

01 Game Design Basics & Design Documents

Tvorba a dizajn počítačových hier (FMFI)

Návrh a vývoj počítačových hier (FIIT)

Michal Ferko

26. 9. 2019

User-Centered Approach

- What can I give the player?
- What does the player want?
 - What type of players are they?
- How will this affect the player?
- Will this have the effect on the player that I want?
- What **value** do I give the player?

- https://www.gamasutra.com/view/feature/134842/personality_and_play_styles_a_.php

Craft Experiences, not Game Mechanics

- What do we want the player to experience?
- How do we want them to experience it?
- Every decision should work toward improving the experience



Player Motivation

- People play because of different reasons
- You should think about what motivates the player
 - To start playing, to keep playing, to return the next day
- **It changes over time and as you play**

- **Exercise:** Pick a game you played recently
 - Why did you start playing?
 - When you decided to play again, what was your motivation?
 - Why did you stop playing?

Example: Overwatch

- Start playing
 - Word of mouth (fame/hype)
 - Looks amazing (graphics/animation)
- Play again
 - Feels super good to play (Core gameplay)
 - You want to get the next level (Meta gameplay)
 - You want to show off the skin you just got
 - You want to play with friends
 - You want to be better
- Stop playing: repetitive, Bad teammates, constant loss, lack of skill

GAMER MOTIVATION MODEL



Action "Boom!"	Social "Let's Play Together"	Mastery "Let Me Think"	Achievement "I Want More"	Immersion "Once Upon a Time"	Creativity "What If?"
Destruction Guns. Explosives. Chaos. Mayhem.	Competition Duels. Matches. High on Ranking.	Challenge Practice. High Difficulty. Challenges.	Completion Get All Collectibles. Complete All Missions.	Fantasy Being someone else, somewhere else.	Design Expression. Customization.
Excitement Fast-Paced. Action. Surprises. Thrills.	Community Being on Team. Chatting. Interacting.	Strategy Thinking Ahead. Making Decisions.	Power Powerful Character. Powerful Equipment.	Story Elaborate plots. Interesting characters.	Discovery Explore. Tinker. Experiment.

Creating a game

- Need to target a specific group
 - Creating a game that will satisfy everyone is impossible
 - Target a small subset of gamers
 - Male aged 15-25 will probably play different games than Female aged 40-50
 - Pick a few gamer motivations (2-3 max), don't be Minecraft
- Which of the categories is most fun?
 - Highly subjective
 - Every combination of player/time/platform ensures a unique experience

Why?

Why?





5 Whys



Question everything

Why?

Why not?

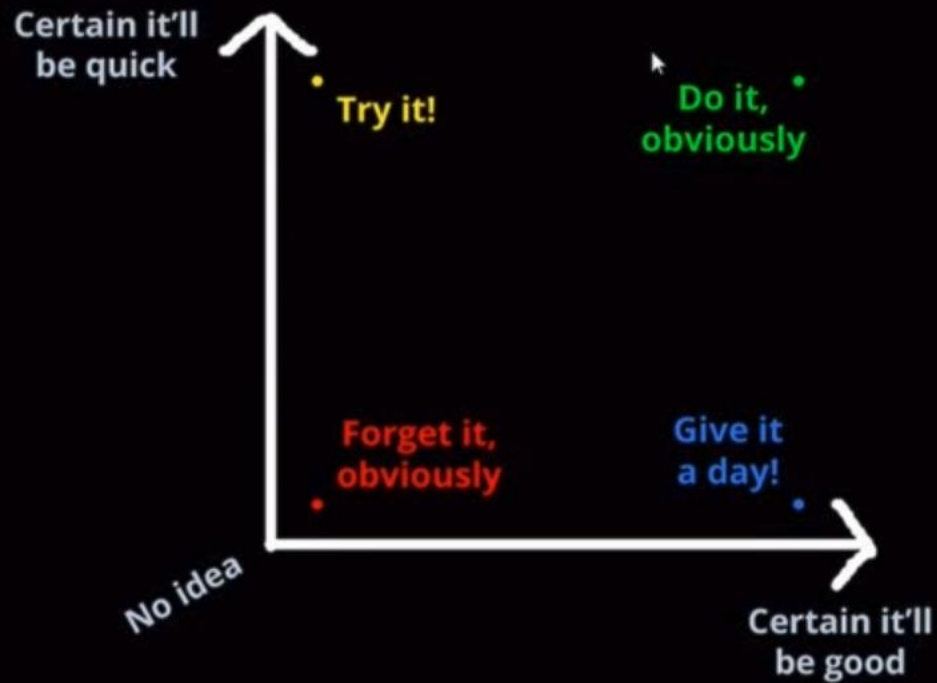
Deliberate Game Design

Deliberate Game Design

- Everything must have a purpose
- How it is fulfilled changes with every change to the game
- Existing purposes need to be re-evaluated
- Remember purposes
- *If you cannot convince the team why it should be in the game, it should be scratched*



Judging an idea without knowing how it'll work out:



Game Design is communication

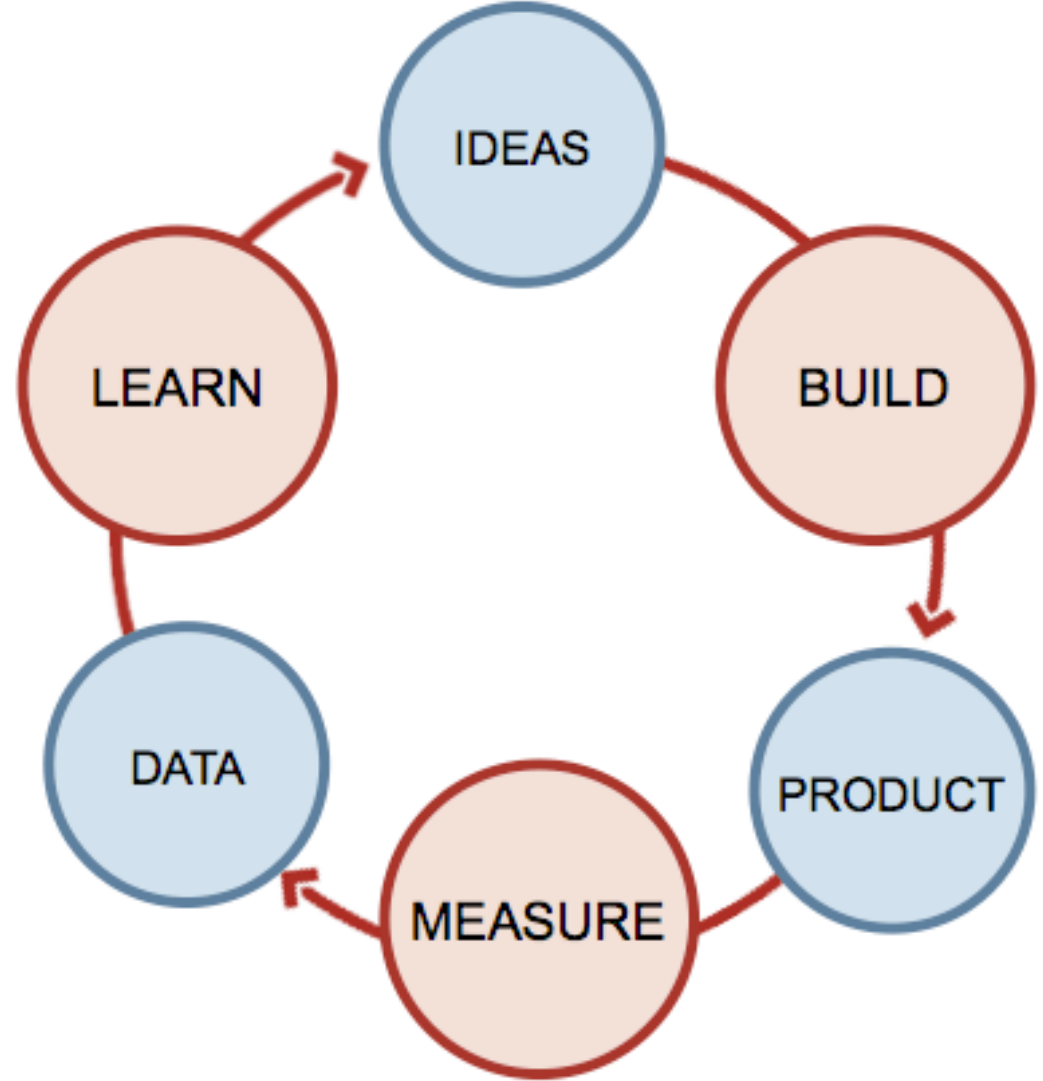
- With the player
 - Always think how you will present something to the player
 - How it will be understood
 - Practice **introspection** – observing yourself while playing games
- With the team
 - Keep a GDD – provides clarity for everyone involved, reminds of purpose
 - Get on the same page
 - Two teams with just a GDD (the same one) should end up with 2 very similar games
- With the customer
 - Think how you will market the game
- *A game is defined more by what you **don't** put in than by what you put in*

It's Research

- Collect, Evaluate, Process, Learn
- You will take in huge amounts of information
 - Do not underestimate processing it
- “I read it once” vs. “I apply these principles on a daily basis”

- **Listen!**

Build, Measure, Learn



Kill Ideas Quickly

Flow in Games

https://www.jenovachen.com/flowingames/Flow_in_games_final.pdf

Flow Attributes

- Clear Goals
- Balance between Challenge vs. Ability
- Immediate Feedback
- Feeling of Control
- Concentration on the Task at Hand
- Sense of Self Disappears

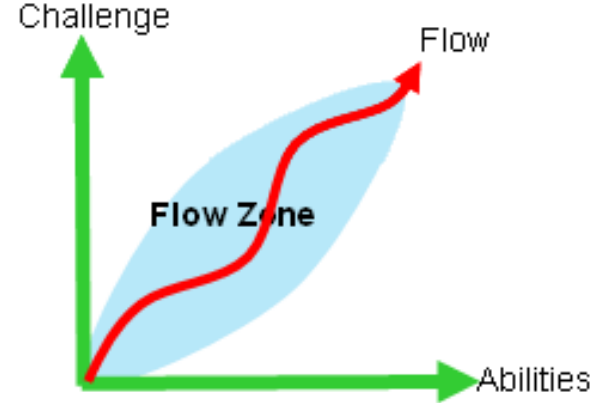


Figure 2 Player in-game Flow experience

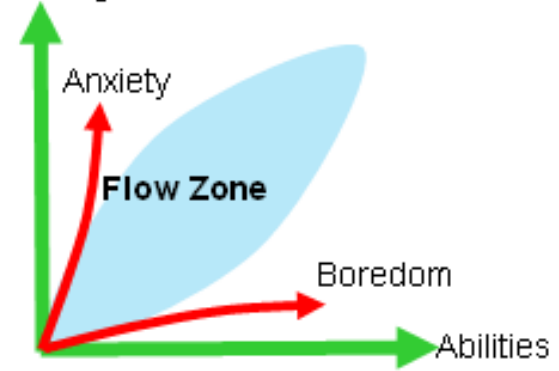


Figure 3 Player encounters psychic entropies

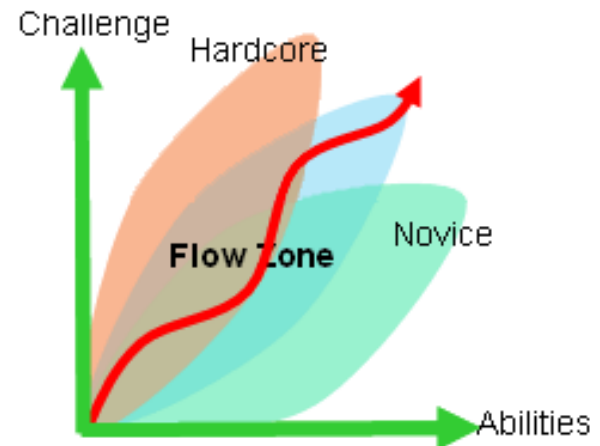


Figure 4 Different players and Flow Zones

**Every Game Developer
is a Game Designer**

Copy stuff!

- Large amounts of research were done into how good games are designed
- If you're doing something completely new
 - It's not tested
 - You don't know exactly what effect it will have
 - You need to test it
 - Testing takes time
- Even if you think it's new, someone might have tried it before
 - But you don't know it
 - And they know why they did not use it
- *Innovate more in fewer places*

Always Steal Like an Artist

- Amazing book – Steal Like an Artist
- Useful for all kinds of creative work
- Read it, it's 50 small pages

GOOD THEFT	VS.	BAD THEFT
HONOR		DEGRADE
STUDY		SKIM
STEAL FROM MANY		STEAL FROM ONE
CREDIT		PLAGIARIZE
TRANSFORM		IMITATE
REMIX		RIP OFF

Game Classification

Classifying games and identifying the target audience

Platforms

- Arcades
- Consoles
- Computers
- “VR”
- Web/Facebook
- Mobile
 - “Toilet and bus” test

Platform dependent development

- The platform usually dictates different I/O devices
 - Controllers
 - Wii U
 - Kinect
 - Gyroscope
 - Screens
 - Multiple screens
 - HMD
- Single-platform games – easier development, more focused market (Switch)
- Multi-platform games
 - Need to solve differences in I/O devices
 - Varying game difficulty due to different I/O methods

Time modes

- Turn-based
 - Strategic, slower
 - Thinking in minutes
 - Less limited by controllers
- Real-time
 - More about reflexes
 - Thinking in milliseconds
 - Difference in controllers can make a huge impact on performance

Player modes

- Single-player
- 1v1 or local multiplayer
 - Need to provide controller for both players
- Local Co-op (PvE)
- LAN multiplayer/co-op
- Online multiplayer

Goals of your game

- Entertainment
- Social
- Educational & Serious games
- Training and Recruitment
- Health and Fitness
- Marketing and Advertising
- Creativity

Genres

- Action – physical challenges
 - Shooters
 - First-person (FPS)
 - Third-person
 - Top-down
 - Shoot'em'up
 - Platformers
 - Racing
 - Fighting

Genres (2)

- Adventure
 - Puzzles, story, freedom of choice
- Action-adventure
- Casino
- Puzzle
- Simulations – tycoons and sims
- Role-playing games (RPG)

Genres (3)

- Strategy
 - Turn-based
 - Real-time
- Massive Multiplayer Online (MMO)
- Games don't have only one genre! Each game is a combination of more genres
 - Skyrim - RPG, Action, Adventure, FPS, occasional puzzles
- Horror, roguelike, MOBA, Battle Royale, Tower Defense, 0451 ...

Game Mechanics

What is a game?

What IS a game?

- *A game is a type of play activity, conducted in the context of a **pretended reality**, in which the participants try to **achieve at least one** arbitrary, nontrivial goal by **acting in accordance with rules**.*
- *A game is a system in which players engage in **artificial conflict**, **defined by rules**, that results in a **quantifiable outcome**.*
- *A game is a **rule-based system** with a variable and quantifiable outcome, where different outcomes are assigned different values, the player exerts effort in order to influence the outcome, the player feels emotionally attached to the outcome, and the consequences of the activity are negotiable.*
- Many games, and game components, can be understood as state machines

Games are unpredictable

- Otherwise, it will get boring fast
- Several ways to introduce unpredictability
 - **Chance** (Randomness) – Blackjack
 - **Player choice** – Rock, Paper, Scissors
 - **Complex gameplay** – combining simple understandable behavior that might create complex (too complex to be predictable) outcome
- Small impact of player choice
 - The player starts to feel frustrated – their decisions do not matter
- Increasing player skill
 - Reduces the amount of unpredictability for complex gameplay
- Most games mix all three ways
 - Think of Warcraft 3 and name examples for each of the three

What's in a game?

- Assets
 - Sounds
 - Textures
 - Models
 - Animations
 - Levels
- **Behavior**
 - How individual objects interact
 - How is the game played
 - What are the **unwritten** rules

Game Mechanics

- Rules of a game
- Not printed instructions that the player is aware of
- Rules are not known beforehand
- The game teaches the player as he progresses
 - This is where digital games differ from board games and card games
 - Board and card games often require that the player knows the rules, so he does not make invalid actions
 - But digital games have a complex system that **enforces** only valid actions

Video Games are about problem solving

- Each mechanic creates challenges
- Provides tools to overcome challenges – solve problems
- Different challenges
 - Physical
 - Mental

Game Mechanics Classification

- More concrete than rules, include everything that affects the actual gameplay
- Example: Monopoly
 - Rules: several pages long
 - Mechanics: Rules + prices and rent of all properties, as well as the text of all Chance and Community Chest cards
- **Core mechanics** – essential activities that players perform over and over
 - Most influential, affecting many aspects of the game
 - Interact with mechanics of lesser importance
- **Meta mechanics**
 - Mechanics wrapped around core mechanics, tying them together
 - Progression mechanics

Game Mechanics

- Storylines in Skyrim
- Combat in Skyrim
- Leveling in Skyrim
- Lockpicking mini-game in Skyrim

Game Mechanics

- Storylines in Skyrim
 - Meta
- Combat in Skyrim
 - Core
- Leveling in Skyrim
 - Meta
- Lockpicking mini-game in Skyrim
 - Support

Five types of mechanics

- **Physics**

- A core mechanic in many games: FPS/Racing/Sport games, Portal, Angry Birds, Braid

- **Internal economy**

- Mechanics of transactions involving game elements that are consumed, collected, traded. Typically, these elements are called **resources**
- Example: resources in Counter-Strike – what are those?

- **Progression mechanisms**

- Describes how the player progresses through the game world
- Getting to a particular place, obtaining an item, unlocking a door, ...

- **Tactical maneuvering**

- Placement of game units on a map
- Strategic advance by placing units in possible locations (e.g. Chess)

- **Social interaction**

- Teams, clans, giving gifts, trading items, creating alliances, ...

	Physics	Economy	Progression	Tactical Maneuvering	Social Interaction
Action	Detailed physics for movement, shooting, jumping, etc.	Power-ups, collectables, points and lives	Predesigned levels with increasingly difficult tasks, storyline to set player goals	Image from <i>Game Mechanics: Advanced Game Design</i>	
Strategy	Simple physics for movement and fighting	Unit building, resource harvesting, unit upgrading, risking units in combat	Scenarios to provide new sets of challenges	Positioning of units to gain offensive or defensive advantages	Coordinated actions, alliances and competition between players
Role-Playing	Relatively simple physics to resolve movement and conflict, often turn-based	Equipment and experience to customize a character or party	Story line and quests to give player a purpose and goal	Party tactics	Play-acting
Sports	Detailed simulation	Team management	Seasons, competitions, tournaments	Team tactics	

Vehicle Simulation

Detailed simulation

Vehicle tuning between missions

Missions, races, challenges, competitions, tournaments

Image from *Game Mechanics: Advanced Game Design*

Management Simulation

Managing of resources, economy building

Scenarios to provide new sets of challenges

Managing of resources, economy building

Coordinated actions, alliances and competition between players

Adventure

Managing a player's inventory

Story to drive game, locks and key to control player progress

Puzzle

Simple, often non-realistic and discrete, physics generate challenges

Short levels providing increasingly more difficult challenges

Social Games

Resource harvesting and unit building, resources spend on personalized content

Quests and challenges to give player a purpose and a goal

Players exchange in-game resources, mechanics encourage player cooperation or conflict

Problems & Tools

“A game is a series of interesting choices”

Problems & Tools

- Give the player problems to solve (challenges)
 - Non-trivial problems that need to be solved
 - Optional problems that the player can ignore
 - But get rewarded by solving them
 - Repetitive – defined by the game genre
 - Use same structure, different variables – different enemies, locations, quest types...
 - Problems can have sub-problems
- Give the player tools to solve these problems – **interesting choices**
 - If a problem has a single solution, it can become repetitive and tedious (*grind*)
 - Ever played a rhythm game?
 - Offer different tools for different problems – overlap is OK, sometimes even desired

Problems & Tools example

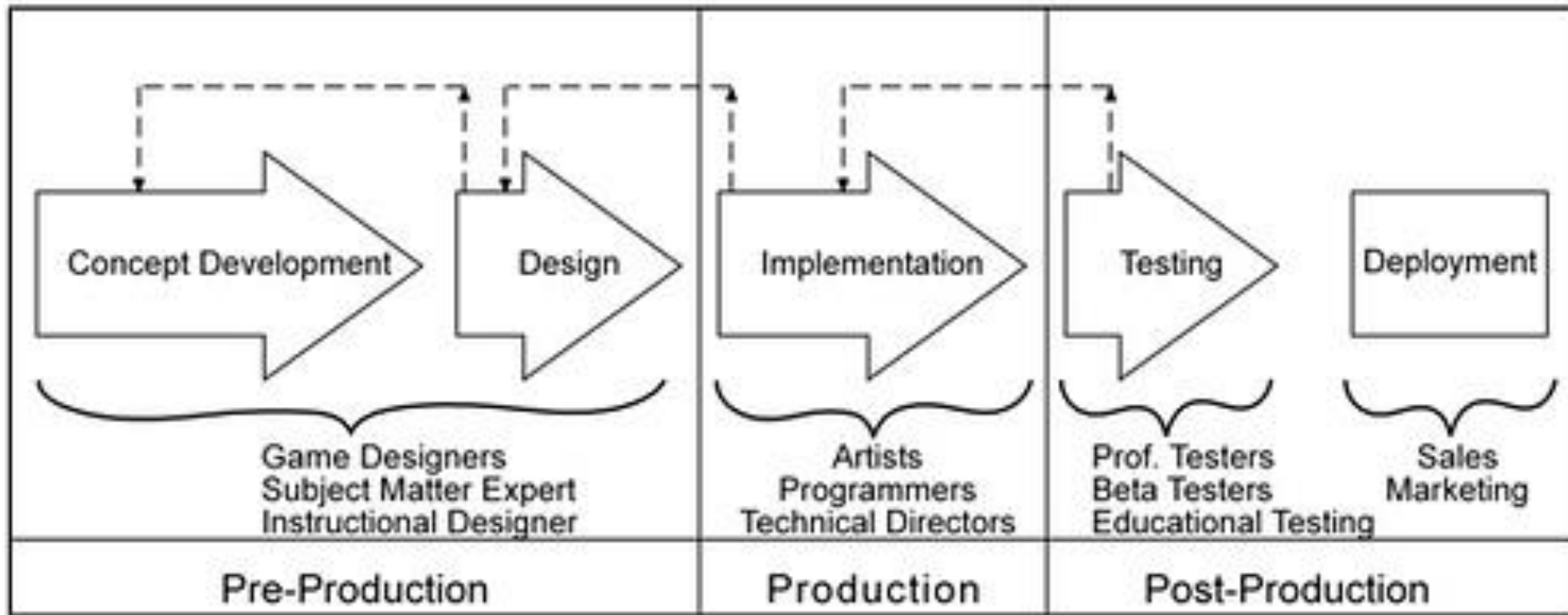
- There's 20 enemies in a room trying to kill you
- Problems
 - I must survive
 - I must kill them to unlock the door
 - I must find something to kill them with
- Tools
 - **Move** to avoid enemies and their shots
 - **Find and pick up** weapons
 - **Fire weapons** to kill enemies
 - **Collect** medkits from dead enemies
- Variability of enemies/levels/weapons/pickups => consistent but varying challenges

Risk & Reward

Risk & Reward

- Actions you take often have a certain risk
 - Running into a group of enemies, moving a chess piece, destructive move in a puzzle game...
- Action you take should result in some reward
 - Loot, damage to enemies, capturing a strategic location...
- Players will always intuitively weigh the risk/reward of actions
 - Nobody will perform a high risk/low reward action
 - Risk & reward are highly context-dependent
- **Rule:** risk should be proportional to reward (low risk => low reward)
- **Exercise:** everyone say a genre and let's analyze

Game development stages



- Image from <https://digitalworlds.wordpress.com/2008/04/10/the-process-of-game-creation-the-game-design-document/>

Development phases

- **Concept** - Start with a small team, work on a concept document
- **Prototype** – not only gameplay, also art
 - *“A working piece of software that captures onscreen the essence of what makes your game special, what sets it apart from the rest, and what will make it successful.”*
- **Pre-production** – Create GDD, TDD, ADD, mechanics
- Production – create all content
- Alpha, Beta, Gold
- Post-production

Problem & Solutions

- Waterfall is not cool
- Moving towards Agile/Scrum
 - Very cool
- Game Development is inherently iterative
- Need to test various systems
 - **Do players like XYZ?**
 - Are the graphics good?
 - Is the music suitable?
 - Are the controls good?
- **Build-Measure-Learn**
 - Read “Agile Game Development with Scrum”

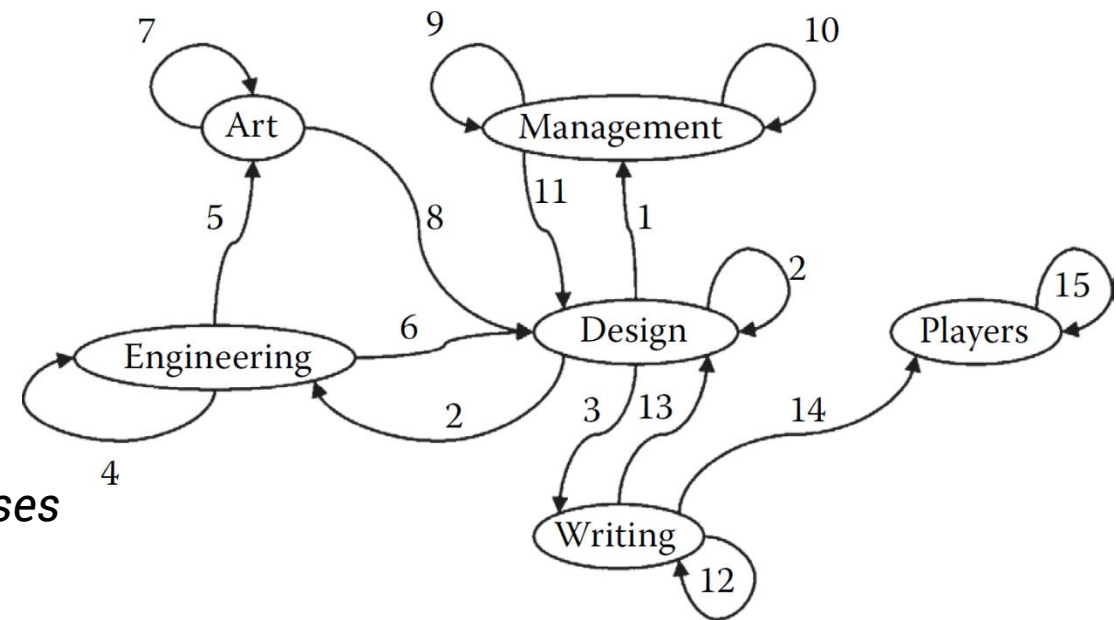
Game documentation

Necessary steps before/during development

Documents related to Game Design

- Concept Document
- Game Proposal Document
- Game Design Document (GDD)
- Technical Design Doc
- Art Style Guide / Art Design Doc
- Project Plan & Schedule
- Test Plan

Image from *The Art of Game Design: A book of lenses*



Concept document

- “pitch”
- Convey the goal and purpose of the game
- For management level to help “sell” the game
 - Assess viability, budget
 - Sell to management, investors, even the team
- Length varies, from 1-2 pages up to 5
- Written by the producer/creative director
- No well-defined form
 - List key things you want to communicate
 - Sort by importance
 - Most important first (since nobody might read the whole thing)

Parts of a concept document

- Premise – high level concept, describe your game and how it is unique in a few sentences
- Player motivation – what will motivate the players? How do they win?
- Unique Selling Points – Why will players pick YOUR game?
 - What makes it different from the rest
- Target Market – Age, Gender(, Platform), ...
 - Target rating (mature content, violence, ...)
- Genre
- HW requirements (platform)
- License

Proposal

- Follow-up of the concept document
- Early stage of a GDD
- Additional sections
 - Hook/unique selling point
 - Audio, Visuals, Gameplay, Story, Mood
 - Gameplay
 - First minutes of gameplay, online features, ...
 - Technology
 - Production details
 - Team, Schedule
 - Art, backstory, characters, ...
 - Market analysis
 - Detailed budget, cost and revenue projections

Game Design Document

- A Game Design Document is very similar to a SRS
 - Not meant to “sell the idea”
- In addition to the complete description of the software, a GDD contains also the “art” part of a game
- The GDD must contain
 - Complete gameplay description – mechanics, interactions
 - Description of all game elements (all creatures, NPCs, items, classes, story...)
- Can contain parts of the SRS
- Often 50 – 200 pages long
- It’s a **living document** – updated throughout

Game Design Document (2)

- Is it necessary?
 - For teams, this is the “holy grail” they refer to when creating a game
 - What about a one-man team?
- The form of a GDD is relatively loose
 - Certain sections are totally irrelevant for certain types of games
 - There is no magic template!
- Purposes: **memory** and **communication**
 - You make lots of decisions and you forget them quickly

One of the possible structures of a GDD

- Game Interface
 - Complete description of all interfaces, their usability, production time and cost, ...
- Game World
 - All elements present in levels
 - Doors, pick-up items, cinematics, triggers, characters, animations, ...
 - Gameplay mechanics
- Character abilities and items
- Game engine
- Incorporated concept art in all sections

Technical Design Document

- Describes the engine on which the game will run
- Comparison to other engines on the market
- Technology production path
 - How they will get from concept to software
- Specific requirements and features, HW & SW
- Refers to the GDD/ADD
- Closer to the SRS, farther from the concept document

Art Design Document

- Describe art that will be used throughout the game
- Mood boards
- Color palettes
- 2D sprites, 3D models
- Animation
- Techniques, software, workflows

Production Plan

- Required team & allocation
- Tasks & time estimates
- When will we ship?
- Marketing strategy

Example Concept Documents and GDDs

Super Mario, Race'n'Chase, BioShock, ...

References

- http://www.gamasutra.com/view/feature/131791/the_anatomy_of_a_design_document_.php
- http://www.gamedev.net/page/resources/_/creative/game-design/tom-slopers-format-for-game-design-specifi-r243
- <http://sloperama.com/advice.html>
- <http://www.cs.tufts.edu/comp/150CIS/AnAntsLife/AnAntsLife-GameDesignDocument.pdf>
- <http://www.scribd.com/doc/53563149/Grand-Theft-Auto-Design-Document>
- <http://www.gamepitches.com/>
- <http://code.tutsplus.com/articles/effectively-organize-your-games-development-with-a-game-design-document--active-10140>

References (2)

- Novak, J. (2012). *Game Development Essentials : An Introduction*. Clifton Park, N.Y: Delmar. 3rd ed.
 - Chapters 1, 2, 3, 4, 11
 - Further reading: Chapters 10, 12
- Adams and Dormans (2012). *Game Mechanics: Advanced Game Design (Voices That Matter)*.
 - Chapter 1 – Designing Game Mechanics
- Keith, C. (2010). *Agile Game Development with Scrum*. Addison-Wesley
 - Just read the whole book!
- Schell, J. (2014). *The Art of Game Design: A Book of Lenses*. A K Peters. 2nd ed.
 - **Must read for every single game dev**

Bonus slides

Structure of a game development team

- Production
 - Design
 - Art
 - Programming
 - Testing and Quality Assurance
 - Audio
-
- This is valid for large companies
 - In indie games, the roles are reduced to only a few people

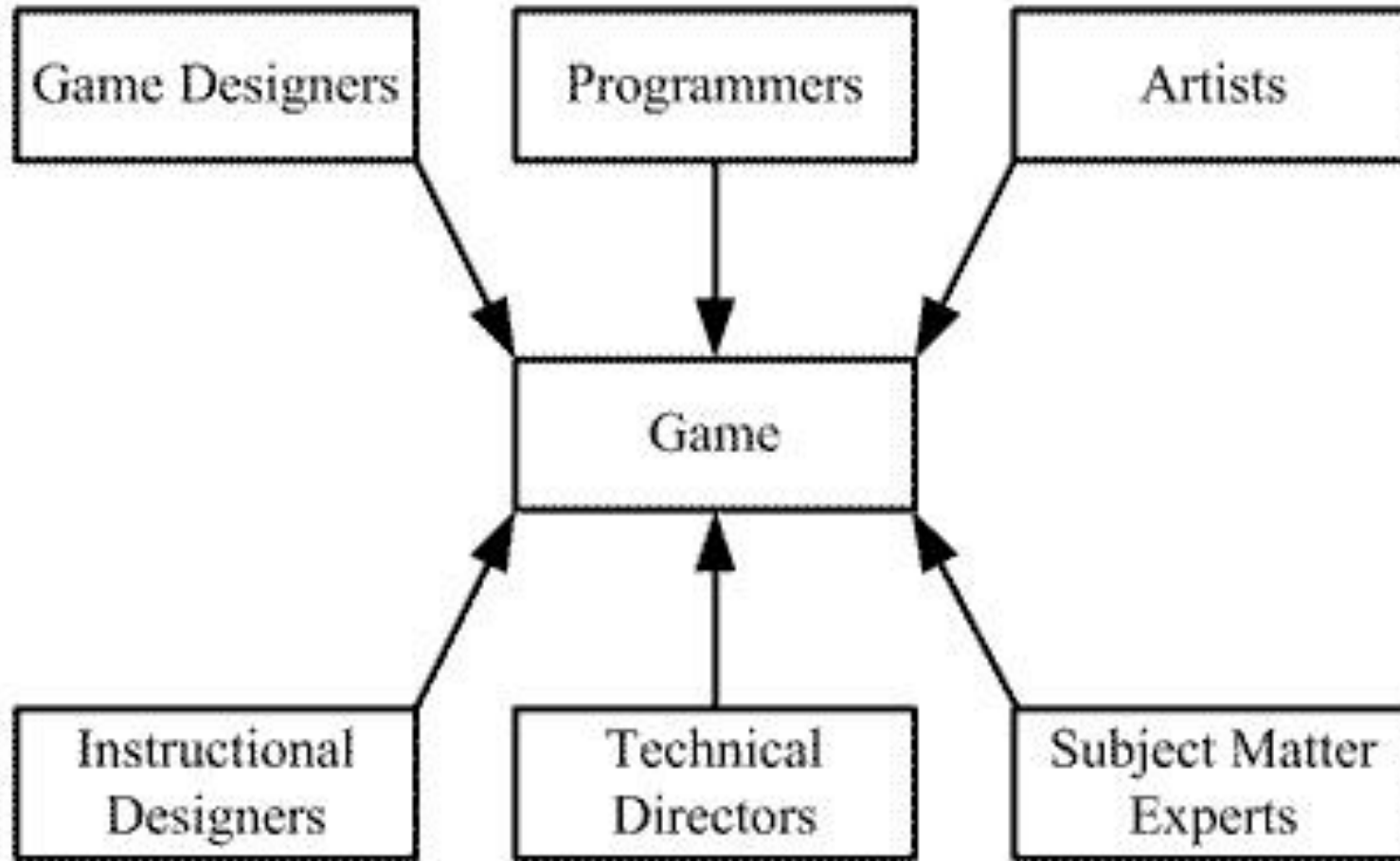


Figure 1. Educational game development team (Click for Larger Image)

- Image from <https://digitalworlds.wordpress.com/2008/04/10/the-process-of-game-creation-the-game-design-document/>

Producers and Management level

- Executive Producer – “Project manager”
 - Management of the proposal (concept), prototype and production
 - Decides which games get made, money distribution, choosing the team
- Producer
 - Keeping the schedule, managing the budget...
- Creative director
 - Management position in design
 - Ensures the overall style and game content are consistent with the original vision
 - Helps maintain the art style while communicating with the art director
- Art director
 - Management of the art development
 - Determining the mood and look & feel of the game

Design

- Often confused with art
- (Creative director)
- Lead designer
 - More hands-on to the actual product
 - Gameplay development, documentation assembly and level design
 - Keeping the vision of the game, the game mechanics, the overall story, breaking up mechanics into simpler parts and game balancing
 - This is usually the “idea guy”
- Interface designer
 - Determines the layout, content, navigation and usability features of the game UI
 - The art team creates the style, which happens after the design stage

Design (2)

- Level designer
 - Creates the environment of the game
 - Some build only physical environments (or worlds)
 - They might also create gameplay and incorporate it into the world
 - Even writing dialogue or stories

Art

- Lead Artist
 - Responsible for parts of the produced art assets
 - Concept Artist, Modeling Artist, Texture Artist, Animator...
- Concept Artist
 - Creates drawings and sketches of characters, environments and props
 - Fast way to visualize the looks and mood of the game
 - Great for presenting an early concept
 - Most of the concept art is not present in the final product
 - Might be used for loading screens, books, paintings... in the game
- Modeler
 - Create 3D assets (buildings, objects, characters) from 2D drawings (usually concept art)

Art (2)

- Texture Artist
 - Generating 2D images for 3D models
 - Photographing or drawing the textures
 - Work closely with the modelers
- Animator
 - Applies movement to game objects
 - Key-framing and motion capture
 - Skinning and Rigging
- Technical Artist
 - Bridge between artists and programmers
 - Understands the technical requirements and engine limitations
 - Ensuring artists use a consistent art pipeline and output assets in the correct format

Programming

- Technical director
 - Creates the technical design for the project
 - Overseeing, selecting tools and hardware, maintaining code standards
 - Build systems, patching systems, installers...
- Lead programmer
 - Supervisor of the programming team
 - Does a lot of coding himself
- Graphics, AI, Audio, Physics, Interface, Network, Engine programmer
- Tools programmer