns:** interest curves can be fractal, with more layers
videogames, typically three levels:

- overall game (introduction >> levels >> end (major climax)
- Each level (new aesthetics & challenges >> harder challenges >> boss at the end of the level)
- Each challenge (introduction >> stepped rising challenge)

**Evaluating the interest:** touchy-feely, unity of interest does not exist
Care about changes of interest (relative metric, behaviour of player)
3 helping factors: can be used together, to estimate interest of a game

**F1: inherent interest**: some events more interesting than others, for instance:
Risk > safety, fancy > plain, unusual > ordinary
In general, events do not stand alone: story are, fables mix of ordinary & fabulous events

**F2: poetry of presentation:** aesthetic of entertainment experience
more beautiful artistry used in presenting the experience, more players find it compelling / interesting
can mix writing, music, dance, acting, graphics design, recitation, etc.

**F3: Projection:** events happening to us are more interesting than those to other people (E.g. lottery)
Power of empathy: create character that players can empathize with easily
more empathy = more interest; interaction allows players to be heroes
Power of imagination: consistent and compelling worlds "immerse" the player
contradictions expulse immediately the player, provide multiple ways to enter world

**Story and Game:** Mid 1970s, videogames with storyboards
delicate relationship between story & gameplay
story, gives game a context and meaning, just like children add stories to abstract game
Story can be part of the experience

**Myth of passive entertainment:** "interactive storytelling is completely different from traditional storytelling"
traditional storytelling involves the listener: questions, imagination, participation (e.g. don't open that!)
Only difference: the ability to take action (in video games)
Desire to act and the emotions exist in both experiences

**The dream:** player has full choice when acting, thinking and communicating
Idea is wonderful, but very hard, probably impossible to realize
The truth: 2 common manners of storytelling: string of pearls, story machine

**String of pearls:** the string: non-interactive story: text, animations, cut scenes, etc.
Pearls: periods of free movement and control
Advantages: finely crafted story; Reward (more story, new challenges), balance between storytelling & gameplay

**Story Machine:** A story is a sequence of related events; Good game is machine generating interesting events
Events stimulate the players to tell someone else what happened
More scripting, less stories produced                    Examples: Sport games, The Sims

**Thoughts About SOP and SM:** Cover almost 99% of existing games; Opposite methods
What about branching story tress? Full AI characters? Multiple endings?
Lot of problems not yet soled

**P1: Good stories have unity:** simple to create interactive story tree (how many are enjoyable?)
First 5 minutes: driving force until the end (intense unity)
1 beginning cannot be perfect for several endings
Most interactive stories with many branching paths feel watery, weak and disconnected

**P2: Combinatorial explosion:** amount of outcomes increases fast
3 choices per level, 10 levels → ~90'000 outcomes
Storytellers fuse outcomes: all choices end up at the same place, really meaningful?

**P3: Multiple Ends Disappoint:** Designer's point of view – players can play more times in the game
Player's point of view: Is this the real ending? Do I have to play the whole game again to see another?
Exceptions: different quests, goals, and endings (good / evil)
2 completely different story → lots of work for "half" the content

**P4: Not enough verbs:** Characters in videos able to jump, run, shoot, crouch, climb, fly, etc.
Most of what happens in story is communication (talk, ask, negotiate, argue, etc.)
→ not properly supported by video games yet

**P5: Freedom against destiny:** most insoluble problem: Why don't video games make us cry?
Tragic stories considered most serious, important, and moving types
Generally, off limits to story tellers: must give up inevitability
tragic stories already anticipate their unstoppable end

**The Dream of Storytelling:** dream of interactive storytelling obsessed with story, not experience
experience is a story, as well as technology, aesthetics and gameplay
8 Tricks to make story elements involving and interesting

**T1: Goals, obstacles, and conflicts**
Old maxim of Hollywood: character with a goal
obstacles keep him from reaching
When the character tries to overcome obstacles, conflicts tend to arise
Interests of this structure: 1 clear goal, obstacles >> problem-solving; conflicts >> unpredictable results

**T2: Provide simplicity & transcendence:**
Simplicity: the game world is simpler than the real one
Transcendence: the player is more powerful in the game world than in the real one
Recurrent worlds because of this combination: medieval & fantasy
futuristic (technology often similar to magic), war, modern-like (GTA, Sims)

**T3: Consider the Hero's Journey:** Propp's functions – structure of fables
Campbell 1949 – Monomyth (Hero's journey!)
Underlying structure that mythological stories seem to share
Vogler 1992 – guide based on Campell's archetypes (write story first, use it as a lens)

**Vogler's Synopsis of the Hero's Journey:**
The ordinary world, Call to adventure, Refusal of the call, Meeting with the mentor, Crossing the threshold, Test / allies / enemies, Approaching the cave, The ordeal (hero faces a peak life or death crisis)
The reward, The road back (to ordinary world), Resurrection (greater crisis), Returning with the elixir

**T4: Put your Story to Work** Many designers begin with story, which can be a mistake (follow story too slavishly)

e.g. point & click adventures; is the most pliable element of the tetrad

Adapt to the story: overcome technical limits, make gameplay coherent

**T5: Keep Story World Consistent:** small inconsistence breaks reality of world

Define set of rules for your and respect them!

Breaking your own rules will frustrate the player and your world will appear ridiculous

**T6: Make Story World Accessible:** Truth is not always your friend when you are a storyteller

**T7: Use Clichés Judiciously:** Common criticism: game overuse of clichés, are familiar to players

Best to combine them with something novel

**T8 Sometimes a Map brings a story to life:** game contains physical spaces, sketch matches & places

Story can naturally take shape through this



5. Interest curves and story

## 6. Indirect Control and Worlds

**Story and Game Structure can be merged with indirect control**

**Feeling of Freedom:** Heart of conflict between story & gameplay, player sense of control
facilitate projection in the world
Not necessary to give player true freedom; only feeling of freedom
Designer does not have direct control on what player does, but indirect control; 6 proposed IC methods:

**ICM1: Constraints:** Freedom of choice; selection sometimes better than open

**ICM2: Goals:** will indirectly control the player, sculpt game around goals: (Distance of goals as additional incentive)
players only do things useful to accomplish a goal, creating content that players never see is a waste

**ICM3: Interface:** influences the player, virtual interfaces have the same effects (mental models via avatar!)

**ICM4: Visual Design:** People go where they look, graphical composition directs players / give freedom

**ICM5: Characters:** Control the player through computer-controller characters
Players willingly obey, help, protect, or destroy them (empathy is key!)

**Collusion:** Characters in the game have 2 goals:
Personal goals (e.g. destroy player)
Story related goals (e.g. drive player towards one place)

**ICM6: Music:** "language of the soul" – not only useful for atmosphere
Restaurants: Fast music (people eat fast), slow music (people will stay longer!)
Games: Look for something hidden, destroy everything, move slowly & carefully, ...

**Transmedia Worlds:** fantasy world, entered through many different media (print, video, toys, games, etc.)
World exists apart from media supporting it
Real product: the world – can not be sold directly, products act as gateway, gateways must be consistent
"We want those worlds to be real!"

**Transmedia Worlds are Powerful:** personal utopia for fans, fantasy lasting all along life, occasional visits through gateways

**Transmedia Worlds are Long Lived:** Continue for a long time, adults share their worlds with the children

**Transmedia Worlds evolve over time**

**What Transmedia Worlds share:** rooted in a single medium, are intuitive, have creative individuals at core
facilitate telling of many stories, make sense through any of their gateways
they are about wish fulfillment

**Transmedia worlds are future of entertainment, designers asked to create more and more gateways**

## 7. Characters

**Worlds contain characters**

**Nature of game characters:** great stories require memorable characters
Characters in game different from other media?

**(Re-) Current Patterns:**
mental to physical: novels – deep psychic struggles; movies – emotional & physical, video games – physical
reality to fantasy: novels – often reality; movies – reality rooted, leaning into fantasy, VG - mainly fantasy
complex to simple: plot and characters depth

**Avatars:** character controlled by the player "god taking physical form on the earth" (Sanskrit)
Double relationship between player & avatar: sometimes apart, sometimes completely projected
First-and third-person views are both immersive: virtual reality: immersion and extracorporeal experiences

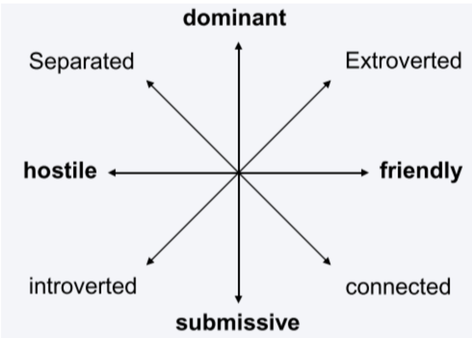
**Good avatar:** ideal: players dream about being might warriors, wizards, princesses, secret agents, ...
Blank Slate: iconic avatar, less details; more opportunities to project
alien, foreign, or scary characters often represented with more details
Allowing players to put their photo on avatars interesting only in short term

**Persuasive game characters:** avatar corresponds to protagonist of traditional stories
many other characters (how to create compelling characters? 9 Tips:

**T1: List Character Functions:** list functions to fulfill in the game & associate your imagined characters
e.g.: hero, mentor, tutor, final boss, hostage, the girl
Characters can have multiple roles! (if you need new characters to fill all roles, create them)

**T2: define & use character traits:** avoid flat dialogs, characters have to talk & act as real people
Define traits of your characters: loves & hates, how they dress, their past
Traits can be contradictory, because real persons have contradictory traits

**T3: Use Interpersonal Circumplex**




**T4: Make a character Web:** explores how characters feel about each other and why

**T5: Use Status:** any time persons interact, a negotiation of status takes place
Low status behaviour: fidgeting, avoiding eye contact, touching one's own face, and generally being tense
High status behaviour: relaxed, in control, strong eye contact, not moving head while speaking
subconscious behaviour, not absolute to an individual
**Conveying the status:** characters aware of other characters (space occupation, size difference, etc.)

**T6: Power of the Voice:** human voice can affect us at deep subconscious level
Video games: voices (were) often too primitive: developers inexperienced, backwards process, translations!

**T7: Power of the Face:** humans complex and expressive face,
video games, actions more attention than emotions
video games with good facial animation particularly appreciated
discern faces easier than number and text

**T8: Good Stories Transform characters:** in great stories characters change
Sometimes protagonist can't change, but secondary characters may evolve
transformations can be temporary or permanent

**T9: Avoid uncanny valley:** the closer something is seeming human, the more people empathize with it
not true for robots imitating humans → uncanny valley is not static!

# 8. Spaces, aesthetics and other players

**Worlds contain spaces**

**Purpose of architecture:** general perception – modern buildings with unusual shapes
Primary purpose: control a person's experience
  Shade and dryness → shelters
  Safety and Security → Walls
  Houses, schools, churches, stadiums, etc.
A well designed building creates the experience we want when we are inside
Architects use indirect control

**Organizing the game space**
Linear spaces: forwards/backwards along a line
Grids: easy for people & computers to understand
Webs: point connect with paths
Points in space: idea of wandering in a desert and occasionally returning to an oasis
Divided Spaces: similar to a real map
→ These principles can all be combined

**Landmarks:** too much order and chaos are both monotonous
Landmarks help players to find where they are going, make space interesting to look at

**Real vs. Virtual architecture**: perspective on architecture useful, but games allowed to have strange spaces: wasted space, weird and dangerous architectural features, no relationship with outside environment physically impossible overlaps
We don't notice how strange buildings are! (human mind weak when translating 3D → 2D)

**The Sense of Scale:** in real spaces, the sense of scale comes naturally to us (lightning, shadows, own bodies)
Virtual spaces, scale not always clear: lack of real-world cues, confusing for the player
Avoid out of scale proportions in your world (strange / funny concepts, use standard metrics!)
Even proportioned elements look out of scale:
  eye height: low & high camera distort the world
  people and doorways: use sized man-made objects
  texture scaling: matching between it and the object

**Third person distortion:** every has developed sense of how our bodies fit into the world that we see
Duality in third-person video-games: in body of avatar, but also floating behind our body
Distorted sense of proportions in interior spaces: space feels crowded

**Aesthetics:** Part of the tetrad and of the experience (not only surface details)
Good artworks can: draw the player into a game the might have passed over
  make the game world solid, real and magnificent
  reward the player
  make players tolerate defects in the design

**Learning to See:** Key for creating great artworks is in the ability to see; "See" is more than recognizing object
Really see an object means see: shapes, colors, proportions, shadows, reflections, textures
  relationships to its environment & people using it
  functions and meaning
our brain categorizes information (left side)
Seeing is ability to train (right side)

**Aesthetics can guide vision:** our minds are very visual (illustrations & sketches change course of design)
Game in the mind looks different on paper
Concept arts: provide uniting vision of game experience
validate feasibility and usability of the interface
Artistic skills are big advantage for designers
If you are not skilled, find an artistic partner with whom you communicate well

**Advantage of Concept arts:** Strong game designs with good concept arts
Make your ideas clear to everyone
Let people see your game world
Make people exited about playing / working on the game
Allows you to secure funding
Illustration is a kind of prototype!

**How much is enough?** Most artist want to make everything look gorgeous → beautiful arts take time!
Few details in the right places can make the game world seem far larger & richer than it is ("distant mountains")

**Use Audio:** Aesthetics more than visual arts
Audio feedback: more visceral than visual, more easily simulates touch
Common development mistake: choice of music in the end of the project!
Choose music as early as possible: can channel design of the game

**Balancing Art and Technology:** Artist are empowered and restrained by technology
Engineers are empowered and restrained by art
Do not ignore engineers' aesthetics participation
Ideal case: integrate "technical artist" in your team: eye of artist, mind of computer programmer!

**We are not alone – games played together!**
**Why we play with others:** competition, collaboration, meeting up, exploring our friend & ourselves

**Multiplayer games:** lot of work, hard to control, 4 times the effort due to debug & balancing

## 9. Communities, Team, Documents, & Playtesting

**Other players sometimes form communities**

**Communities:** Games inspire real passion (fans, plaer, designers)
Sense of community: Membership, influence, integration / fulfilment of needs, shared emotional connection

**Interest of creating communities:** Being part of a community fills a social need
Longer "period of contagion" (recommendation of friend most influential purchasing factor)
More hours of play (game engendering community played even if other qualities lack)
10 tips for strong communities:

**T1: Foster Friendship:** meaning online relationships require
Ability to talk (nonverbal is not enough)
Someone worth talking to (consider demographics)
Something worth talking about (Strategy, events, rule changes)
Game should support friendship's phases: breaking the ice, becoming friends, staying friends

**T2: Put Conflicts at the heart:** conflict is at the heart of each community
Soccer team with other teams; teacher association fighting for better schools
Conflicts part of all games: not always resulting in communities; against player or together against game

**T3: Use Architecture to shape communities:**
2 Types of neighbourhoods (don't know neighbours or everybody knows everybody)
Side effect of community designed: walkable / few traffic > see each other > chance to communicate
MMO games: Places, game business zones

**T4: Create Community Property:** Things owned by several players encourage them to band together
Tangible: ships, buildings, etc.
Abstract: Guild status

**T5: Let Players express themselves:** do not reduce expression to gameplay strategies/styles of play
e.g. add expressive avatar creator, monopoly is for 2-8 players, but 12 playing pieces
Some game based on expression: Pictionary

**T6 Support Three levels:** Newbie (learning to play, experienced players as mentors)
The players (immersed in game activities)
elder (could stop to play, propose alternatives: teaching, management, creation, difficulty, privileges)

**T7: Force players to depend on each other:** aid from other players will help to solve conflicts
Games that can be mastered when playing solo reduce the value of community

**T8: Manage your community:** create appropriate tools & system (Communication, organization, editors)
Involve professional community managers (Feedback loop between designers and players)

**T9: Obligation to others is powerful:** deeply felt in all cultures
Create situations where players make promises: e.g. to meet up to fight, to save someone that saved us
Players will take them seriously; e.g. obligation to guild

**T10: Create Community Events:** Real world (meetings, parties, competitions, awards, etc.)
Pretty much the same in the virtual world
Events give players: something to look forward to; shared experienced; punctuate time; connect with others

**Challenge of Griefing:** some players enjoy to steal, trick and torture other player
Anti-griefing policies: Problem: have police and court of law
Better to avoid some game systems: Player vs. Player combat (if not at the heart of the game)
stealing (Even indirectly), trading, obscenities, blocking the way, loopholes

**Designer usually works with a team!**
**Team:** enormous, diversity of skills required – artistic, technical, design, business
Secret to successful teamwork is love
Love problems: Members incapable of loving any game
Members in love with a different game than the one they are making
Members in love with different visions of the same game

**When you do not love the game:** Mediocre games at the best
Try to find aspects that you can love (interface, mechanics)
If you can't love the game, love the audience (imagine you're offering a special gift)
Otherwise, pretend to love the game!

**Designing together:** everybody has opinions
Ignoring the team implies catastrophic consequences
Include team (whenever possible) in the design:
more ideas to choose from, weed out flawed ideas quickly, view game from many perspectives
Makes everyone feel involved and responsible

**Core Design Team:** do not involve everyone in the design all the time
Compose core design team: interested & productive persons during meetings
After taking a decision with the core design team, inform rest of team
Typical process: Brainstorm, independent design, discussion, design presentation

**Team Communication Key Issues:**
Objectivity (do not impose your ideas, ask and let team discuss)
Clarity, Persistence (write things down)
Comfort, respect, trust, honesty, privacy, unity

**Team communicates through documents**

**The game design document:** simply doesn't exist – each game & team is different
Purpose of documents: Memory, Communication
Rarely, only one document serves all necessary purposes

**Usual Document Groups:** Design (Game design overview, detailed design document, story overview)
Engineering (Technical design document, pipeline overview, system limitation, art bible, concept art overview)
Management (Game budget, project schedule)
Writing (story bible – everybody contributes to write the story, script, game tutorial & manuals)
Players (Game walkthrough)

**Games are created through playtesting**
**Playtesting:** necessary to solve problems in the experience
4 main types of testing:
Focus group (interview players about likes / dislikes, used to determine if they like an idea)
QA (quality assurance; looking for bugs)
Usability testing (are interfaces and systems easy to use?)
Playtesting (Does game engender designed experience?)

**Why Playtesting:** kind of prototype of the game experience
Clear goals for playtesting are necessary (Waste of time otherwise)
Define specific questions!

**Who should test the game?** Common target demographics!
Developers: useful feedbacks & know confidential information; too close to the game
Friends: highly available & comfortable talking to you, predisposed to like the game
Expert gamers: Detailed account, know technical terms / examples; demand more complexity & difficulty
Tissue testers: fresh eyes, very valuable; risk of game with strong first-time appeal, then boring

**Where should I test the game?**
In your studio (uncomfortable for play testers)
Playtesting lab (expensive and rare, but wonderful)
At some public venue (shopping mall, campus, etc. – cheap but difficult to find right testers)
Play tester's home (real conditions, but limited evaluation)
On the internet (lot of people & hardware configurations, but low quality of testing)

**What should I test?** Things you know you are looking for (answers to questions; special release if needed)
Things you don't know you are looking for: surprises, deeply observe players,
record & understand unusual reactions

**How should I test the game?**
Should designers be present? Risk to influence players, but enriching
What to tell up front? Don't use misplaced words, prepare speech and improve it (tutorial?)
Where do designers look? Players faces, but also hands, controls, screen, etc.
What other data should designer collect during play? Logs about everything possible
Will designer disturb players mid-game? Delicate trade-off; "think-aloud protocol"?

**Data collected after play session?** Survey (quantifiable answers pictures / five-point scale, existing systems)
Interviews (Ideal to replace too complex questions, prepare script, ask more than you need)

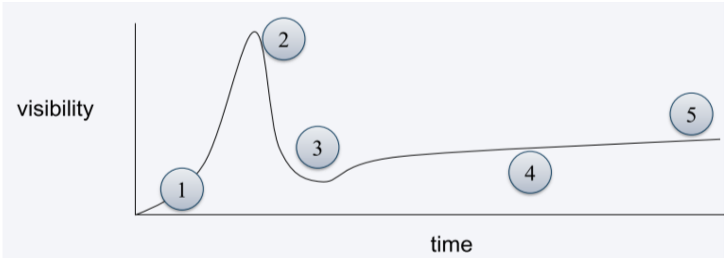
## 10. Technology, clients, and pitches

**Team builds game with technology**

**Technology:** most dynamic element of the tetrad; (volatile, rapid advancements, unpredictable)
Medium of game: computer & electronics; paper, tokens, dice, etc.
Some innovations imply new kinds of games: players want innovation, last decade "adolescence of domain?"

**Foundational vs. Decorational technology:** Foundational: new kind of experience possible
Decorational: make existing experiences better

**The Hype Cycle:** Model by Gartner Research
1. Technology trigger: discovery / announcement
2. peak of inflated expectations
3. Trough of disillusionment
4. Slope of enlightenment
5. Plateau of productivity




**Design and the hype cycle:** Hype cycle happens every time
Designers need to know about it for: Immunity: do not risk on technologies you haven't seen work
Inoculation: someone in your team wants to integrate game in new craze technology
Fundraising: investors sometimes charmed by inflated expectations

**Innovator's dilemma (clayton Christensen)**
Companies fail because they listen to customer

**Progress:** Speed of technological progress is accelerating (more difficult to predict future)
Singularity: moment where technological progress too fast to make predictions
Do no try to understand only current technologies, but what is coming next (predict trends!)

**Clients:** game publishers, media companies, someone without entertainment experience
Clients pay you to make the game: strong opinions about your game, need ability to compromise

**Client's bad suggestions:** 3 ways to deal with them: Agree, why it is bad, try to understand client
Suggestions are solutions to unstated problems: state problem → lead to better solution

**Deal with clients:** some clients do not know what they really want
opposite of strong opinions, help them figure out their desires
Learn about client, personally & professionally, understand their wants & needs
Clients have 3 layers of desire: Words, Mind, Heart

**Pitching:** Convince people that your game is worth the risk
You are the best and only person to pitch your game (if you don't believe in it, nobody will)
Who will you pitch to? Team members & potential partners (concept)
Management (approval for prototyping)
Publisher (try to get development deal)
Reporters at game conferences

**Negotiation of power:**
Power is the ability to get what you want; Knowing "what you want" is essential (focus effort effectively)
Pitching game is negotiation of power (your wants, clients' want), idea not judged on overall merit, but by how useful it is

**Successful pitch tips:** Get in the door, Show you are serious, Be organized, Be genuinely passionate, Assume their POV, Design the pitch, Know all the details, Exude confidence, Be flexible, Rehearse, Get them to own it, Follow up

## 11. Profit, transform, responsibilities and motivation

**The designer and the client want the game to make a profit**

**Profit & Love:** Designers love creating games, “accept work as *amateur*”
Industry & investors want money (client are ready to finance more profitable domains)
“The one with the gold makes the rules” – understand enough about games business, making money

**Units Sold:** Compare your game to others that have come before (SteamSpy)
Units sold equals success (or not) – lot of excellent games were unsuccessful
Hard to get these numbers (published web, magazine, easier with big developer or publisher)
Publishers use these data to estimate the potential success of your game (very hard to argue)

**Breakeven:** amount of game units to sell before publishers makes back invested money (important!)

**Know the Top Sellers:** know the list of the top 10 best-selling games publishers know it
The game industry is a hit-driven business
Electronic Entertainment Design and research breaks games down to discover top sellers’ important features
Be familiar with the hits in your market and demographic... and possibly why they are (money people like)

**Some Common Terminologies:**
SKU (Stock Keeping Unit): unique inventory item for a store
Cogs (Cost of Good sold): cost to make a game unit
Burn rate (cost of the studio per month)
Sold in vs. Sold through (sold in: retailer buys form the publisher; sold through: players buy)
NPV (net present value) Money in your hand now worth more than in your hand in the future)
Christmas: 75% of all games in USA sold during that season

**Games transform their players.**
**How do games change us?** Effect of games on long term?
Great debate: distraction? Dangerous? Benefic?

**Can Games be Good:** Emotional maintenance, connecting socially (not only multiplayer), exercise
Education, new insights, curiosity

**Can Games be bad:** Traditional dangerous things, violence, addiction

**Experiences:** do games change people? Experiences do → game designers create experiences
Too early to really know how games change people

**Designers have responsibilities**
**Dangers of obscurity:** game designers are not famous & respected (like screen writers)
Publishers don’t want you to become famous → cost would increase
but you have to take anyway ethical responsibilities

**Ethical responsibilities:** most real risk for people playing online is to meet dangerous strangers
Try to make people’s lives better
Games are experience that can affect people in the future

**Designer is motivated!**

**Listening is the most important skill for a designer, is making a game worth your time?**
**Why are you doing this?** Find your true personal reason to design games & use it to motivate & empower

## T. Technical Slides Topic Overview

**T1 – 2D Game Engines**
Video Games general info, Time Constraints in Games, engine
Real-time loops in VG, Simultaneity, Simple Loop algorithm, The Game Object
Rendering – 2D / Sprite
Updating the Game World Physics, Input, Game Engines, Rigid bodies

**T2 – A glance to 3D**
3D Engines, scenes, geometry, mapping, textures, lighting, shaders

**T3 – Advanced SFX**
Shaders, advanced graphics effect, 3D pipeline, shaders, vertex / pixel shader function

**T4 – AI basics**
Pathfinding, characteristics of AI, different models, planning & problem-solving, evolutionary

**T5 – Action Oriented AI**
Action, choreographed, tracking, chasing, evasion, patrolling, waypoint, cover

**T6 – Tactical AI**
Action, Thinking, path-finding (crash & turn), Dijkstra, A*, Group behavior, formation based movement, influence maps

## T1. 2D Game Engines

**Static World** – doesn't update, reactive to user interaction, essentially "working GUI"
**Living World** – changes independently, user entity of world, "monitoring application"

**Time constraints:** real-time software (time critical nature, time-constrained conditions)
Graphics (> 25 fps), Music (continuous), Interaction (<50ms)
Loading time, world updating, etc.

**Engine Functionality:** Load/Unload/Init resources (Engine / Game / Level resources)
Manage Game Loop (Rendering, Network, Physics, World Update)

**Real-time Loops:** State of the world (Objects, NPC, etc.)
Interaction (Inputs, User as special entity in the world)
Aesthetics (Outputs: sound, music, grahpics)

**Simultaneity:** "2 main parallel routines"
1. World update (constant speed independent of hardware, gameplay affected by speed, 10-15 updates suffice)
2. Rendering (as often as possible); new PCs allow smoother animation, better fps
Solution: Single threaded program, simulate threads with software loops & timers
Idea: execute update & render sequentially, skip update cells, render as often as possible
*Simple Loop Algorithm*

**Game Object**: Any possible resource in the game (avatar, NPC, scoreboards)
Properties (Transform, Rendering, Physics [rigid body, colliders], specific properties)
Life cycle (Init, Start, Update, Render, Destroy; similar to game loop

**2D Game Characters:** characters stored in rectangular bitmap (drawing surfaces is fast)
Animation obtained by alternating images (frames)
Named "sprite" (reference to first games involving ghosts, sprites, knights, etc.)
Blitting: operation of layer sprites onscreen: "block image transfer" (BLIT)

**Sprite Representation:** contains visible & non-visible pixels
Maps, "transparent color" or alpha channel
Several images or spritesheet

**Level Graphics:** levels use maps bigger than screen
Background: big image or composition of small images (tiles)
Often big images split into sub-images for optimization purposes

**Parallax Scrolling:** Optical phenomenon (displacement of distant object, viewed from 2 positions)
Foreground objects seem to move faster than background
Semi 3D effect, adding sense of depth

**Isometric tiles:** Simulation of 3D Scenario (invented by architects & industrial designers)
Representation of objects and levels as if they were rotated of 45 degrees
Parallel perspective, lines do not converge in conic perspective, no distortions

**Page-Swap / Big maps:** scrolling background, without being limited to a set of tiles
draw complete image, divide into sectors (dynamic loading during game)
active sectors in main memory, rest on secondary devices, leads to faster loading

**Update:**

1. Refresh Inputs
2. Update game objects (usually behaviors triggered at this stage)
3. Update physical engine (rigid body, particles, etc.)
4. Collision detection (2 stages: broard + narrow)
5. Collision resolution

**Input Devices:** 1 and only 1 peripheral (related to gameplay & experience!)
difficult to integrate multiple peripherals
Abstract model can be solution (controller mapped to each device, no impact on game engine)

**Input: Synchronous Model**

1. World is waiting
2. User interactions create events
3. World reacts to event (e.g. Java event model)

**Asynchronous Table:** table contains "Boolean" values mapped to different entries
Better than consulting each possible entry (consistence of data: use values recorded at the same time)

**Physics Engine:** Simulation of bodies, possible to define parameters, based on Newton's mechanics
Newton's three laws of motion: Inertia, Force = mass * acceleration, Action = Reaction

**Rigid Bodies:** Solid that don't deform (don't exist in the real world)
Simplify dynamics of solids, useful for games
Properties (position, mass, velocity, shape, sounds, etc.)

**Rigid Bodies Update:** mass, position vector, velocity vector
Angular properties: in 2D, rigid bodies can only rotate on z axis
Angular velocity: scalar for radians per second, represented as ω (omega)
Rotational force

**Collision Detection:** Collision (two shapes intersect, distance smaller than tolerance)
Detection is computationally expensive: $O(n^2)$
Solution: 2 phases – broad and narrow phase
Broad: Find pairs of shapes potentially colliding and exclude non colliding pairs
    space partitioning coupled with bounding boxes (or volumes)
Narrow: Analyze potential colliding pairs found in broad phase (detect collision really happening)

**Strategies:** Process convex shapes only (e.g. use convex hull for concave shapes)
Test intersections: use different techniques for different colliders
    e.g. separating Axis Theorem (SAT) for convex complex shapes, distance between polygons
Continuous vs. discrete collision detection

**Separating Axis Theorem:** Idea – if a line can be drawn between two polygons, they do not collide
**Distance Between Polygons:** closest points give distance between convex polygons
If distance is zero, then overlapping (for calculating compenetrating)

**Collision Resolution:** Update game objects properties when collisions detected
Constraints: non-penetration constraints, hinge, or ball joints, etc.
Force or impulse based approaches, optimization through islands technique

## T2 – A glance to 3D

**Computer Graphics:** "graphics created using computers, and more generally, the representation and manipulation of pictorial data by a computer."

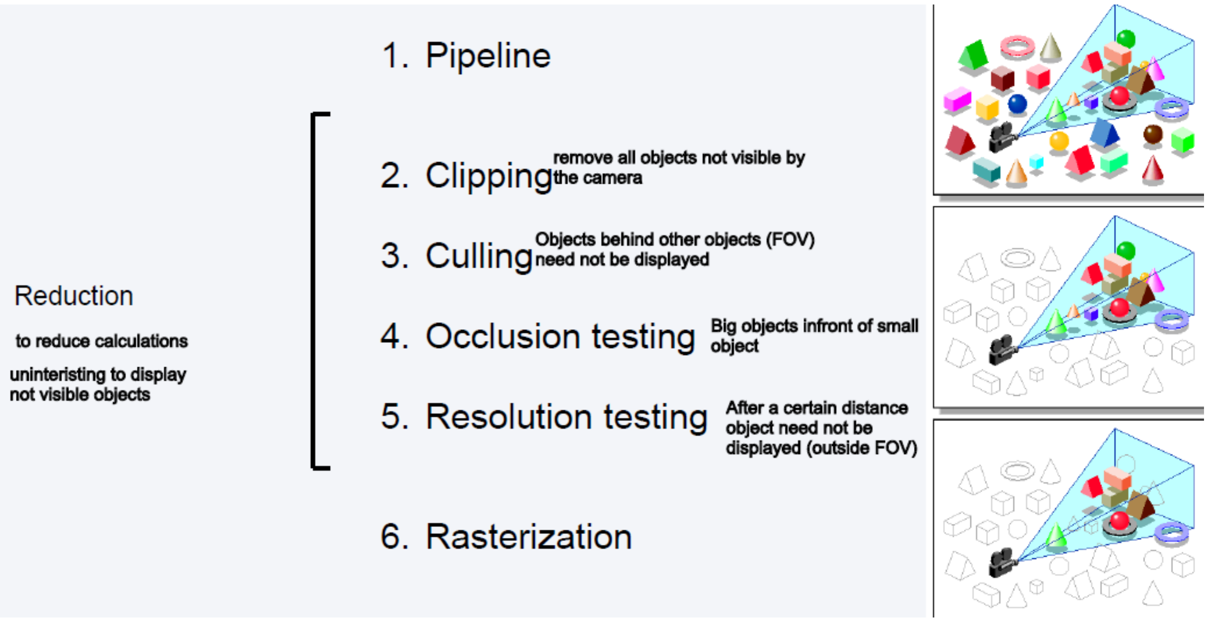
**3D Engines:** Polygonal (Hardware), Voxel
(a.k.a. Boxel) Volumetric pixel, regular grid in space, used for medical and scientific data

**3D Scene:** Elements (vertices, triangles, meshes, skeletons, textures, cameras, lights, etc.)

**Creating a Scene:** Geometry calculation, texturing, lighting, shading (pixel operations after scene creation)

**3D Algebraic Notions:** Cartesian coordinates system (vectors, points)
Matrices: Translation, rotation, scaling transform, change of coordinates system, cameras, projection
Quaternions (Solution to gimbal lock problem)

**Geometry (and Scene) Processing (Calculation):**




**Adding Materials:** Texturing is explicit (artist / realism) vs. procedural (mathematical functions)
static vs. dynamic

**Mapping:** Texture corresponding to all the polygons in the scene (XYZ, Volumes, Triangles, Tiling)
**Multiple textures:** Multipass technique (additional textures for representations; e.g. FPS → holes
Multitexturing: Environment + Glass Mapping, or Bump Mapping (cheaper than calculating everything)

**Lightning:** Ambient (environment), diffuse, specular
Light mapping: combining texture and light map for final effect

Variants: Phong (not realistic, but cheap to reproduce (ambient + diffuse + specular)
Ray-tracing + advanced (realistic but expensive)
Radiosity (simulate real-time effect of lights)
Bidirectional Reflectance Distribution Function

**Pixel Shaders:** Pixel operations and functions on our image
Programmable; Manipulation of light absorption / diffusion, texturing, reflection / refraction, shadows, primitive displacements, post processing

# T3 – Advanced SFX

**Particles Systems**: Technique to simulate effects (fire, snow, sand ,water, sparks, dust)
Animals, abstract effects

**Mathematical formalism:** describe phenomena that are dynamic, time dependent, highly parallel with small individual components, complex

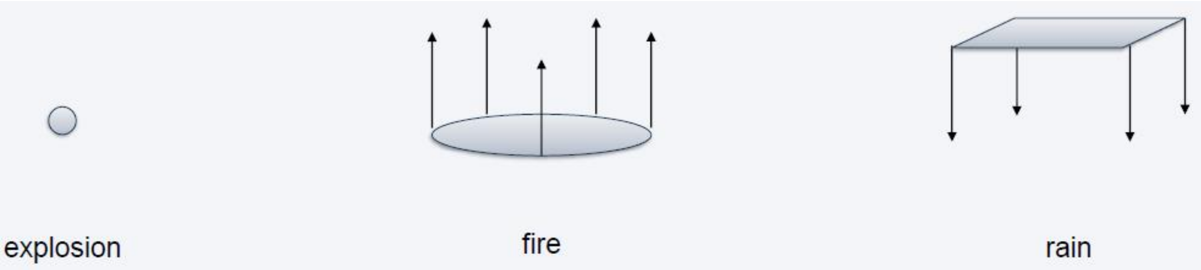
**Local:** particles all independent

**Global:** particles interact and react to each others

→ Made up of particles and the engine itself

**Particle:** specific properties (behavior, appearance)
Have a lifetime; number of parameters related with quality of simulation

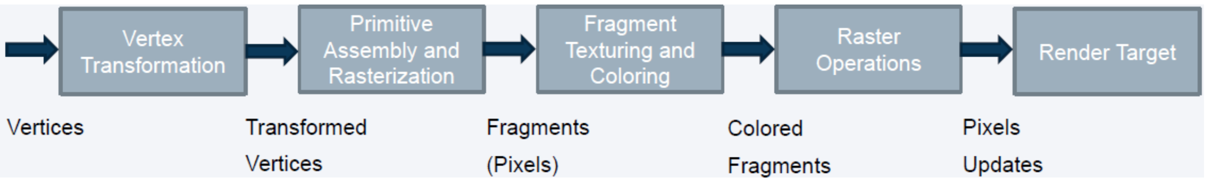
**System / engine** defines how particles are born (emitter), what simulation process



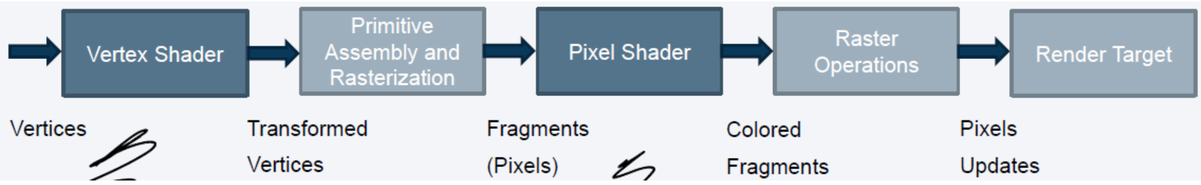

**Shaders:** act on levels of light, colors, etc. Special effects and postproduction
Mostly GPU side

**Advanced Graphic Effects:** lightning, texturing, bump mapping, normal mapping, etc.
Used predominantly in 3D, but also in 2D

**Past was fixed function pipeline (FFP):** limited set of function to configure models & textures
Made games share the same look and feel




**Today Programmable graphics pipelines:** shaders (special programs compiled and sent over GPU)




**Languages for shaders:** HLSL; GLSL, ShaderLab

**Ambient Lightning:** “Basic light in a dark room” is “Hello World” of shaders

$$AL = AL_{Intensity} * AL_{Color}$$

## T4 – AI basics

Optimal behaviour is often feels unrealistic

**Structure of an AI system:**
**AI entity (agent)**: enemies, armies, NPC, animals, etc.
4 main elements: sensor (or input system), working memory, reasoning core, action (or output) system
**Abstract controllers:** strategies, tactics, etc.; Routines for group dynamics;
structure similar to one of the entities

**Sensing (the game world):** Sensor depend on game type (computationally expensive!)

**Memory:** Individual AI (points & orientations, numerical values, states [walking, jumping, etc.])
Abstract controllers: much more complex data structures

**Reasoning Core:** Subsystem analyzing sensors and memory to take decision
Speed of decisional process depends on sensors and alternative (great games = simple methods)

**Action System:** AI coupled with action subroutines; games often exaggerate system
Perceived intelligence often enhanced

**4 common methods to simulate AI:**
state machines, rule systems, state-space searches, biology-inspired methods

**Finite State Machines (FSM):** 2 sets: set of states (AI configurations); set of transitions (conditions)
Difficult to change/update AI without graphics representation

**Parallel Automata:** complex behaviors → FSM grows cluttered and difficult to manage
define parallel subsystems (entity with more "brains) (e.g. motion, fighting, collision detection, ...)
Parallel automata works fine where behaviors are independent and simultaneous

**Synchronized FSM:** Inter-automata communication (enemies attacking in squad with different actions)
Entities share common memory area, non optimal method for strategic games with lots of entities

**Non deterministic finite automata:** FSM limitation is their predictability → dominant strategies
Variation through non-deterministic automata, i.e. controlled randomness

**Limits of FSM:** some phenomena difficult to describe in states & transitions
FSM optimal for local and sequential behaviors in nature

**Rules Systems:** Prioritized and global behaviors can be described with rule systems
RS are set of rules in the form Condition → Action

**RS Execution:** 1. Test LHS of each condition in order ("Priority List")
2. Execute RHS of first rule activated
RS imply sense of priority, as top rules have precedence over bottom rules
Global model behavior: at each execution time, all rules are tested, no sequences (FSM)

**Planning and Problem Solving:** FSM and RS ideal for simple behaviors
Sequences and phrases (FSM), Priority concurrent tasks (RS), Action AI
Complex behaviors are: Puzzles, Path finding, Games like Chess, Military strategies
AI has to "think" and "plan"

**State-Space Search:** Explore candidate transitions and analyze sustainability for reaching a goal
essentially a complement to FSM and RS
**SSS paradigm**: Propagate each possible move into future, evaluate consequences
Game represented as **tree** (nodes = configurations, Branches = moves)

**Integration of state-space search:** brute force (depth-first search with all possibilities)
Heuristic based (A*; look for promising candidate)
→ even heuristic based methods not able to solve all problems, e.g. Chess

**Biology-inspired AI:** FSM, RS & SSS based on comp. sci. & mathematical concepts
Intelligence emerging from cognitive processes
Alternative: methods simulating cognitive processes
domain just being explored, biology-inspired models are not stable

**Genetic:** based on Mendel's genetic theory
Each living being defined by DNA; every species different DNA structure
all individuals have minimal variations in species DNA
DNA of new generation is combination of parents' DNA, plus some degrees of mutation

**Evolution:** Darwin – living beings in ecosystem survive depending on suitability of DNA to environment

**Genetic programming:** Genetic and evolution well suited processes for computer simulations
Population generator: array of values represents virtual DNA (e.g. gender, race, height, hair, etc.)
add statistic variance in population

**Evolutionary Computation:** add evolution to population generator
1. generate randomly N first individuals
2. Perform fitness test on them (related to gameplay)
3. Select top P Percent
4. Generate N descendants that are: linear combinations of parents' DNA or mutations
5. go to step 2 until desired number of generations reached

**Learning Systems:** Racing games (most suitable path)
Production: Quantic Dreams → animations ("learning system)
NVIDIA RTX: AI → from low to higher resolution

**Conclusion:** techniques are currently used to develop most game AI systems
Other techniques: fuzzy logic, procedural techniques, neural networks, Bayesian networks, etc.

# T5 – Action Oriented AI

**Action:** Intelligent activity with rapid behavior changes (locomotion, attack, defense, etc.)
Contraposition to tactical reasoning: immediate vs. planned activity Simple tests

**Choreographed AI:** programmed action sequence (e.g. walking security guard)
Simple behaviors: security guards walking, shoot-em-up ships, objects (e.g. elevators)
State machines without optional transitions, often simple scripting engine

**Object tracking:** maintaining eye contact (action games!); Static or moving target (e.g. rotating turret)
Useful for: chasing, evading, navigation (reaching waypoint), etc.

**Basic Chasing:** eye contact: advance while keeping eye contact; if lost → correct orientation, keep moving
Success of technique depends on hunter's speed, target's speed, turning ability

**Predictive Chasing:** improve basic chasing through anticipating target movements (keep history of positions)
**Approach:** Calculate project position, aim at position, advance
Techniques to calculate project position are e.g. interpolation of last n positions

**Evading:** opposite of chasing; maximize distance to target
**Patrolling:** NPC (policemen, dogs, etc.)
Define set of waypoints (cycling = in order, or ping-pong = ascending, then descending order)
Following a waypoint → chasing sequence of targets

**Taking cover:** usually two actions: identify hiding spot, move quickly (e.g. chasing)
Detection spots require 3 data items: position & orientation of player
position & orientation of AI
geometry of level (e.g. map and list of objects)

**Detecting a spot:** 1. select closest object 2. shoot ray from player's position to barycenter of object
3. propagate ray and choose a point

**Platform Games AI Coding Habits:** aesthetics important to increase AI believability (sequences = personality)
different AI with simply behaviors: FSM (+ parallel automata), sequential / choreographed, chasers
end-of-level bosses ability to carry out complex choreography

**Shooters AI Habits:** slightly more complex than platform games, core behavior usually FSM
aesthetics also important
enemies think in terms of scenario and maps (follow player, take cover, act in group (shared memory)

**Fighting Games AI Habits:** Algorithms vary: quasi-random action selection or sophisticated AI
Moves can be represented in FSM (attack, block, jump, etc)
**Reactive**: learn from player's action (predictive AI) and avoid player's patterns
Keep list of last n player moves (use statistics to compute degree of independence of sequential moves)
e.g. count number of occurrences of combination kick + punch and kick + something else
**Proactive:** use for instance a space-state search with limited depth search
Create a table containing simple attacks and damage
after new attack combination, store in table distance and damage to player
afterward, choose combination that obtained max. damage for given distance

**Racing Games AI Habits:** easily implemented by mean of rule-based system
track followers Ais with additional rules (advance on slower vehicle, block way of another driver)
often pre-recorded trajectory that traverse track optimally pre-recorded and built-in

# T6 – Tactical AI

**Action:** simple methods, illusion of intelligent behaviors, sequential tests
**Tactical AI:** analysis, making "right" decision in complex scenarios
Paths, combat strategies, general solutions

**Tactical Thinking:** Tactics: sequence of operations designed to achieve a goal
States: initial state, goal state, plan to move from a state to the other
Human brain vs. machines: cognitive processes vs. numerical computation
Difficult to answer questions like "who is winning the battle"
Tactical algorithms can have side effects: too good to be realistic → frustrating for the player

**Path finding:** problem: finding path between start and an end point
Sequence of transitions (movements) between them
2 categories of path finding algorithms:
**local:** surrounding of current position, calculated step-by-step
**global:** analyze whole area, pre-calculate solution in one step & execute

**Crash & Turn:** animal behavior (local method, build solution step-by-step)
Idea: move in a straight line, if obstacle: turn left/right, try to go around obstacle
    When line of sight open, continue on in straight line
**Particularities:** either random side, or side deviating less from initial trajectory
Algorithm always finds solution if obstacles are convex and not connected

**Dijkstra:** optimal solution when geometry of world: set of vertices, weighted connections (distances)
Popular in FPS, global algorithm, optimal to find destinations, but not to trace a path

**Application of algorithms depends on your game; balancing is fundamental**

**A*:** state-space search algorithm (real-time strategy games)
Path-finding method for chessboard, States (location), transitions (unitary movements)
Evaluates alternative from best to worst, convergence to best solution
**Idea:** Init: Base Node; Destination Node; Set of Movement Rules left/right/up/down)
    at each step compute possible movements until: all tried or reach destination
    !Necessary to evaluate how each node is good (explore better paths first)
**Rating Nodes:** see slide 18
**Conclusion:** most widely used path finding algorithm
**Problems:** precompute whole path in one step (unusable with dynamic geometries, fog-of-war?)
    A* always produces unrealistic "optimal" results
    Memory problems!
**Improvements:** region-based A* (e.g. rooms connected by edges)
    interactive-deepening A* (compute whole path in small pieces)

**BOIDS:** algorithm designed to create movie special effects (e.g. The Lion King)
**Groups behavior:** BOIDS are real-time simulation of human & animal groups
    Core hypothesis: behavior of group governed by small set of simple rules
**The 3 Flocking behavior rules**: separation; alignment; cohesion

**Separation:** Avoid collision in group (distance threshold between members)
Distance lower than threshold, both members change their orientation
**Alignment:** all members will aim in the same direction
**Cohesion:** all members try to stay close to barycenter of group (respecting separation rule)

**BOIDS Implementation:** do not need internal state or working memory
One of the member is AI, others will adapt to the leader
Algorithms based on attraction and repulsion laws
Possible to add additional rules: simulate two populations; obstacles can also be BOIDS

**Formation-based movements:** Human groups dynamic simulated with summations of fields
Military formation: 1 field separating units (repulsive)
1 field keeping position in formation (attractive)
1 field detecting collisions with obstacles (repulsive)

**Influence maps:** data structures useful to represent balances of powers, frontlines, etc.
Simple to consult; dynamic

**Influence maps Usage:** creation of map for 2 armies (matrix)
+1 represent soldiers of first army
-1 represent units of second army
empty cells contain interpolated values
Size of the influence map can be smaller than size of world
Influence map can also map several values
Useful tests: frontlines between armies are extracted from zero values
sum of all values indicates winning army
other tests include breakability, weakest enemy, etc.