FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Games

Human-Computer Interaction Lecture

Slides adapted from the Game Design Workshop by V. Schwind, Image from: https://pxhere.com/de/photo/764632

Learning Goals

- Understand …
  - what a games is and why video games are important subject in HCI
  - game types and classifications
  - principles of a meaningful game
  - the player-centered development process
  - gameplay and flow mechanics
  - technology-related questions
- Be able to explain …
  - game principles and game development processes
  - game types and technologies

WHAT IS A GAME?

“Games are the voluntary attempt to overcome unnecessary obstacles”.

- Bernard Suits [42]

What is a Game?

- An interactive **piece of entertainment**
- Structured form of play and governed by **rules**
- **Uncertain**, **unforeseeable**, **quantified** outcome
- Circumscribed in **time** and **“place”**
- Undertaken for **fun** and **non-productive**
  - Exceptions: serious games, gamification
- Accompanied by the awareness of being in **another reality**
  - Simulating **an artificial conflict**
  - Solving **an artificial problem**


WHY DO PEOPLE PLAY GAMES?

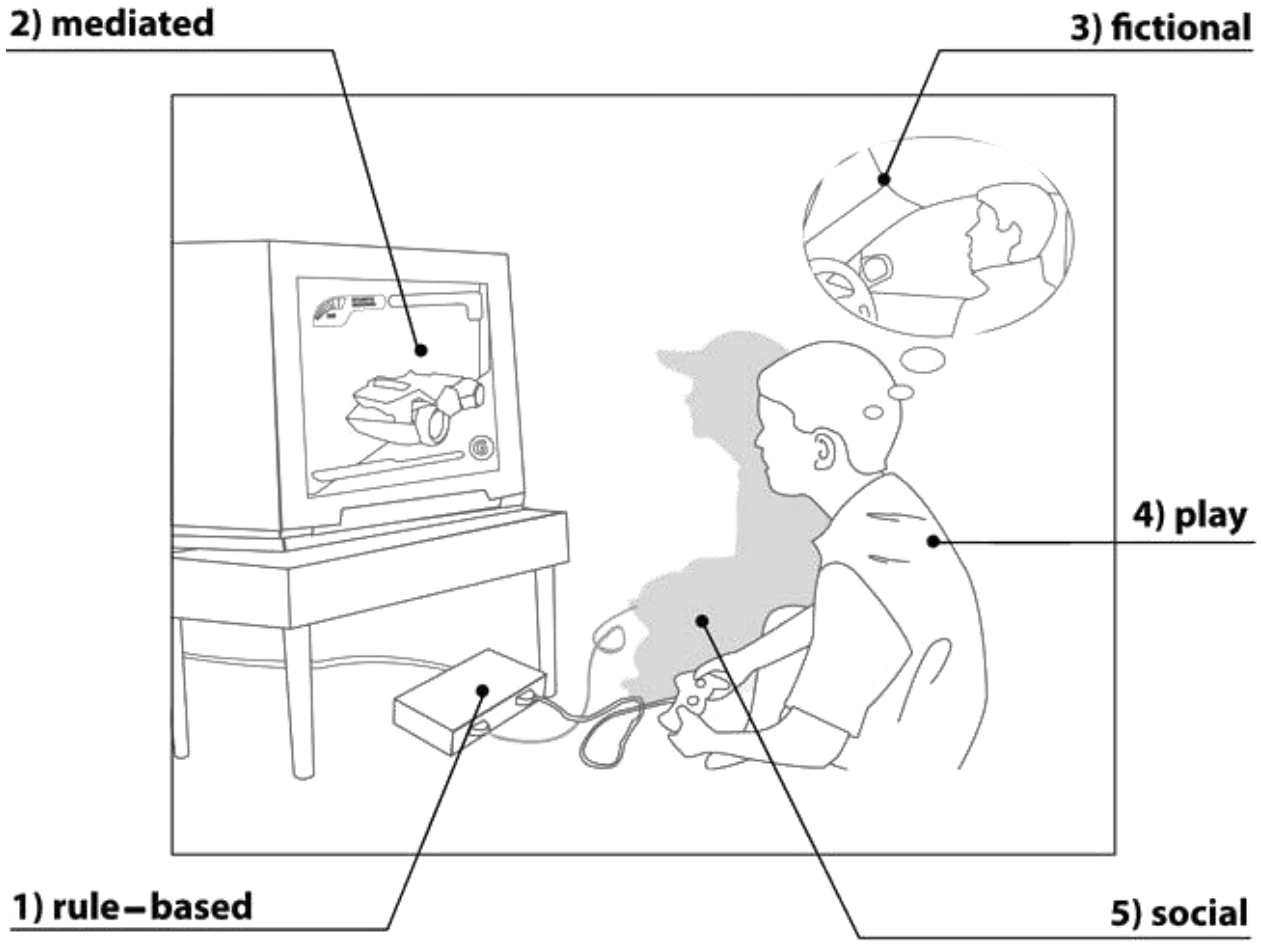
Game Classfication

- **Cooperation**
  - Cooperative or non-cooperative (alone or together)
- **Symmetry**
  - Symmetric or asymmetric (identity based payoff or all equal)
- **Sum of Choice**
  - Zero-sum / Non-zero sum or constant-sum (+ sum of all wins - sum of all losses = 0)
- **Simultaneous / Sequential (Dynamic)**
  - Moves / Prior Knowledge / Time Axis
- **Information**
  - Perfect (all moves are known)
  - Imperfect (some moves are unknown)

Video Game Classfication

View	Type	Genre
Side-Scroller	Action	Fantasy
First-Person / Ego	Adventure	Science-Fiction
Third-Person	Role-Play-Game	Sport
Top-Down	Puzzle	Party
2D	Party	Medieval
3D	Racing	Horror
...	...	...

Five Planes



M. Nietsche, Video game spaces: image, play, and structure in 3D worlds. MIT Press. (2008)


HOW TO CREATE A GAME?

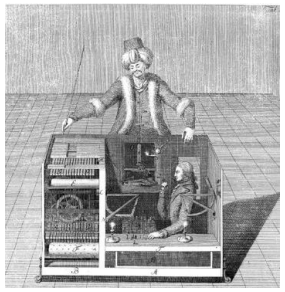
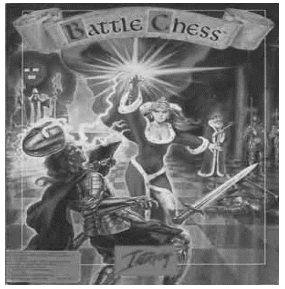
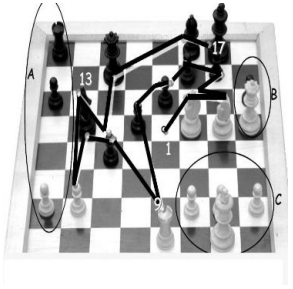
Who makes a game?

„they“

Game Producer

Project Manager

Technical Director(s)

Game Design	Developer	Art	Visual	Sound	Quality Assurance
					

Public Relations

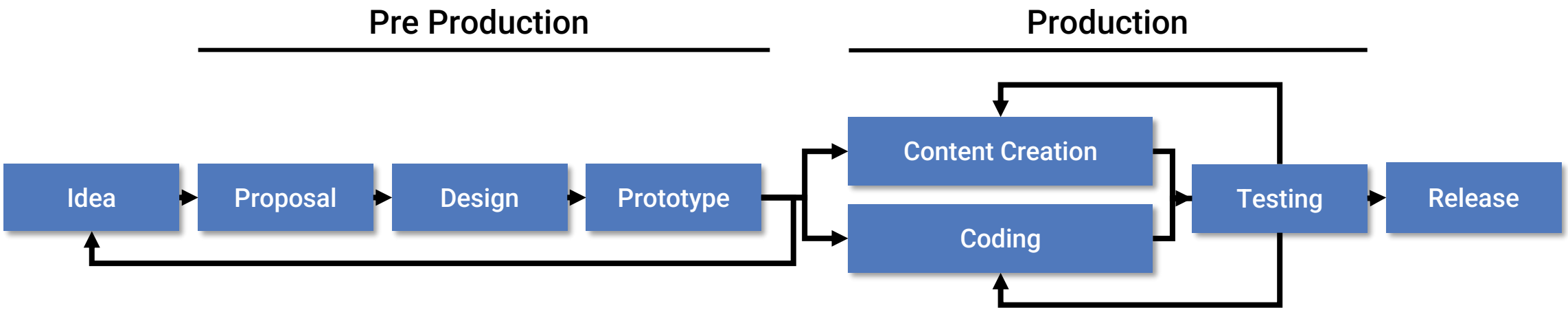
Product Designer

Community Manager

People who obtain the infrastructure

Image references from left to right: [6,7,8,9,10,11]

Development Process



Game Design

- A **designer** creates a **context** to be encountered by a **player**, from which **meaning** emerges [1].
- Meaning is an interpretation of context based on
  - **Semiotics**: The study of signs and symbols and their interpretation
  - **System**: A set of interconnected components working together to achieve a common goal or purpose
  - **Interactivity**: The ability of a system or technology to respond to user input or actions in a meaningful way
  - **Choice**: The act of selecting or making a decision between two or more options

[1] Peter Shankar - CSE 497 – Topics on AI & Computer Game Programming

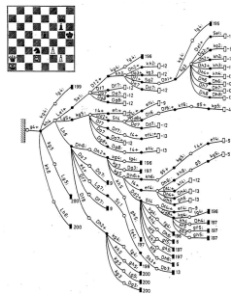
Meaning in Games

Semiotic

System

Interactivity

Choice



- A **symbol (or icon)** represents something other than itself
- Symbols are **interpreted**
- **Meaning** results when a symbol is interpreted
- **Context** shapes interpretation

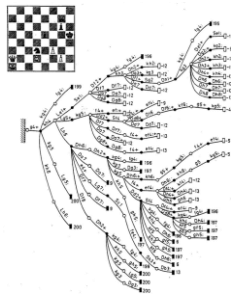
Meaning in Games

Semiotic

System

Interactivity

Choice



- **Objects** are the parts, elements, or variables within the system
- **Attributes** are qualities or properties of the system and its objects
- **Internal** relationships are relations among the objects
- **Environment** is the context that surrounds the system

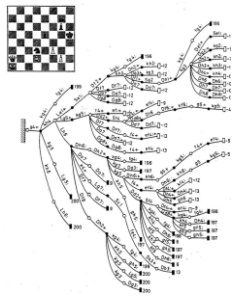
Meaning in Games

Semiotic

System

Interactivity

Choice



- **Cognitive**: interpretive participation
- **Functional**: utilitarian participation
- **Explicit**: participation with designed choices and procedures
- **Beyond-the-object**: participation within the culture of the project

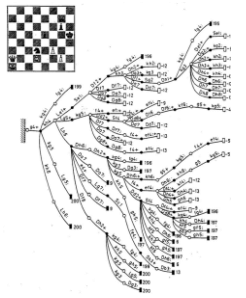
Meaning in Games

Semiotic

System

Interactivity

Choice

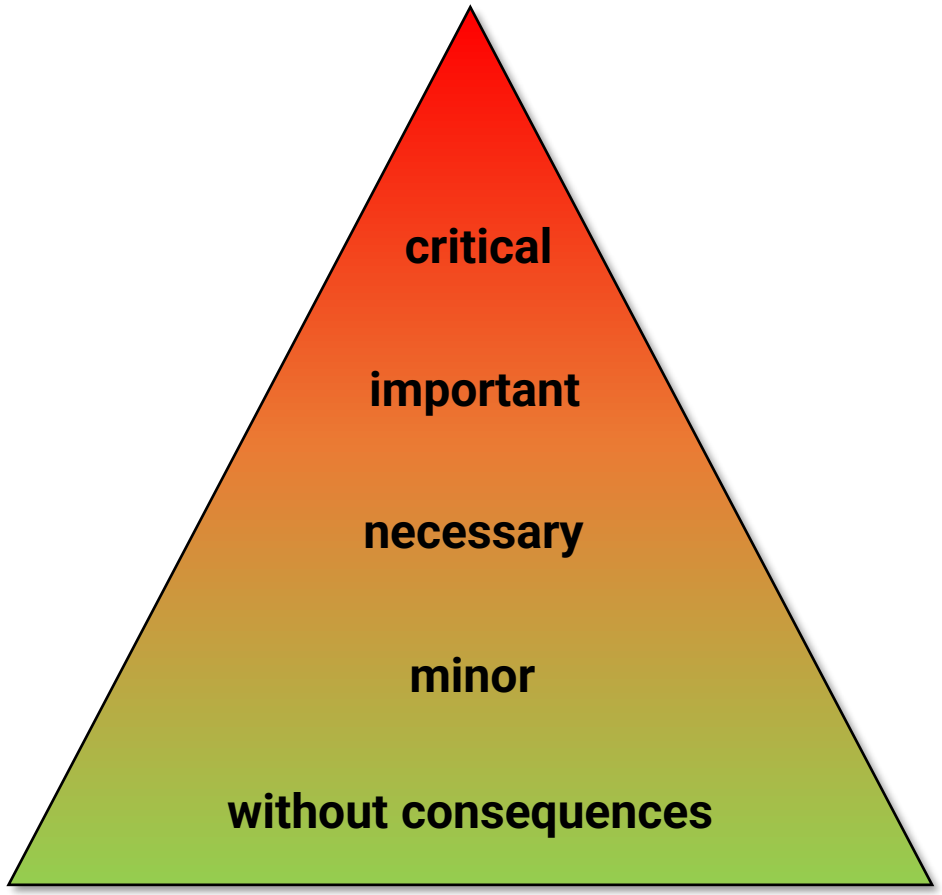


- Choices at **micro level**: each decision at it’s smallest level
- Choices at **macro level**: aggregated choices form a larger outcome
- A cluster of choices is a player’s **tactic** (local planning)
- The sum of choices is a player's **strategy** (global planning)
- The **outcome** also depends on the action of others.

Choices in Games

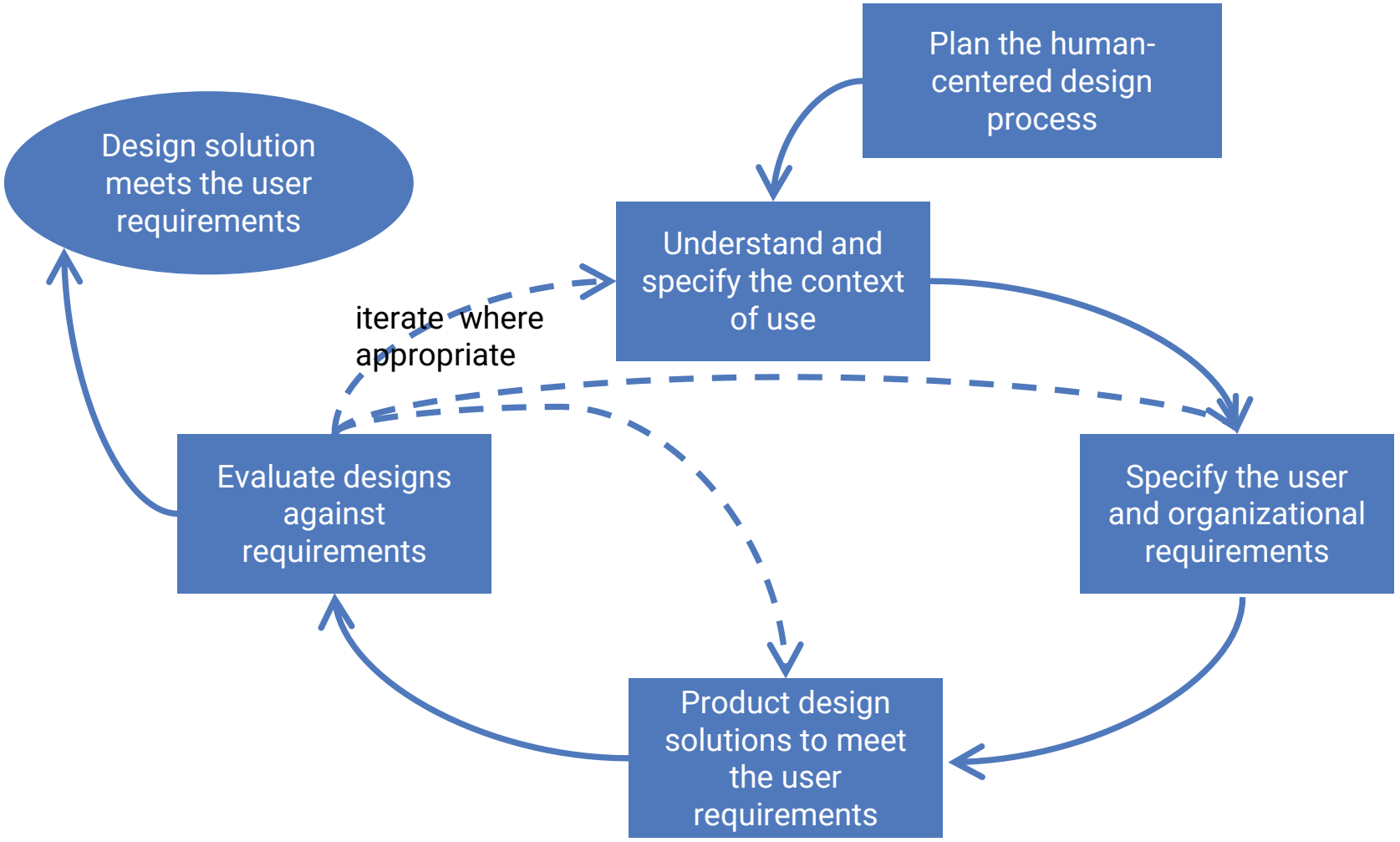
- A choice is a **non-trivial, two-sided question** to the player
  - **upside**, one step closer to victory
  - **downside**, hurts the player's chances of winning
- **A choice has consequences**
  - The player **can not go back** after exploring the consequences
  - Simple **risk** management: low risk / low rewards – high risk / high reward
  - Avoid **dominant strategies**
  - Avoid **trivial choices**
- The player must be **aware** of making a choice.
- Choices have **consequences**

- life or death
- direct and immediate impact
- indirect or delayed impact
- small impact, direct as well as indirect
- no impact, no outcome

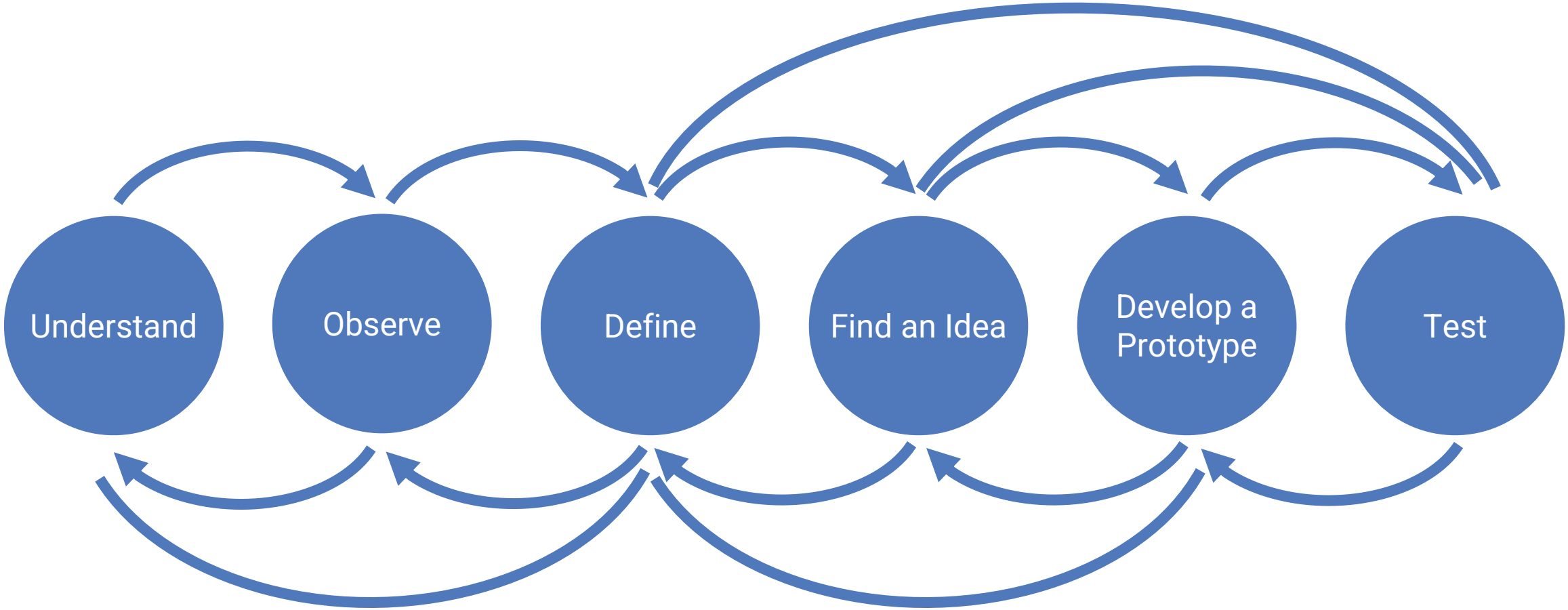


Is the **illusion** of having a choice
more meaningful than the choice itself?

~~Human~~ Player-centered Game Design



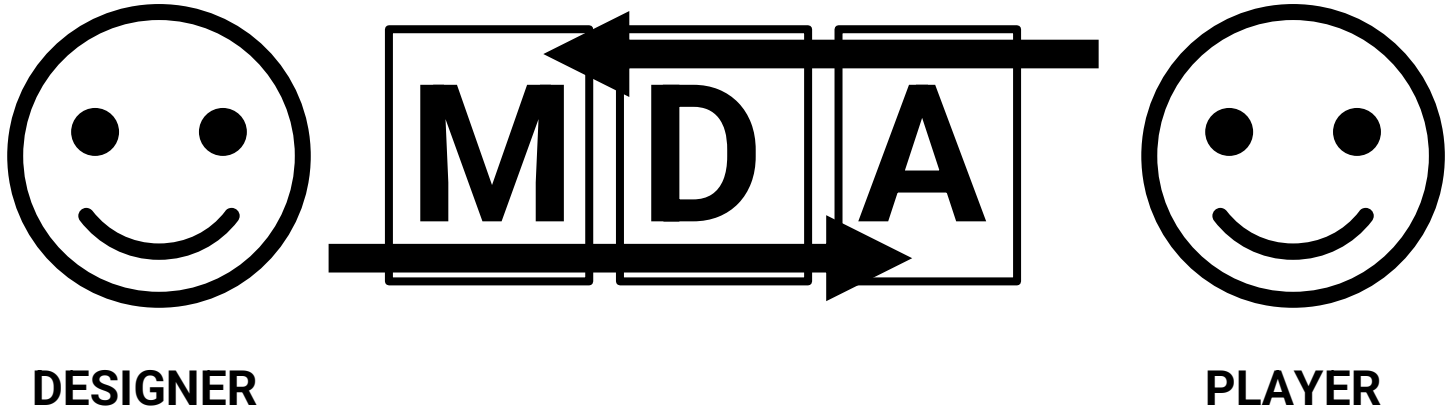
Design Thinking



The Design Thinking process, referring to an Anja Wölbling, Kira Krämer, Clemens N. Buss, Katrin Dribbisch, Peter LoBue, and Abraham Taherivand 2012. “Design Thinking: An Innovative Concept for Developing User-Centered Software”, in Software for People, Mädche, Alexander (eds.), Berlin: Springer, pp. 121ff.

MDA Framework

- **Mechanics**: Rules and algorithms define the actions
- **Dynamics**: Behavior arising while players interact
- **Aesthetics**: Visual qualities, Experiences, emotions



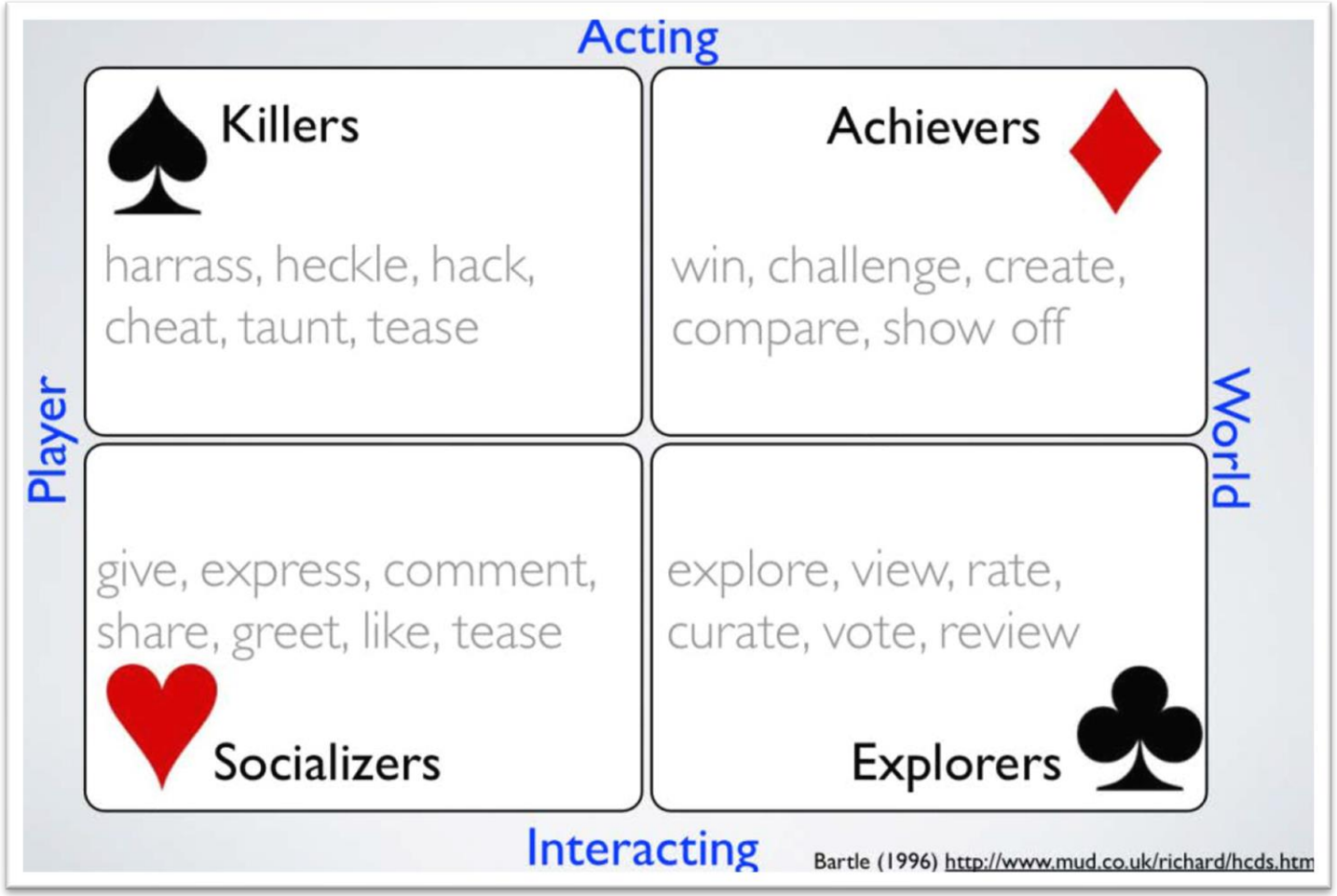
Players‘ Demographics

- 0-3 **Infant/Toddler**: Interested in Toys. The target group in this age are the parents!
- 4-6 **Preschooler**: First interest in simple video games
- 7-9 **Kids**: Age of reason, Childrens are able to solve complex problems
- 10-13 **Pre-Teen**: Age of obsession, neurological growth with the ability to think deeply
- 13-18 **Teen**: First gap - boys interested in competition / girls on real-world issues
- 18-24 **Young Adult**: The main game consumer group: have both time and money
- 25-35 **Twenties and Thirties**: Family formation and hardcore gamers
- 35-50 **Thirties and Forties**: decisions about expensive games (for children or themselves)
- 50+ **Fifties and Up**: Empty nesters have a lot of time, children's have moved out

Players‘ Demographics

- **Gender Gaps**
  - Males: Mastery, Action, Competition, Destruction, Spatial Puzzles, Trial & Error
  - Females: Emotion, Link to Real World, Dialogues & Verbal Puzzles, Learning by Example
- **Cultural Differences**
  - Eastern: Strategic affinity, prefer young figures, female protagonists, androgyny, colors
  - Europe: High diversity, lowest religious affinity for video games.
  - Western: Patriotic, but strong restriction of sexual and homosexual representations
    - Many subcultures
  - India: High affinity for polytheism religious content.

Bartle‘s Taxonomy of Players



Players‘ Expectations

Create what players expect…

- Meaning
- Consistence
- Tasks
- Immersion
- Solutions
- Challenges
- Rewards
- Aesthetics

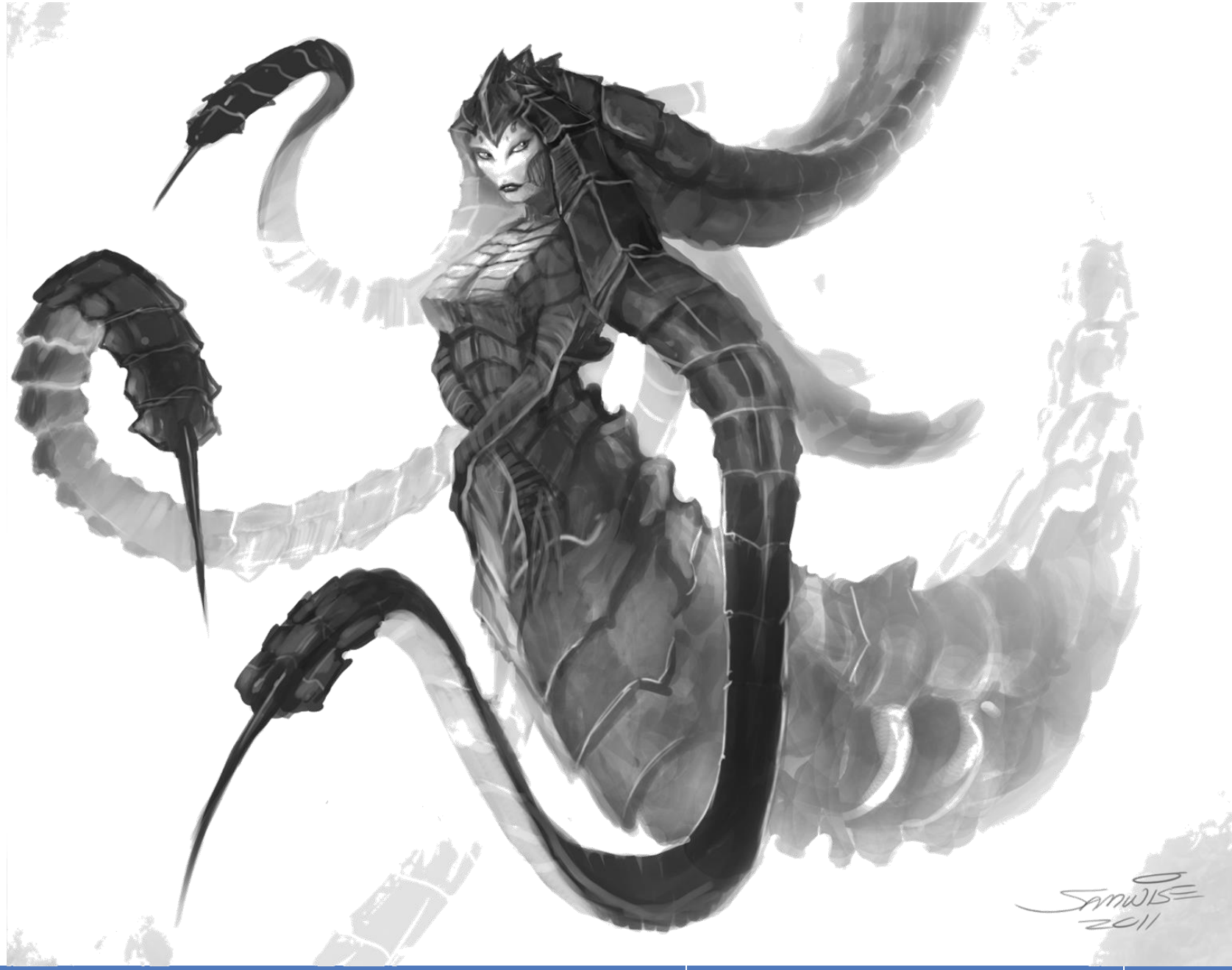
… and what not!

- Dynamics
- Curiosities
- Surprisings
- Changing game states
- Increase the pace
- Details
- Limits
- Subgames


But what players really want is an experience!

Create…

- listen
- observe
- consider
- analyze
- dissect
- select
- abstract
- recombine
- mutate
- …napkins!



…a game…

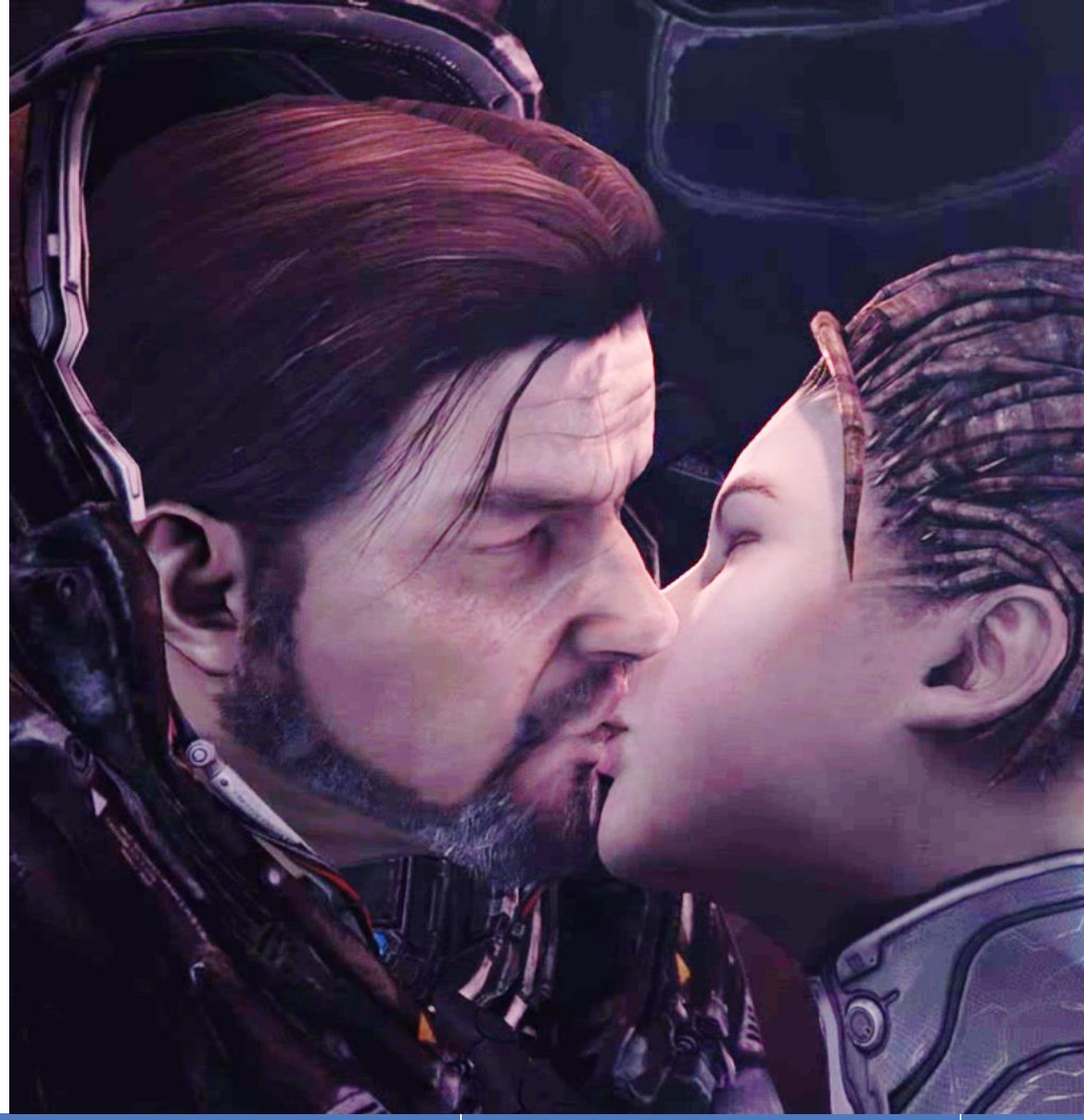
- Goals
- Constraints
- Mechanics
- Obstacles
- Rules
- Rewards
- Interfaces
- Styles

…with a meaningful…

- Protagonist
- Challenge
- Conflict
- Circumstance
- Location
- Age
- World
- Universe

…plot…

- Overcoming the monster
- Rags to riches
- Quest
- Voyage
- Comedy
- Tragedy
- Rebirth
- Romance



…to convey…

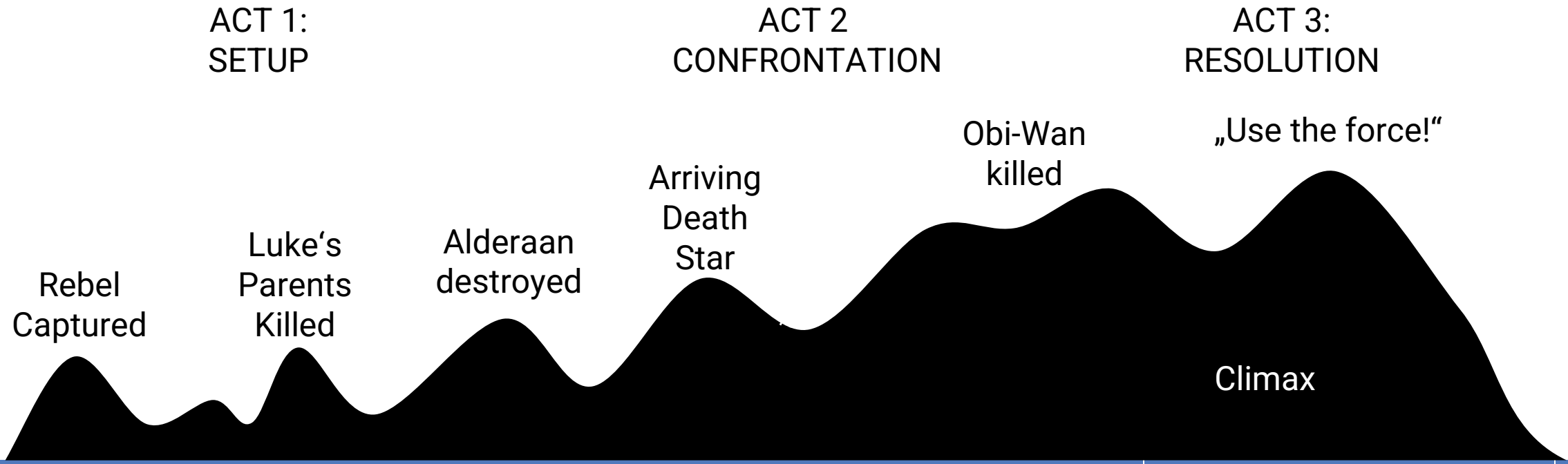
- Impression
- Perspective
- Learning
- Sensation
- Stimulation
- Reaction
- Feedback

… an experience!

- Internal Value
- Satisfaction
- Membership
- Community
- Message
- Thrill
- Immersion
- Transformation
- Rethinking
- or in other words….

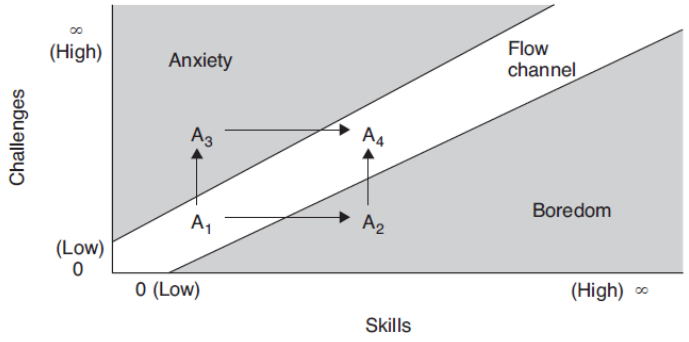
The Plot

- Experiences shouldn‘t be linear
- People get used to external stimuli and require changes and more stimuli to be as satisfied as before
- Alternations between action and resting

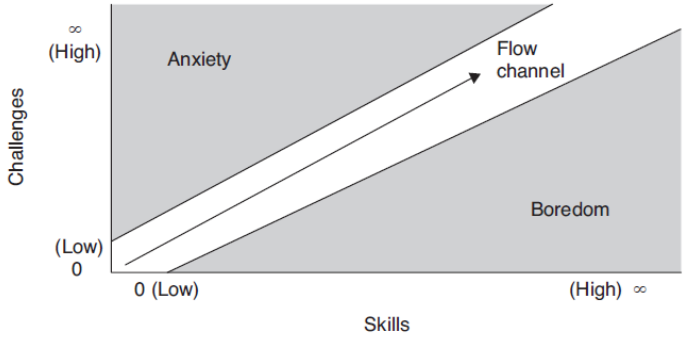


The Flow

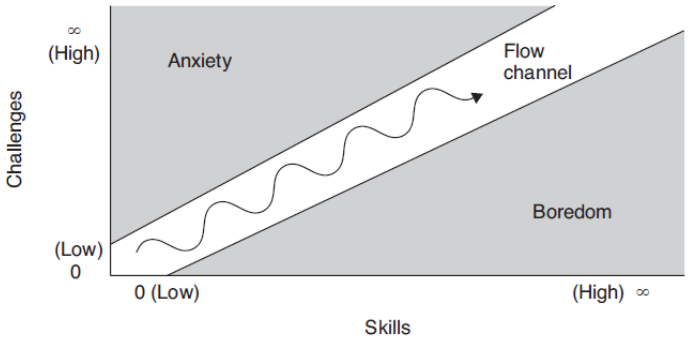
- Clear goals
- No distractions
- Direct feedback
- Continuously challenge



Distractions, no flow



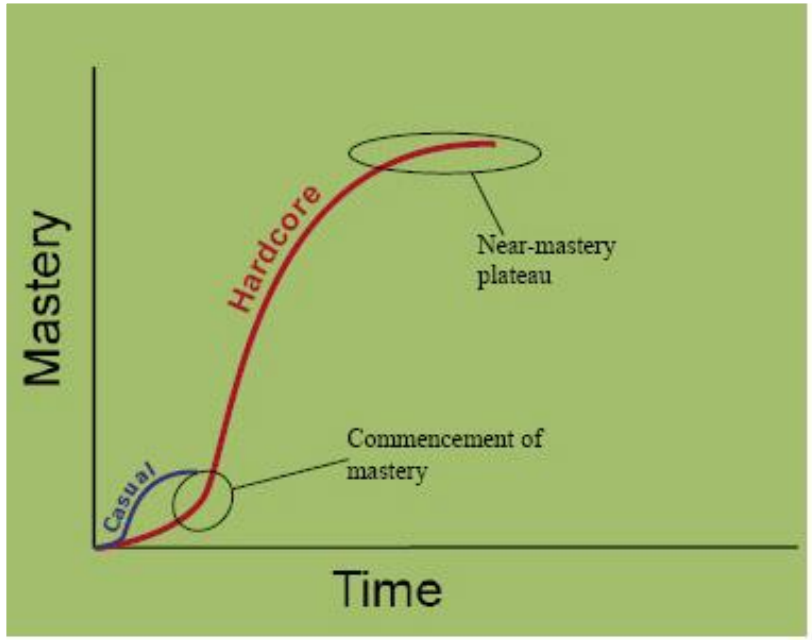
Flow, but too linear “run”



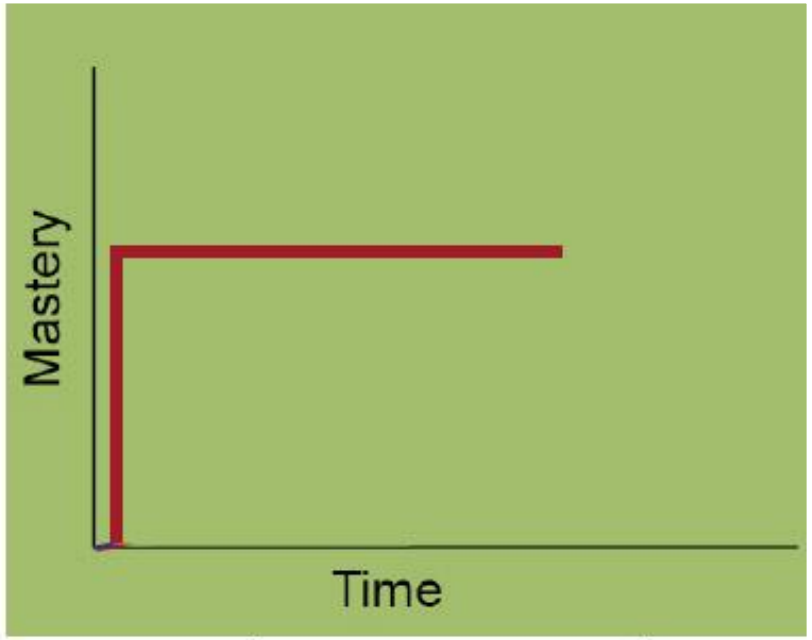
Most interesting flow

The Art of Game Design: A Book of Lenses, Second Edition, Jesse Schell, CRC Press, 2014, ISBN 9781466598645

The Learning Curve



Casual Game



Flappy Bird

Gameplay Variations

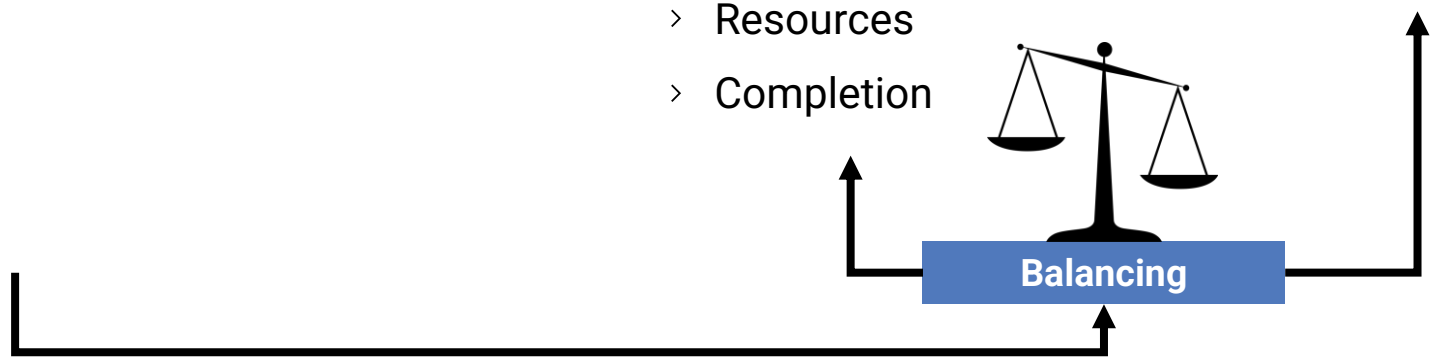
- **Asymmetric**: Different Players can play the same game simultaneously in a different way (e.g. Rayman Legends for Wii U GamePad).
- **Asynchronous**: Players can play the same game at different times.
- **Cooperative** :A gameplay feature that allows players to work together as team to reach a goal (e.g. Gears of War)
- **Deathmatch**: Kill the other player until a certain condition (e.g. Quake).
- **Emergent**: Complex simulations in video games with relatively simple game machanics (e.g. The Sims).
- **Hack and slash**: The usage specifically implies a focus on combat with hand-to-hand weapons (e.g. Diablo)
- **Leveled**: The process of automatically changing parameters, scenarios, and behaviors based on the player's skills. (e.g. Homeworld).
- **Micromanagement**: Describes detailed gameplay elements that must be manually addressed by the player (e.g. Anno).
- **Nonlinear**: Multiple sequences to finish the game, a choice between paths to victory, or optional side-quests and subplots (e.g. Mass Effect).
- **Twitch**: Tests a player's reaction time. Keeps players actively engaged with quick feedback to their actions (e.g. Quake III).

Motivation & Playability

- **Intrinsic:** Game rules, goals, objectives, rhythm and other design mechanics.
- **Mechanical:** Quality as a software system. Fluency of the movie scenes, correct lights, shadows and rendering, sound and music, graphics motions, character personality implementation and communication systems.
- **Interactive:** Player interaction and video game user interface development, for example interaction dialog and game controls. This playability is easily visible in the Game Interface.
- **Artistic:** Arts and aesthetics in the game elements: visual graphics, sound effects, music and melodies, storyline and storytelling.
- **Personal:** Individual vision, perception, and feelings that the video game produces in each player when they play the game. It has a highly subjective value.
- **Social:** The group consciousness and different user perceptions when the player plays with other player in a competitive, cooperative or collaborative way.

Balancing

- **Mechanics**
  - Turns / Time-keeping systems
  - Action points
  - Bidding / Auction
  - Cards
  - Capture / Eliminate
  - Catch-up
  - Movement
  - Resource Management
  - Role-playing
  - Tile-laying
  - Worker Placements
  - Game Modes
- **Rewards**
  - Praise
  - Points
  - Prolonged Play
  - A Gateway
  - Spectacle
  - Expression
  - Powers
  - Resources
  - Completion
- **Punishments**
  - Shaming
  - Loss of points
  - Shortened Play
  - Terminated Play
  - Setback
  - Removal Powers
  - Resource Depletion



Rules & Goals

- **Rules**
  - What are the **foundational** rules of my game?
  - What are the **operational** rules of my game?
  - **Who** enforces the rules?
  - Are the rules **easy** to understand, or is there **confusion** about them?
  - If there is **confusion**, should I fix it by changing the rules or by explaining them more clearly?
  - Are there "**laws**" or "**house rules**" that are forming as the game develops?
- **Goals**
  - What is the ultimate **goal** of my game?
  - Is that goal **clear** to players?
  - Is there a **series** of goals?
  - Are the **different** goals related to each other?
  - Are they **concrete**, **achievable**, and **rewarding**?
  - Do I have a good **balance** of short- and long-term goals?
  - Do players have a **chance** to decide on their own goals?

Note: Only the rules lead to the goals….

… and the rules have to cover all circumstances!

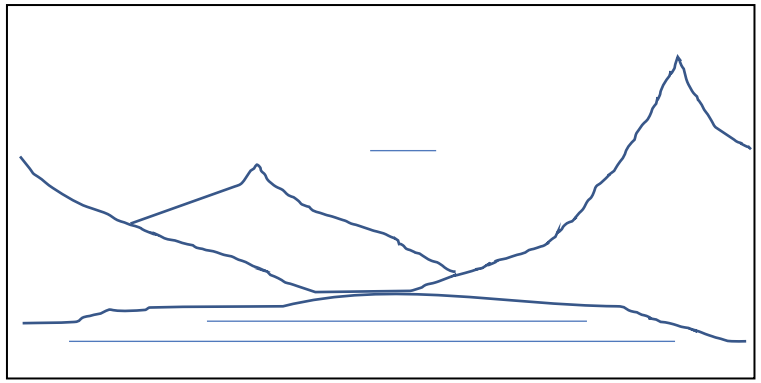
Gameplay & Game Mechanics

- Gameplay ≠ Game Mechanics
- “Gameplay is the specific way in which players interact with a game.” Wikipedia.org
- “A series of interesting choices.” Sid Meier
- “One or more causally linked series of challenges in a simulated environment.” Andrew Rollings und Ernest Adams

Game Art

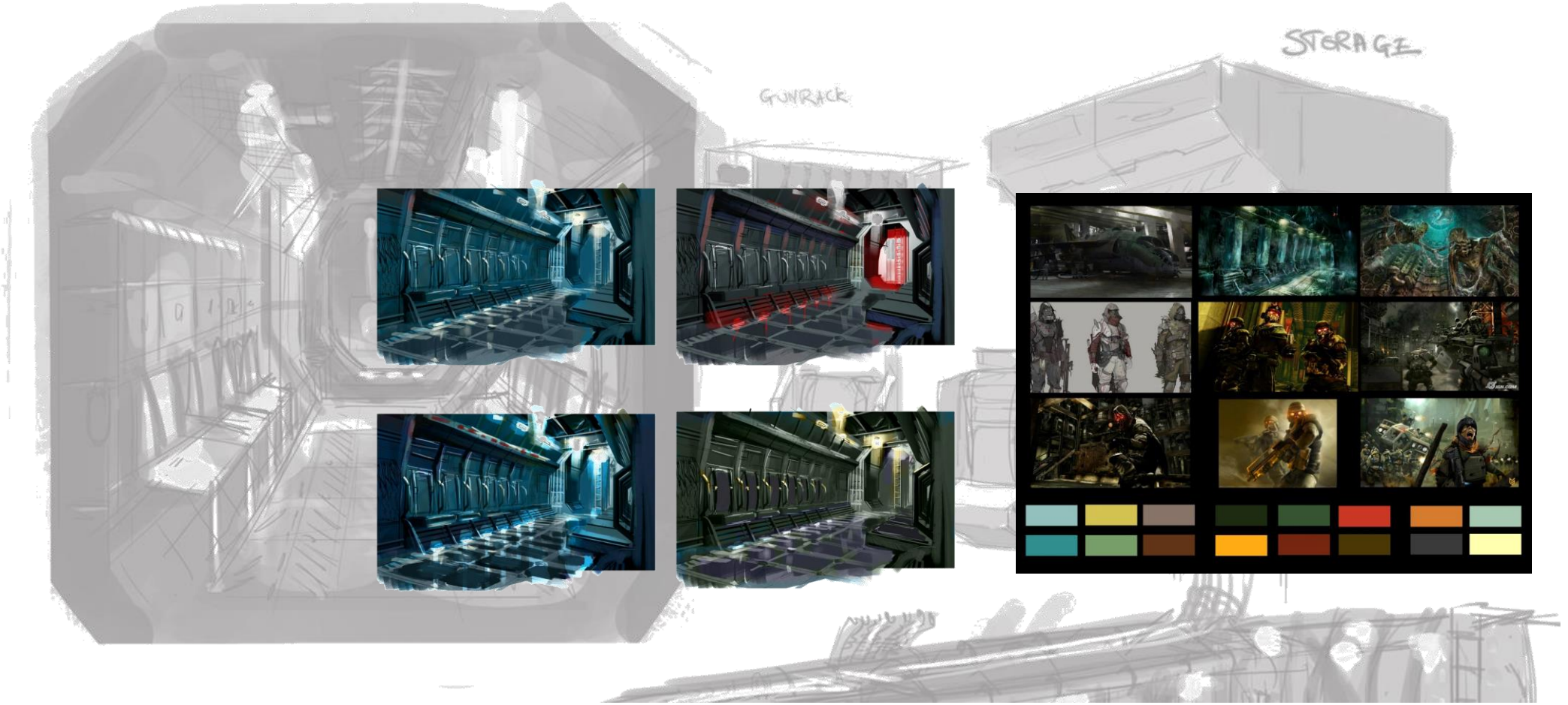
Aesthetics

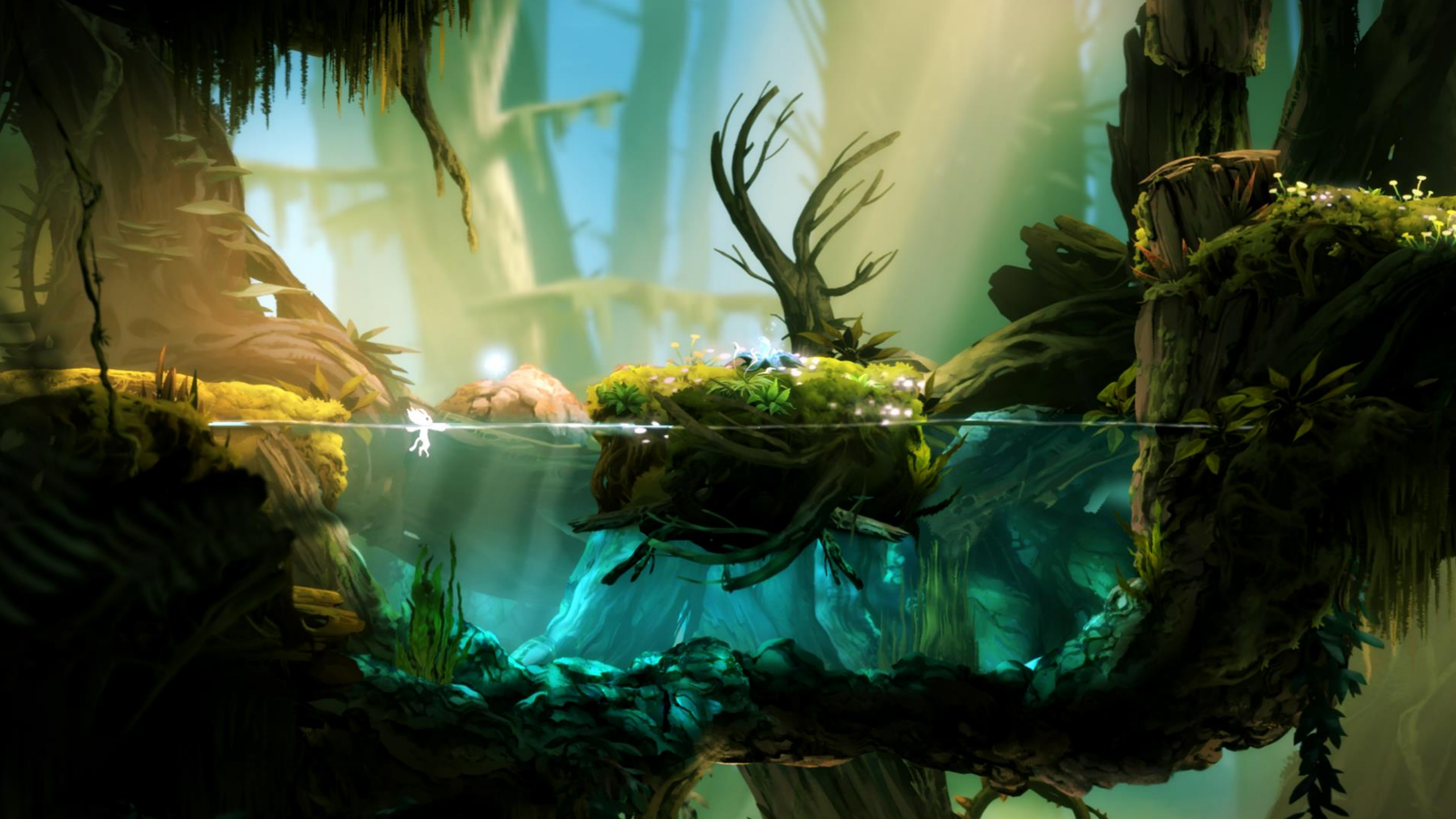
- Look and Feel of a game is defined by its **style guide**.
- Documents are abstract, **illustrations are concrete**.
- A well-rendered image of a good idea is compelling in a way that few people can resist (but also increases expectations).



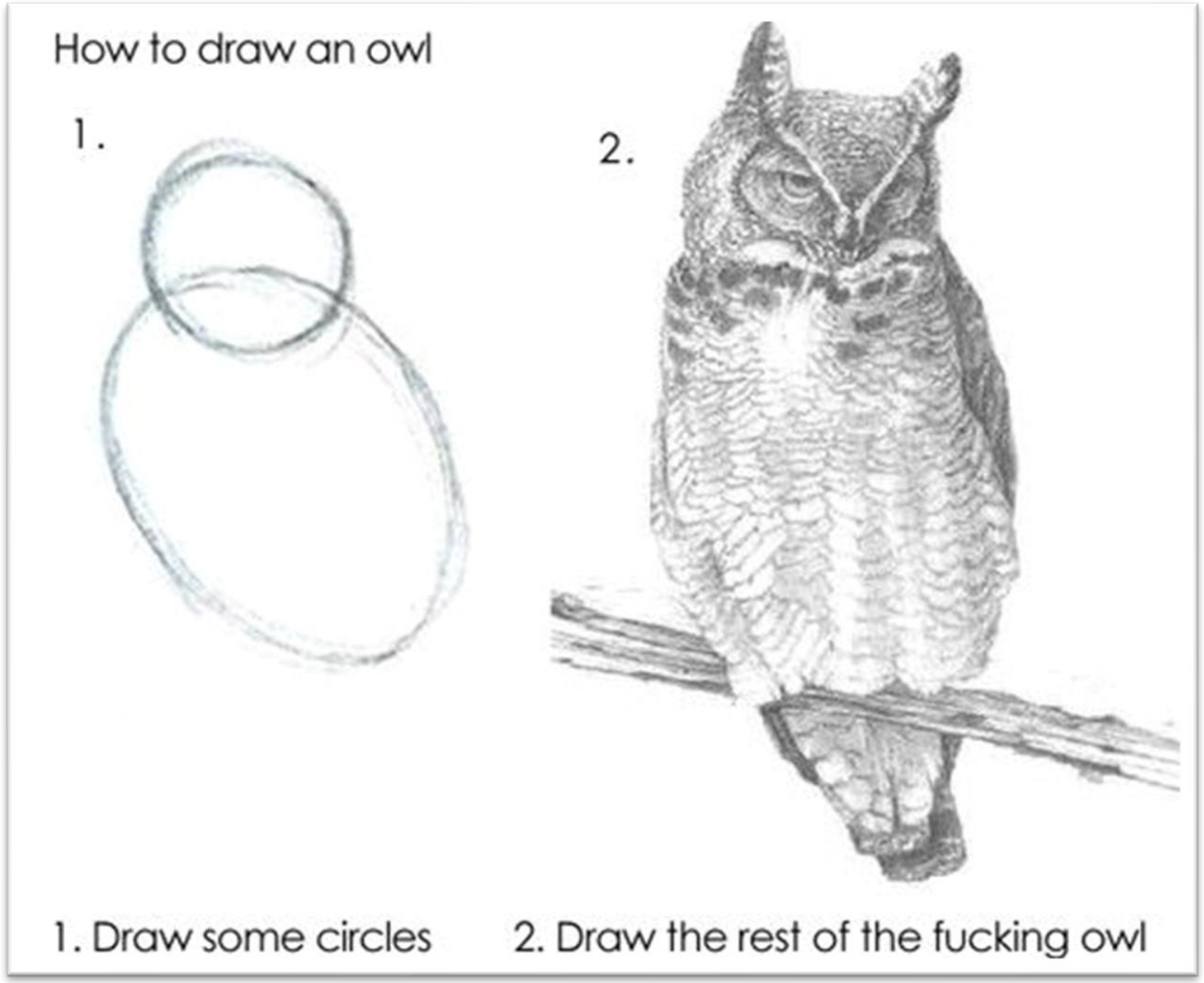
Microsoft Flight Simulator X: DirectX 10-Screenshots, 2011, http://www.pcgames.de/Flight-Simulator-X-PC-123220/News/Microsoft-Flight-Simulator-X-Erste-grandiose-DirectX-10-Screenshots-veroeffentlicht-PC-Games-vor-5-Jahren-841187/

Concepts and Moodboard





Drawing


How to draw an owl
1.
2.
1. Draw some circles
2. Draw the rest of the fucking owl

Realism

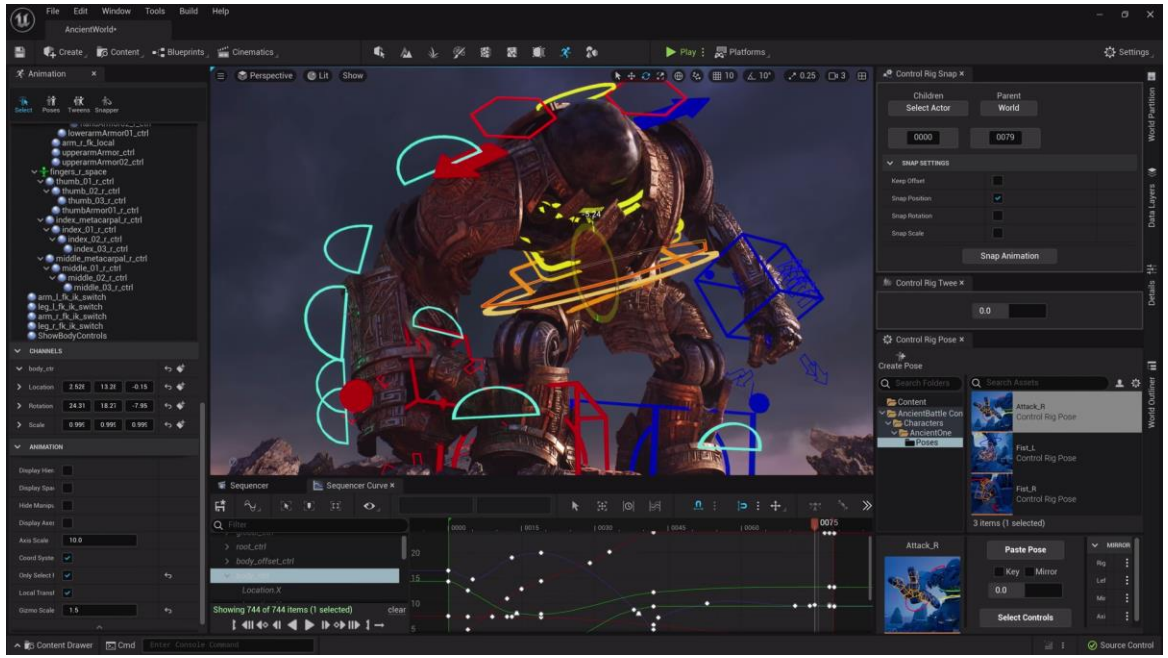
Technology

Game Technologies

- **Hardware**
  - PC, [x] GHz, [y] RAM, [z] graphics card…
  - Playstation [x]
  - Xbox [x]
  - …
- **Software**
  - Unity3D
  - Unreal Engine
  - CryEngine
  - LibGDX
  - OGRE
  - RED
  - ….

Game Engines

Unity Game Engine (C#)

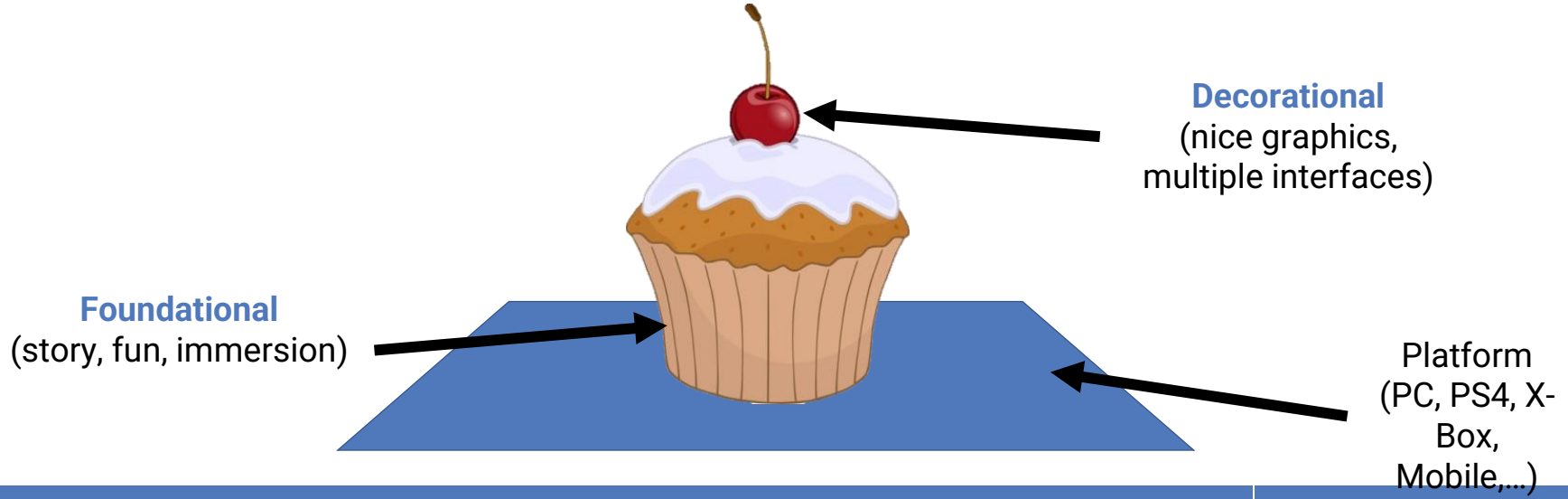


Unreal Game Engine (Blueprint, C++)

https://www.dice.com/career-advice/how-unity3d-become-a-game-development-beast
https://beforesandafters.com/2021/05/27/you-can-get-early-access-to-unreal-engine-5-now/

Technologies/Features

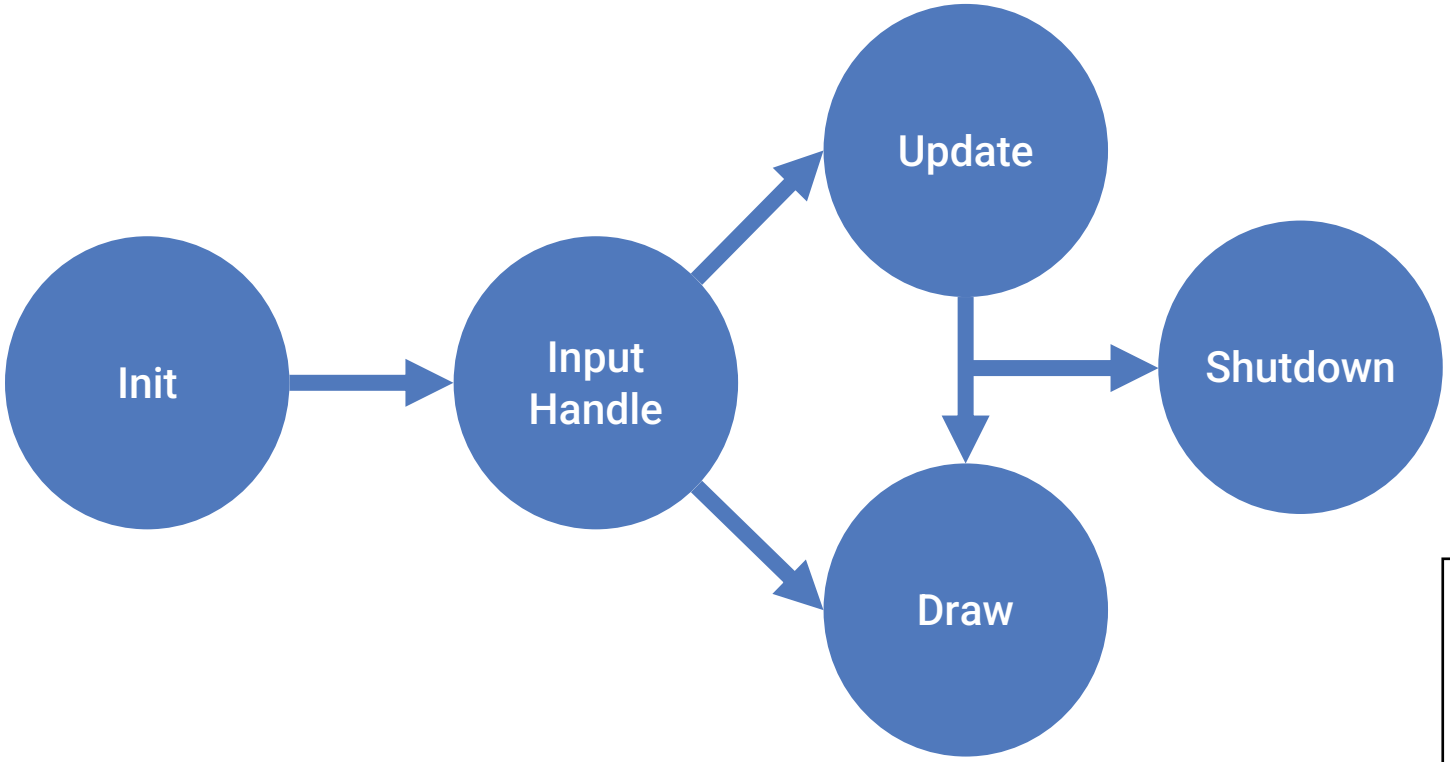
1. What technologies will help deliver the experience I want to create?
2. Am I using these technologies in ways that are foundational or decorational?
3. If I'm not using them foundationally, should I be using them at all?
4. Is this technology as cool as I think it is?
5. Is there a "disruptive technology " I should consider instead?



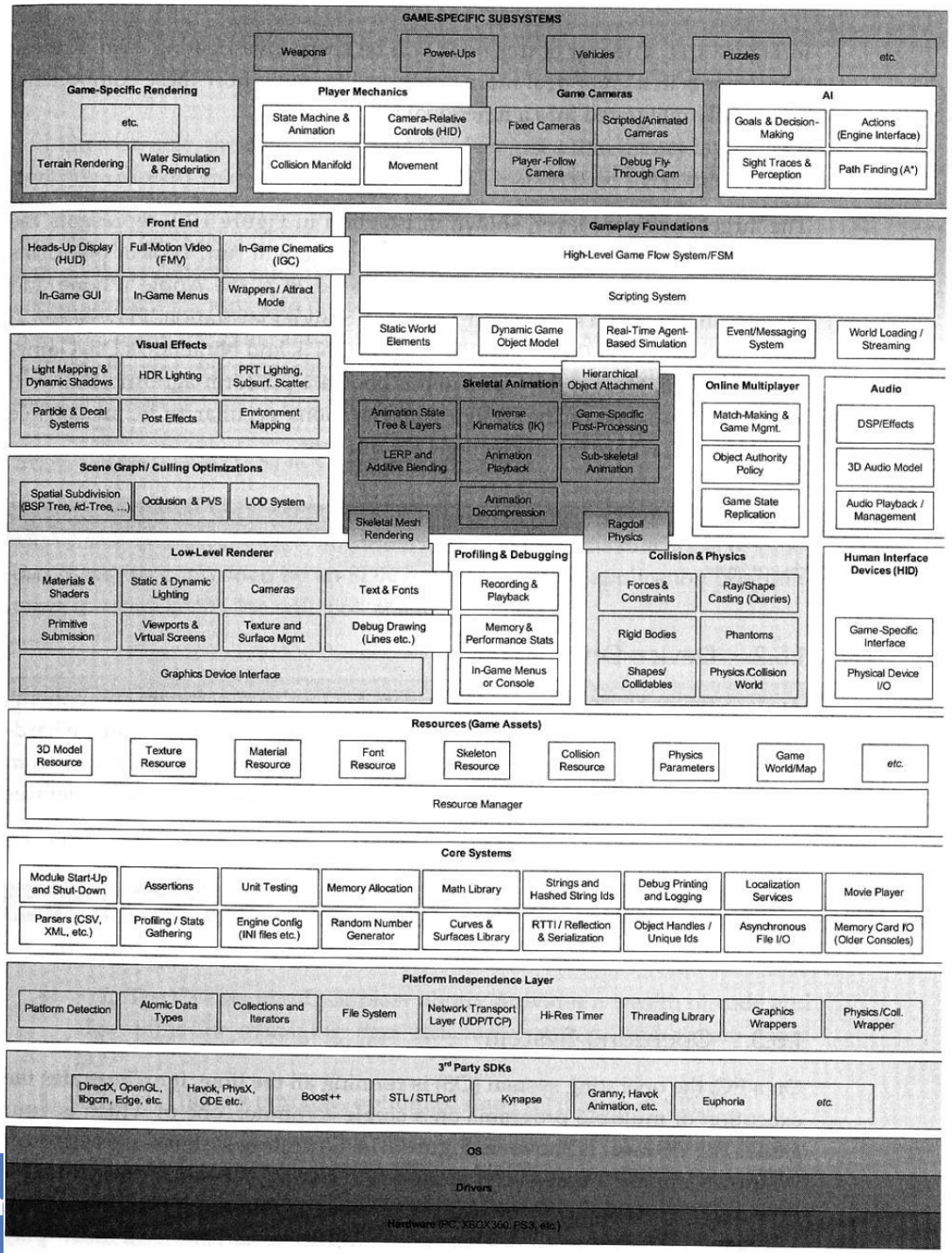
Technology in Games

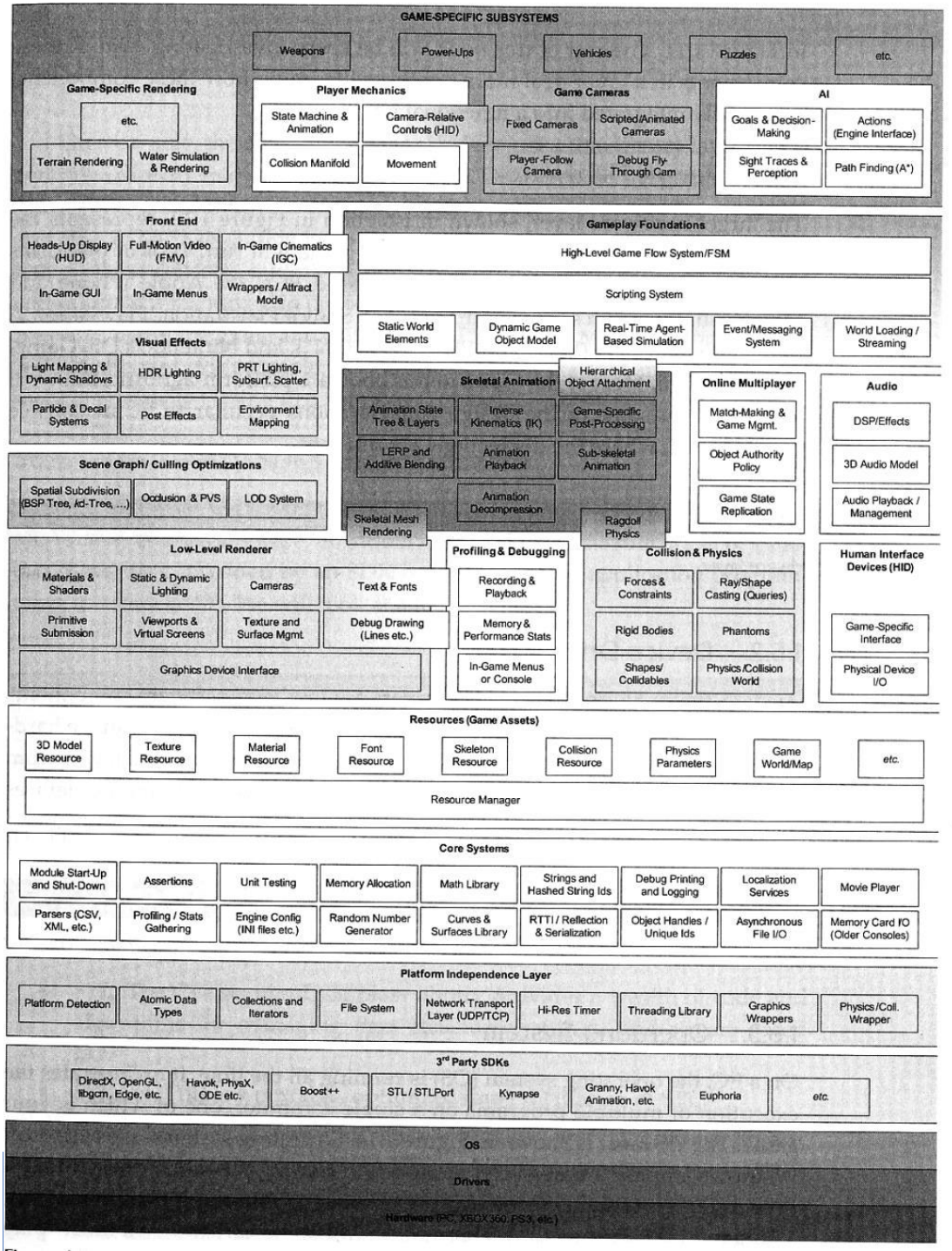
- **Every game is a simulation**!
  - It runs in discrete time steps
  - There is an initial state
  - A new state is calculated based on the current state
  - The new state gets displayed / visualized
  - The changes leading to the new state are based on
    - the game logic (which implements the game mechanic)
    - the input of the user
- Is every simulation a game?

The Game Loop




```
void main() {
    initialize();
    while(running) {
        handleInput();
        update();
        draw();
    }
    shutdown();
}
```

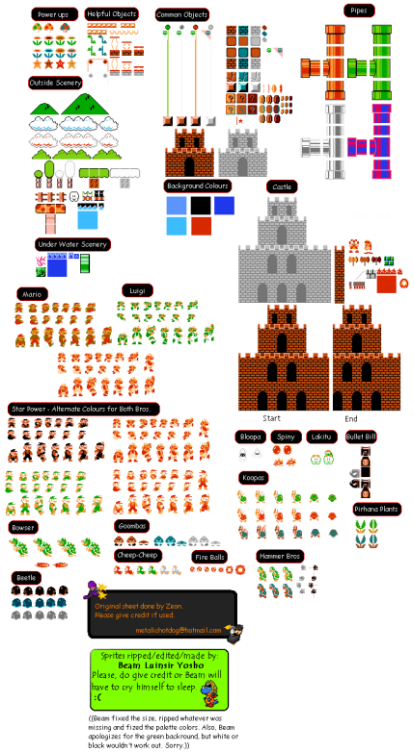

GAME-SPECIFIC SUBSYSTEMS
Weapons
Power-Ups
Vehicles
Puzzles
etc.
Game-Specific Rendering
etc.
Terrain Rendering
Water Simulation & Rendering
Player Mechanics
State Machine & Animation
Camera-Relative Controls (HID)
Collision Manifold
Movement
Game Cameras
Fixed Cameras
Scripted/Animated Cameras
Player-Follow Camera
Debug Fly-Through Cam
AI
Goals & Decision-Making
Actions (Engine Interface)
Sight Traces & Perception
Path Finding (A*)
Front End
Heads-Up Display (HUD)
Full-Motion Video (FMV)
In-Game Cinematics (IGC)
In-Game GUI
In-Game Menus
Wrappers/ Attract Mode
Gameplay Foundations
High-Level Game Flow System/FSM
Scripting System
Static World Elements
Dynamic Game Object Model
Real-Time Agent-Based Simulation
Event/Messaging System
World Loading / Streaming
Visual Effects
Light Mapping & Dynamic Shadows
HDR Lighting
PRT Lighting, Subsurf. Scatter
Particle & Decal Systems
Post Effects
Environment Mapping
Skeletal Animation
Hierarchical Object Attachment
Animation State Tree & Layers
Inverse Kinematics (IK)
Game-Specific Post-Processing
LERP and Additive Blending
Animation Playback
Sub-skeletal Animation
Animation Decompression
Online Multiplayer
Match-Making & Game Mgmt.
Object Authority Policy
Game State Replication
Audio
DSP/Effects
3D Audio Model
Audio Playback / Management
Scene Graph/ Culling Optimizations
Spatial Subdivision (BSP Tree, kd-Tree, ...)
Occlusion & PVS
LOD System
Skeletal Mesh Rendering
Ragdoll Physics
Low-Level Renderer
Materials & Shaders
Static & Dynamic Lighting
Cameras
Text & Fonts
Primitive Submission
Viewports & Virtual Screens
Texture and Surface Mgmt.
Debug Drawing (Lines etc.)
Graphics Device Interface
Profiling & Debugging
Recording & Playback
Memory & Performance Stats
In-Game Menus or Console
Collision & Physics
Forces & Constraints
Ray/Shape Casting (Queries)
Rigid Bodies
Phantoms
Shapes/ Collidables
Physics/Collision World
Human Interface Devices (HID)
Game-Specific Interface
Physical Device I/O
Resources (Game Assets)
3D Model Resource
Texture Resource
Material Resource
Font Resource
Skeleton Resource
Collision Resource
Physics Parameters
Game World/Map
etc.
Resource Manager
Core Systems
Module Start-Up and Shut-Down
Assertions
Unit Testing
Memory Allocation
Math Library
Strings and Hashed String Ids
Debug Printing and Logging
Localization Services
Movie Player
Parsers (CSV, XML, etc.)
Profiling / Stats Gathering
Engine Config (INI files etc.)
Random Number Generator
Curves & Surfaces Library
RTTI / Reflection & Serialization
Object Handles / Unique Ids
Asynchronous File I/O
Memory Card I/O (Older Consoles)
Platform Independence Layer
Platform Detection
Atomic Data Types
Collections and Iterators
File System
Network Transport Layer (UDP/TCP)
Hi-Res Timer
Threading Library
Graphics Wrappers
Physics/Coll. Wrapper
3rd Party SDKs
DirectX, OpenGL, libgcm, Edge, etc.
Havok, PhysX, ODE etc.
Boost++
STL / STLPort
Kynapse
Granny, Havok Animation, etc.
Euphoria
etc.
OS
Drivers
Hardware (PC, XBOX 360, PS3, etc.)


GAME-SPECIFIC SUBSYSTEMS
Weapons
Power-Ups
Vehicles
Puzzles
etc.
Game-Specific Rendering
etc.
Terrain Rendering
Water Simulation & Rendering
Player Mechanics
State Machine & Animation
Camera-Relative Controls (HID)
Collision Manifold
Movement
Game Cameras
Fixed Cameras
Scripted/Animated Cameras
Player-Follow Camera
Debug Fly-Through Cam
AI
Goals & Decision-Making
Actions (Engine Interface)
Sight Traces & Perception
Path Finding (A*)
Front End
Heads-Up Display (HUD)
Full-Motion Video (FMV)
In-Game Cinematics (IGC)
In-Game GUI
In-Game Menus
Wrappers/ Attract Mode
Gameplay Foundations
High-Level Game Flow System/FSM
Scripting System
Static World Elements
Dynamic Game Object Model
Real-Time Agent-Based Simulation
Event/Messaging System
World Loading / Streaming
Visual Effects
Light Mapping & Dynamic Shadows
HDR Lighting
PRT Lighting, Subsurf. Scatter
Particle & Decal Systems
Post Effects
Environment Mapping
Skeletal Animation
Hierarchical Object Attachment
Animation State Tree & Layers
Inverse Kinematics (IK)
Game-Specific Post-Processing
LERP and Additive Blending
Animation Playback
Sub-skeletal Animation
Animation Decompression
Online Multiplayer
Match-Making & Game Mgmt.
Object Authority Policy
Game State Replication
Audio
DSP/Effects
3D Audio Model
Audio Playback / Management
Scene Graph/ Culling Optimizations
Spatial Subdivision (BSP Tree, kd-Tree, ...)
Occlusion & PVS
LOD System
Skeletal Mesh Rendering
Ragdoll Physics
Low-Level Renderer
Materials & Shaders
Static & Dynamic Lighting
Cameras
Text & Fonts
Primitive Submission
Viewports & Virtual Screens
Texture and Surface Mgmt.
Debug Drawing (Lines etc.)
Graphics Device Interface
Profiling & Debugging
Recording & Playback
Memory & Performance Stats
In-Game Menus or Console
Collision & Physics
Forces & Constraints
Ray/Shape Casting (Queries)
Rigid Bodies
Phantoms
Shapes/ Collidables
Physics/Collision World
Human Interface Devices (HID)
Game-Specific Interface
Physical Device I/O
Resources (Game Assets)
3D Model Resource
Texture Resource
Material Resource
Font Resource
Skeleton Resource
Collision Resource
Physics Parameters
Game World/Map
etc.
Resource Manager
Core Systems
Module Start-Up and Shut-Down
Assertions
Unit Testing
Memory Allocation
Math Library
Strings and Hashed String Ids
Debug Printing and Logging
Localization Services
Movie Player
Parsers (CSV, XML, etc.)
Profiling / Stats Gathering
Engine Config (INI files etc.)
Random Number Generator
Curves & Surfaces Library
RTTI / Reflection & Serialization
Object Handles / Unique Ids
Asynchronous File I/O
Memory Card I/O (Older Consoles)
Platform Independence Layer
Platform Detection
Atomic Data Types
Collections and Iterators
File System
Network Transport Layer (UDP/TCP)
Hi-Res Timer
Threading Library
Graphics Wrappers
Physics/Coll. Wrapper
3rd Party SDKs
DirectX, OpenGL, libgcm, Edge, etc.
Havok, PhysX, ODE etc.
Boost++
STL / STLPort
Kynapse
Granny, Havok Animation, etc.
Euphoria
etc.
OS
Drivers
Hardware (PC, XBOX 360, PS3, etc.)

Rendering

- Rendering is the **process** where your game **throws graphics on your screen**.
- The rendering process can be computed **2 or 3 dimensions**.
- Calculations of graphics need very much **computational power**, thus we often need **additional hardware**.
- At this point we are talking about **hardware acceleration** (e.g. through **GPUs**)
- **3D games** need more computational power but they **are not easier to realize than 2D games**!

2D

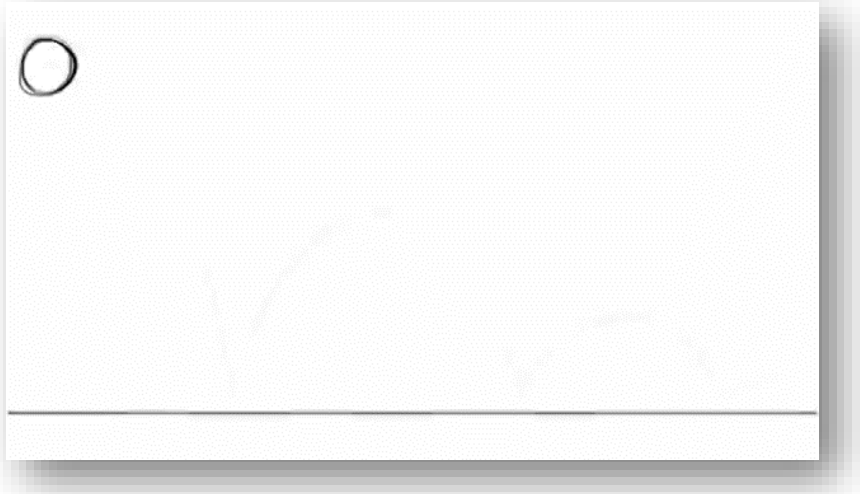


The Dimensions

	2D	3D	Virtual Reality
Ego / First-Person View	Nur bedingt	Ja	Ja
Third-Person View	Nein	Ja	Nein *
Fahrzeug / Cockpit	Nur bedingt	Ja	Ja
Side Scroller	Ja	Ja	Nein *
Top Down	Ja	Ja	Nein *
Isometric	Ja	Ja	Nein **

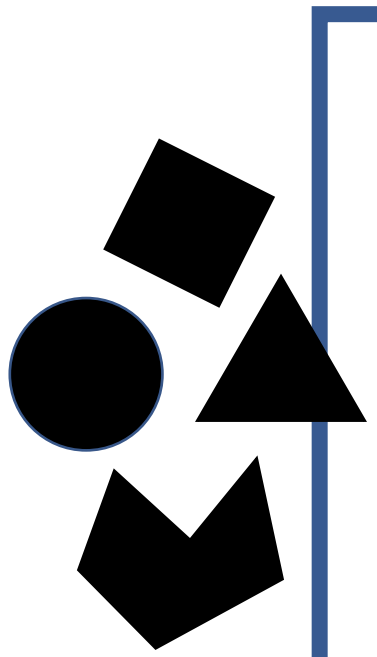
Animations and Physics

- Using physics adds realism and (some kind of) predictability
- Objects usually have **mass**, **friction**, and **volume**.
- Objects only move when **forces** applied to them.
- Motion is not linear, its **curved**.
- Interaction must be **responsive**.
- States change with **transitions**.
- **Sounds** can support animation.
- **Physics** can support animation.

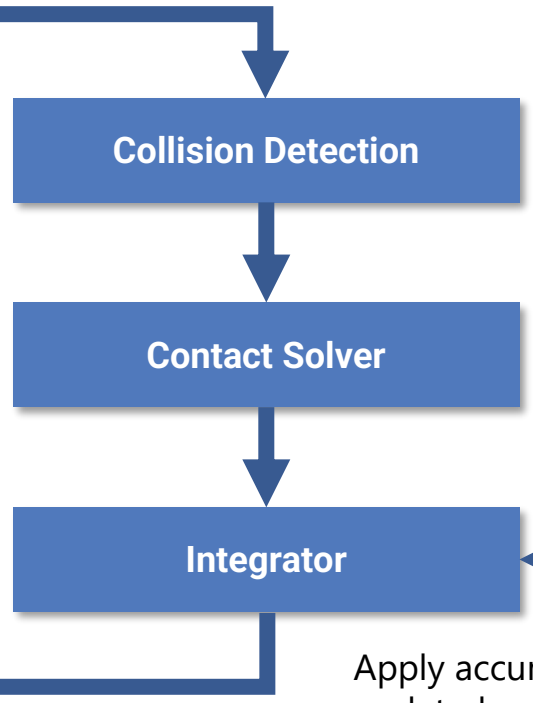


Disney's 12 Principles of Animation

Collisions



Collision Objects = Simple Geometry



Generate contact details for interpenetrating /touching bodies

Resolve contacts to separate any interpenetrating bodies and produce resultant forces/impulses.

Apply accumulated forces/impulses to bodies to produce updated positions and velocities

Invent or Pick Up?

- **INVENT**
  - Draws much attention
  - All players have the same pre-knowledge
  - You can be much more creative
  - Very hard to realize
  - Designing new rules can be incredibly complex and time-consuming
- **PICK UP**
  - Existing game concepts are well explored
    - Tower Defense
    - Turn-based Strategy
    - …
  - You can learn from others (Mechanics, Balancing)
  - People are (fast) familiar with your interface

Create Your First Game

- Conceptualize!
- Keep your first game small
- Hold back larger ideas for the 2nd, 3rd, …game you make
- Every idea, content and/or technology you plan takes time (and experience and manpower) to implement!
- Learn the technology while you go
- Technology shapes your idea
- Your idea shapes technology
- Keep it simple!
- Don’t explain everything, leave enough space for the player’s interpretations

Flow Chart

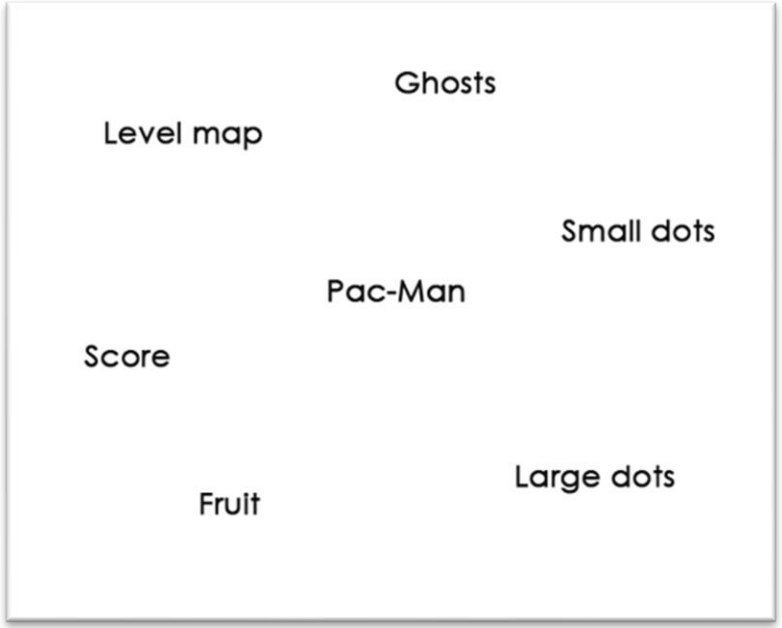


Image from: http://livelyivy.com/gravityghost-com/5-alternatives-to-a-game-design-doc/

Flow Chart

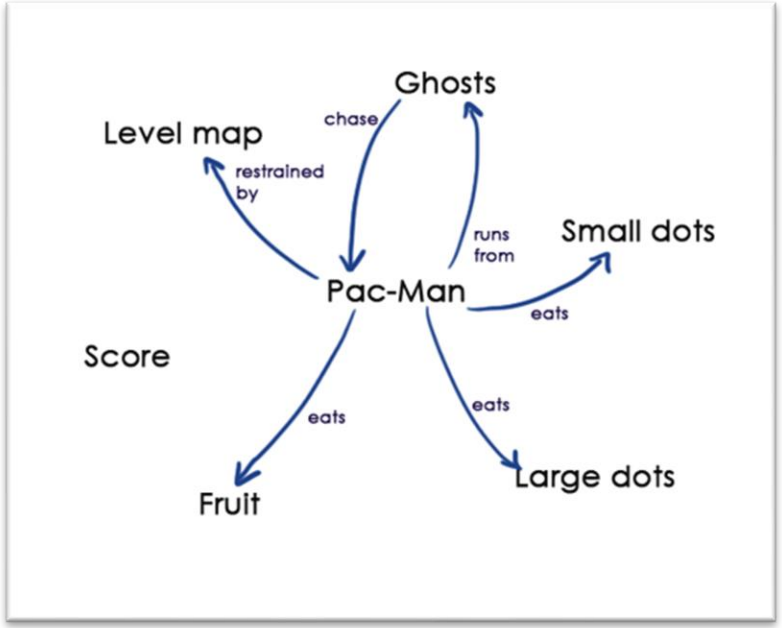


Image from: http://livelyivy.com/gravityghost-com/5-alternatives-to-a-game-design-doc/

Flow Chart

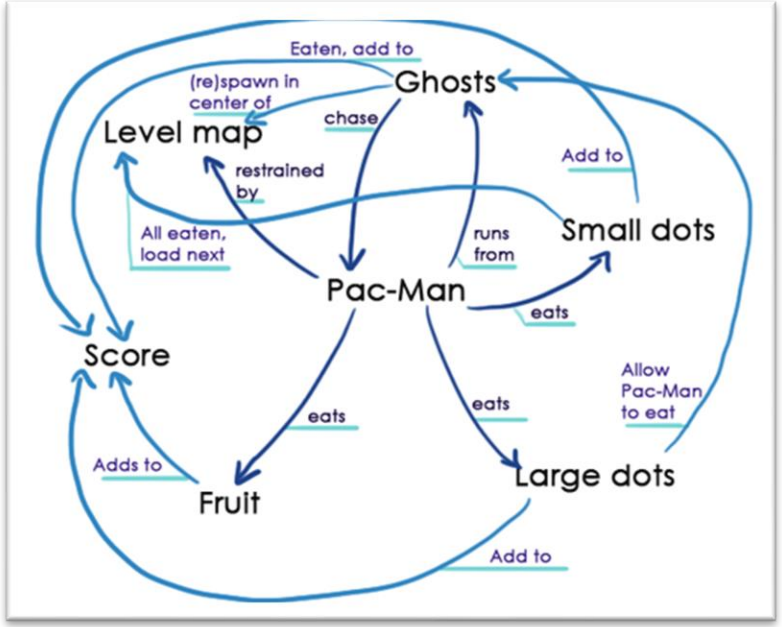


Image from: http://livelyivy.com/gravityghost-com/5-alternatives-to-a-game-design-doc/

About Randomness

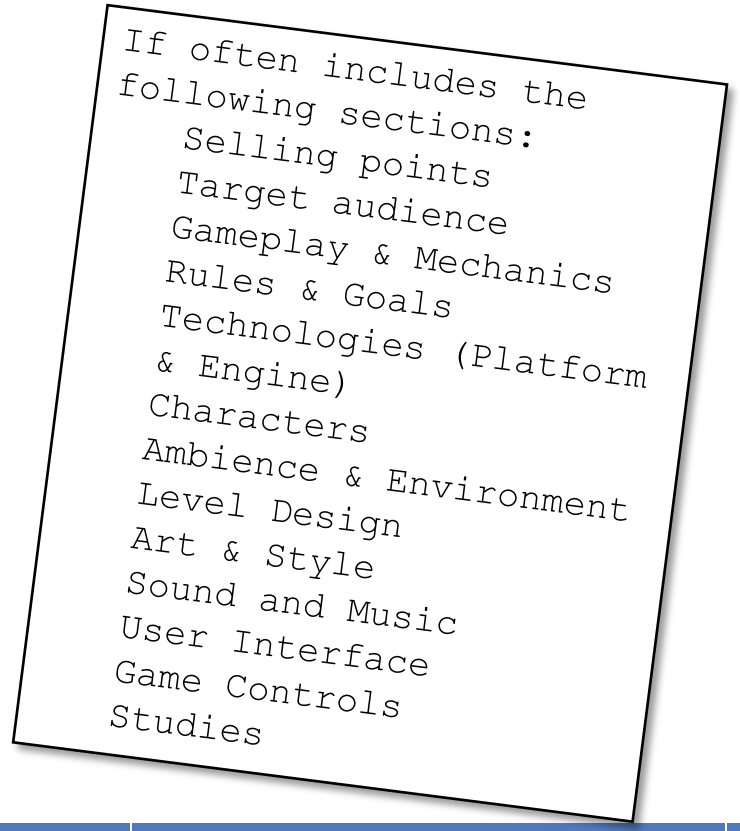
- **Random Number Generators** (Dice, Coins, etc.)
  - Adds (re-)playability
  - Situations are less predictable
  - Randomness should not appear random
  - Randomness should be part of an unpredictable world
- **Pseudo-Randomness**
  - Chess has no random factors but…
  - …more unique games than estimated number of atoms in the Universe
  - Pseudo-Randomness (seed number) is used to generate procedural content
    - e.g. the seed number of the universe in Elite Dangerous is the telephone number of the main developer

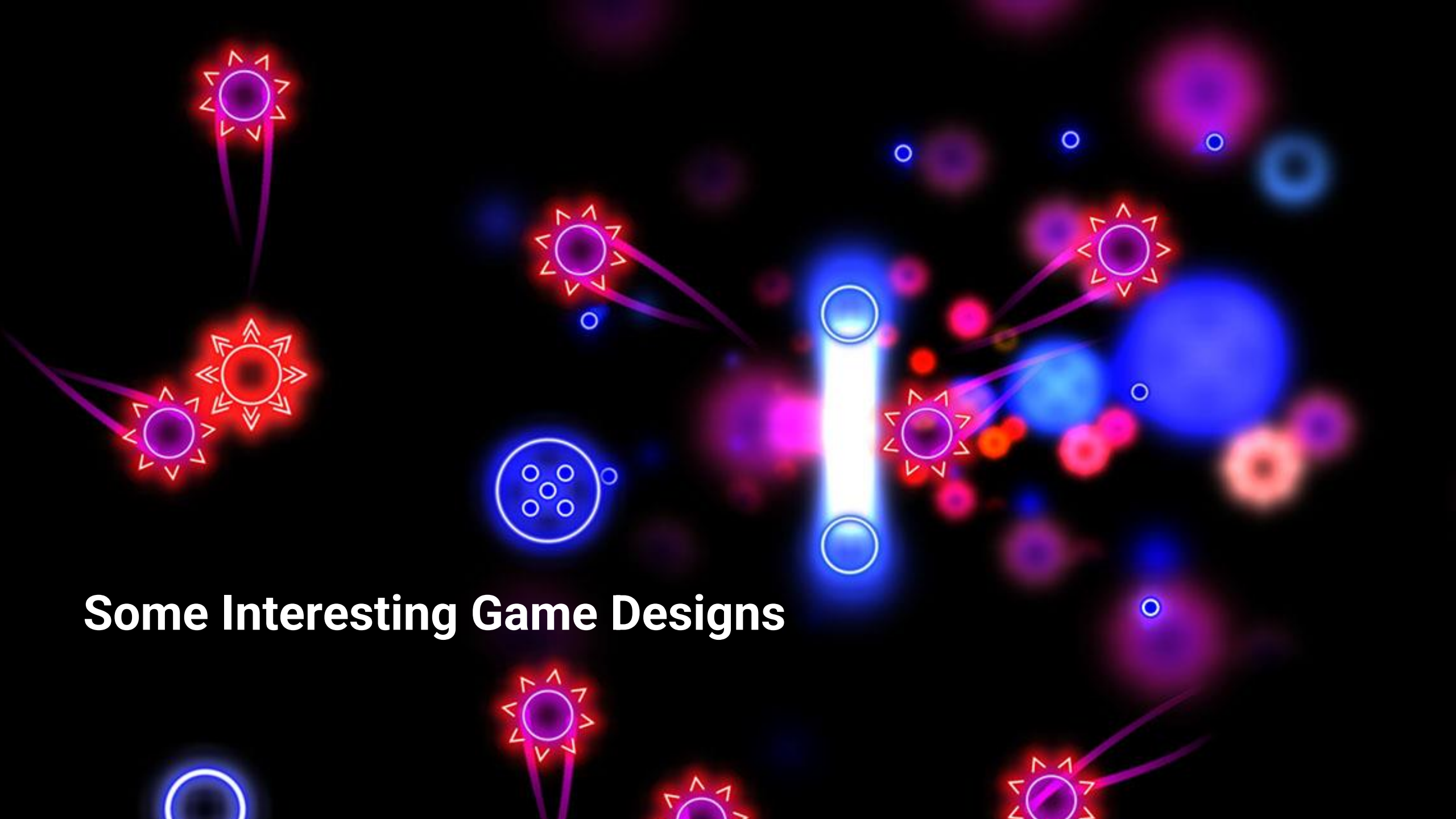
Final Questions of Game Design

- Does this game meet our social and community goals?
- Will the intended audience like this game enough?
- Is it technically possible to build this game?
- Do the testers enjoy this game enough?
- Is this a well-designed game?
- Does this game feel right?
- Is this game novel enough?
- Will this game sell?

Game Design Document

- The GDD has no standard form and is a living document created by several iterations.
- Starts with the basic concept
- Explains detailly every game part
- Varies from game to game
- It will be revised.
- It will be approved.
- It will be expanded.





Some Interesting Game Designs


Intel's
GAME ON
05:00
12:00
DELL


AM 7:53
0
$40,269


42820
10 0 0
2 0 0
0 5 0
1 0 0
0 4 0
15
Training Room

Take Home Messages

- Games are entered willfully.
- Games have goals.
- Games have conflicts.
- Games have rules.
- Games can be won and lost.
- Games are interactive.
- Games have challenge.
- Games can create their own internal value.
- Games engage players.
- Games are closed, formal systems.
- Games are simulations.

References

- Game Design
  - Jane McGonigal: *Reality is Broken*. Vintage, 2011
  - Jesse Schell: *The Art of Game Design*. 2008
  - Tracy Fullerton: *Game Design Workshop*. 2008
  - Scott Rogers: *Level Up!* Wiley, 2010
- Game Engines (free for educational puposes)
  - Unreal Engine: https://www.unrealengine.com
  - CryEngine: http://cryengine.com/
  - Unity: http://unity3d.com/
  - Ogre3D: http://www.ogre3d.org

References

[1] Peter Shankar - CSE 497 – Topics on AI & Computer Game Programming

[2, 3] Chessboard, Grafik: Ilaboy, (2008), GFDL 1.2, CC BY-SA 3.0, Quelle: http://commons.wikimedia.org/wiki/File:AAA_SVG_Chessboard_and_chess_pieces_06.svg?uselang=de

[4] Foto: Unbekannt, CC0 Public Domain Quelle: http://pixabay.com/static/uploads/photo/2012/11/28/10/48/chess-67660_640.jpg

[5] BOTVINNIK, M.M. (1970). Computers, Chess and Long-Range Planning. Springer-Verlag, New York/Heidelberg. LC 75-85203

[6] Foto: Unbekannt, CC0 Public Domain Quelle: http://pixabay.com/en/source-code-code-programming-c-583537/

[7] Racknitz, Kupferstich, Der spielende Schachtürke (1789), mit freundlicher Genehmigung der Humboldt University, Berlin (Die Wissenschaftlichen Sammlungen)

[8] Cover Art of Battle Chess (1988), coypright by Interplay Entertainment, Fair Use Rationale, Quelle: http://en.wikipedia.org/wiki/File:Battle_Chess.png

[9] Screenshot of Chessmaster (Moby Games), http://www.mobygames.com/images/shots/l/409746-chessmaster-10th-edition-windows-screenshot-animated-piece.jpg

[10] Foley Room at the Sound Design Campus Vancouver Film School Flickr, CC BY-SA 3.0 ,Quelle http://commons.wikimedia.org/wiki/File:Foley_Room_at_the_Sound_Design_Campus.jpg

[11] Hans-Werner Hunziker, (2006), CC BY-SA 3.0, Quelle: http://en.wikipedia.org/wiki/File:Eye_movements_of_a_chess_champion_nc.jpg

[12] Rise of the Tomb Raider - Rise of the Tomb Raider Reactions - E3 2014 - ign.com

[13] Samwise Didier, Izsha, StarCraft II Game Concept, Quelle: http://www.videogamesartwork.com/games/starcraft-ii/izsha

[14] Starcraft-II.eu - artworks, avatars, screenshots, vidéos et wallpapers pour Starcraft 2 - © 2007 www.starcraft-II.eu http://www.starcraft2.gamemasters.fr/screen-wall/starcraft-2-05.jpg

[15] The Art of Game Design: A Book of Lenses, Second Edition, Jesse Schell, CRC Press, 2014, ISBN 9781466598645

[16] Arrangement of equal-sized circles at a distance of the golden angle Ψ = 137.5077°, Andree Stephan, CC BY-SA 3.0 http://commons.wikimedia.org/wiki/File:Fibonacci_numbers.jpg

[17] Luc Fontenoy / Game Art Design, http://lucfonz.blogspot.de/2013/05/interiors.html

[18] Die erfolgreiche Konstruktion virtueller Spielfiguren, Jäger, Solveigh (2012)

{19] Activision Publishing, Character Demo (2014) http://www.gamespot.com/articles/activision-shows-off-next-gen-visuals/1100-6406090/

[20] The Incredibles / Mars needs Mom; Image Property Disney Animation Studios / Pixar (2010)

[21] Maslow's hierarchy of needs. Salenger Incorporated, 1987.

[22] "Nico Bellic", Artwork of "GTA IV", Rockstar Games, http://www.theapricity.com/forum/showthread.php?139378-Classify-Niko-Bellic

[22] Image on p 23: Screenshot from "Hitman Absolution", IO Interactive / Square Enix, 2013

[23] Image on p. 19: Artwork, Tomb Raider , Crystal Dynamics / Core Design / Eidos Montreal / Square Enix, 2013. http://cdn.collider.com/wp-content/uploads/lara-crof-tomb-raider.jpg

[24] Screenshot / Kaldir Artwork, StarCraft II – Wings of Liberty, Blizzard Entertainment (2012), http://starcraft.wikia.com/wiki/Category:Kaldir_images

[25] Screenshot StarCraft II – Wings of Liberty, Blizzard Entertainment (2012)

[26] Gamer with Oculus Rift, Quelle. TheVerge.com, Quelle: http://www.theverge.com/2013/3/9/4081912/oculus-rift-at-sxsw-is-virtual-reality-the-holy-grail-of-gaming

[27] OGN 3rd place match: Champion 2014, by Kacper Smigiel , 2014, https://sg.news.yahoo.com/ogn-3rd-place-match-skt-t1-vs-samsung-054426726.html

[28] Microsoft Flight Simulator X: DirectX 10-Screenshots, 2011, http://www.pcgames.de/Flight-Simulator-X-PC-123220/News/Microsoft-Flight-Simulator-X-Erste-grandiose-DirectX-10-Screenshots-veroeffentlicht-PC-Games-vor-5-Jahren-841187/

[29] "Jumpman" from Donkey Kong, Nintendo 1981, http://mygaming.co.za/news/features/41949-mario-the-first-15-years.html

[30] Lara Croft: Tomb Raider Screenshot, Design by Toby Gard, 1996, Core Design, Crystal Dynamics

[31] Tom Clancy's Rainbow Six Las Vegas 2006, Ubisoft, IGN Game Guides, http://www.ign.com/wikis/tom-clancys-rainbow-six-vegas/File:Shield_animation.jpg

[32] Artwork from "The Incredibles", Directed by Brad Bird, Pixar Animation Studios / Disney Animation Studios, 2004

[33] Mars needs Mom, Directed by Simon Wells, Disney Animation Studios, 2011

[34] Human like robot from Hiroshi Ishiguro, © 2013 by Max Braun, Lizenz: CC BY-NC-SA 2.0 / flickr.com.

[35] Human like robot, © 2013 by Max Braun, Lizenz: CC BY-NC-SA 2.0 / flickr.com.

[36] Clu from Tron Legacy, "Tron: Legacy" (USA 2010), Directed by J. Kosinski, © 2013 Walt Disney Pictures, TC: 01:53:54.

[37] Lauren Winter from Heavy Rain, © 2010 Sony Computer Entertainment America LLC / Quantic Dream.

[38] Wax figure of Naomi Campbell, © 2013 by dreistreifen, Lizenz: CC BY-NC-SA 2.0/ flickr.com.

[39] Real human, © 2013 by lev dolgachov / shutterstock.com

[40] Mori, M. (1970). The Uncanny Valley. Energy.

[41] Image on p. 38: Screenshot from Mass Effect 2, BioWare / Electronic Arts (2012)

[42] Suits,B.(1978):TheGrasshopper:Games,LifeandUtopia