

Design of computer video games

2. Game development

Boyan Bontchev

Agenda

- Player-centric design
- The game design process
- Game mod's
- Modern game engines
- Interaction and camera models
- Gameplay modes
- Documenting game designs
- Developers, publishers, distributors, and retailers
- Game Development Timeline
- Game industry trends

References

- Adams, E. Fundamentals of Game Design, Third Edition, Pearson Education, Inc., ISBN-13: 978-0-321-92967-9, 2014
- Fullerton, T., Swain C., Hoffman, S. S. Game Design Workshop: A Playcentric Approach to Creating Innovative Games, Second Edition, Elsevier, ISBN 978-0-240-80974-8, 2008
- WPI. The Game Development Process. Introduction (lecture)
- X. Wang, Modelling and Animation for Games, DESC 9188
- Introduction to Game Development, edited by Steve Rabin, 2nd edition, copyright Charles River Media Incorporated, 2009. ISBN: 9781584506799

Player-centric design

- A philosophy of design in which the designer envisions a representative player of a game the designer wants to create
- Player-centric designer undertakes two key obligations to that player:
 - The duty to entertain: the game's primary function
 - The duty to empathize: to design a game that entertains the player, the designer must imagine that he is the player
- The designer must build the game to meet the player's desires, expectations and preferences for entertainment.

Two common misconceptions about player-centric design

1. I AM MY OWN TYPICAL PLAYER.

- you cannot assume that players like what you like
- you must learn to design for what *they* like

2. THE PLAYER IS MY OPPONENT.

- your design goal is to entertain the player by a variety of means
- but not simply to oppose him/her forward progress through the game.

Player-centric vs designer-centric

■ Following the **player-centric approach** –

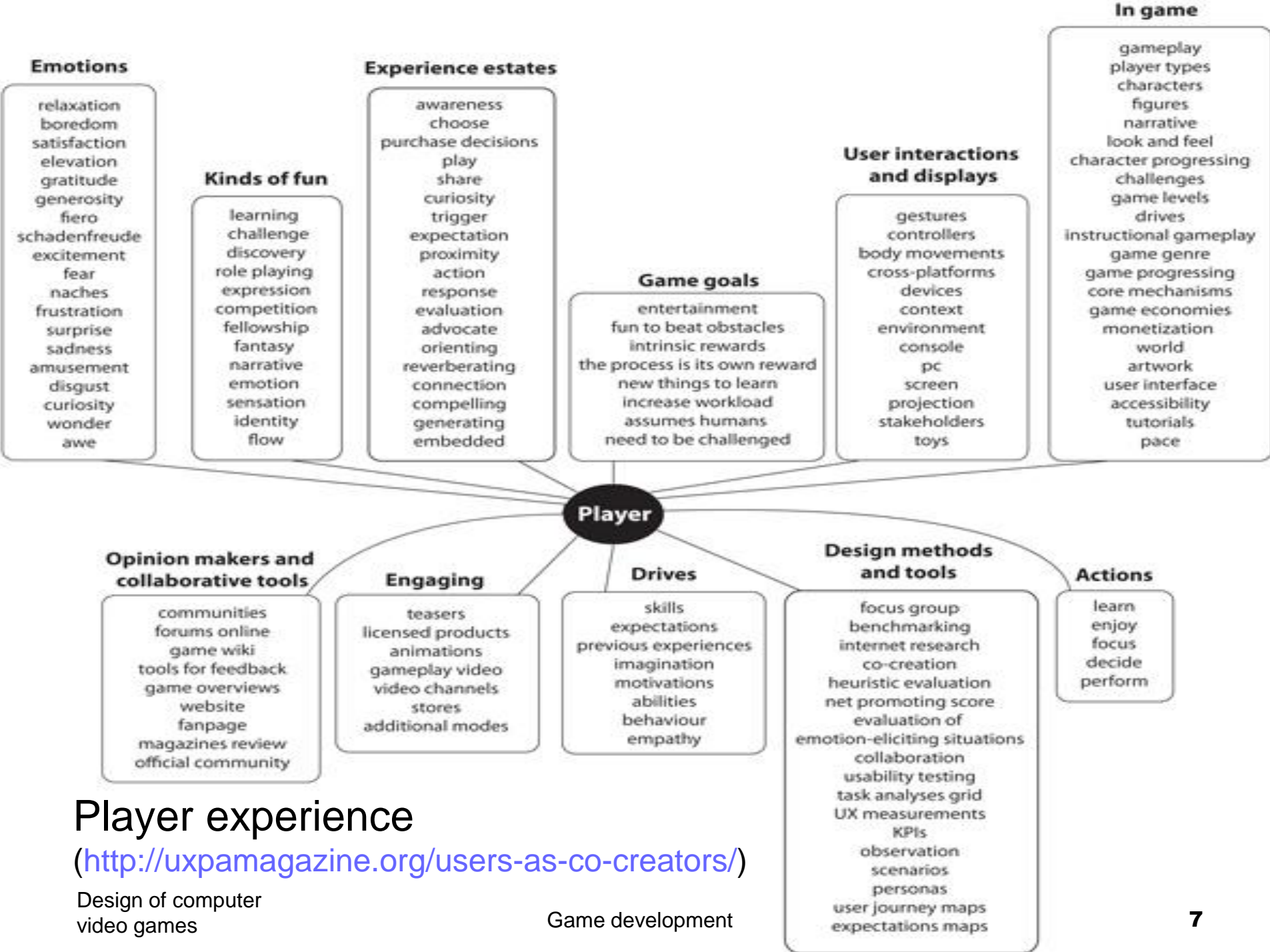
- you use a particular *desirable player experience*
- to constrain the rest of the design, which you then 'solve' for, and hope you end up with a functional game

■ Following **designer-centric approach** –

- you constrain some desirable aspects of the game system,
- and hope it leads to *good player experience*.

See the discussion at

https://www.reddit.com/r/gamedesign/comments/5t9fsb/playercentric_design_vs_designercentric_design/



Player experience

(<http://uxpamagazine.org/users-as-co-creators/>)

Game Design Skills 1/2

■ Imagination

- Visual – characters, environments, objects
- Auditory – environments, voices, music
- Dramatic – emotions
- Conceptual – ideas, interactions, challenges

■ General Knowledge

- Mathematics + Logic
- History
- Literature
- Art
- Science
- Current affairs

Game Design Skills 2/2

■ Technical Awareness

- A understanding of the technology is needed to work with progr.

■ Analytical Skills

- Including statistics

■ Aesthetic Abilities

- Game designers together with the lead artist will set the visual tone

■ Writing Skills

- Technical writing for documenting the design
- Fiction writing for developing story and narrative
- Script writing for developing character dialogue

■ Drawing Skills

- Game designers don't need to be great artists but being able to sketch out ideas can sometimes be the best way to communicate them to other team members

Design process problems

- *“The problem is communication—communication between company and players, communication between game professionals.”* [Fabio Florencio, CESAR](#)
- *“By understanding the player audience’s needs and expectations, it is possible to create better ways of motivation [game-play], realizing efficient products and solutions, spreading the message purposed [communication plan] in an involving manner, raising the chances to promote innovation.”* [Melina Alves, DUX Coworking](#)
- *“Game designer and developers many times don’t see the relevance of including players to test the concept of the games, many times because of time or the costs.”*
[Francimar Maciel, Nokia Institute of Technology](#)

Users as Co-creators 1/4

- Users get more familiar with technology and accessing information - they start speaking about their feelings
- Companies realize the potential value of including customers in the development process
- Good examples:
 - Rovio started with the mobile game Angry Birds and, next, used a transmedia (multiplatform storytelling) approach with various channels to reach new users: Facebook fanpage, Angry Birds store, Angry Birds Wiki, books, toys, movie



Users as Co-creators 2/4

■ Good examples:

- FarmVille game from Zynga enjoyed an great community on Facebook asking the company to create a harvester machine that allows the players to harvest many blocks at the same time



Users as Co-creators 3/4

- Some game companies give players the freedom to create ***add-ons*** or ***mods*** for the game.
- Blizzard Entertainment maintains a large, official community for the game World of Warcraft (WOW). Players are very engaged in it; they talk about add-ons (see <https://mods.curse.com/addons/wow>), better features, experiences, and so on.



Users as Co-creators 4/4

- Minecraft is an indie-game developed in 2009 by Markus Persson with nine million copies sold until 2013:
 - “survival mode” (default) - players have to survive in a world where they can build and destroy blocks
 - “creative mode” - players have an unlimited number of blocks in an expanding and exploratory world



Other motivations for game design

- Market-driven games – most such games have no success: you can't make a brilliant game simply by throwing in all the most popular kinds of gameplay
- Designer-driven games - the designer retains all creative control and takes a personal role in every creative decision. Example: Daikatana
- *Technology-driven* games - designed to show off a particular technological achievement like new graphics engine. Example: Crytek's game *Crysis*



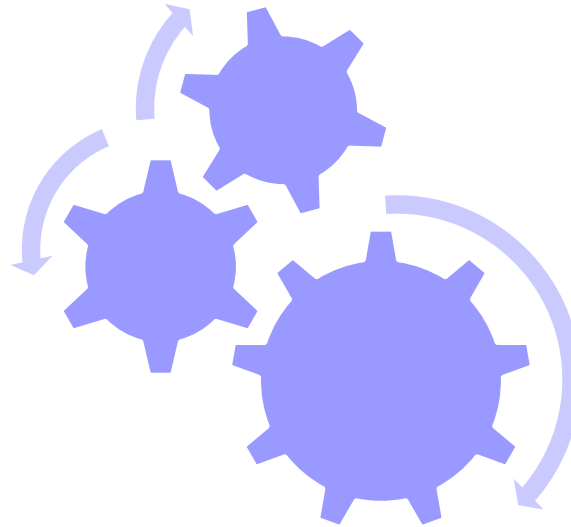
Users in the company mindset

- Customization, based on features that users can adapt to make the game more suitable for them
- Personalization, in which the user makes the design and applies it in a product or service
- Collaboration, where the user contributes with part of the work
- Co-creation, with users helping the designer come-up with better solutions

Source: <http://uxpamagazine.org/users-as-co-creators/>

Stages of the design process

ELABORATION STAGE



CONCEPT STAGE



TUNING STAGE

E. Adams's game design process 1/4

Design Stages	Methods and Tools
1. Concept – Imagining the game and defining the way it works.	
a. Getting a concept	Idea generation, benchmarking, affinity diagrams, business model canvas, concept drawing, and game advertising
b. Defining the audience	Potential players interviews, focus groups, personas, stakeholder map, scenarios
c. Determining the player's role	Use cases, player journey maps, player experiences maps, stakeholder map, and similar games safaris
d. Fulfilling the dream: first steps for the gameplay definition	Group and individual interviews, experience prototype, Five Whys, What if..., mood boards and storyboards

E. Adams's game design process 2/4

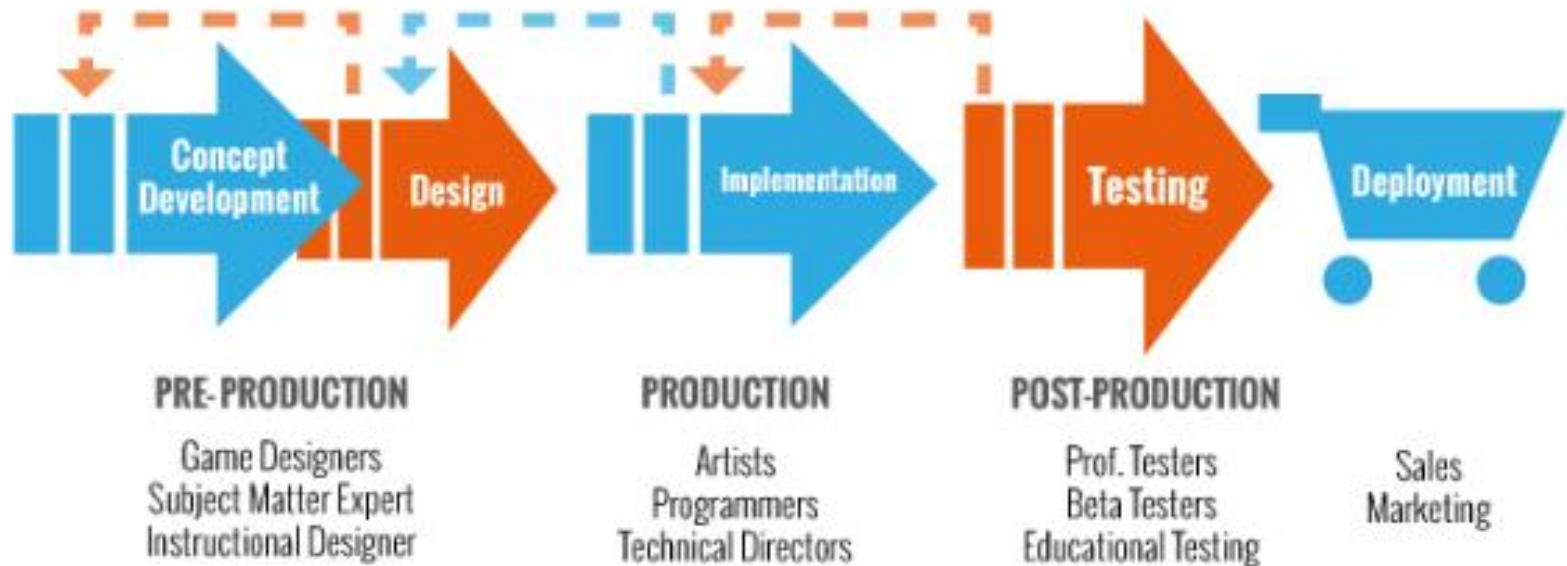
Design Stages	Methods and Tools
2. Elaboration – Transmitting information about the game to the team who will build it.	
a. Defining the primary gameplay mode	Paper prototypes, focus groups and interviews, player expectations maps, and player lifecycle maps
b. Designing the protagonist	Player interviews and focus groups, roleplaying
c. Defining the game world	Players brainstorm, team brainstorm, affinity diagram, player lifecycle maps, mood boards, futures cards
d. Designing the core mechanisms	Players brainstorm and team brainstorm, paper prototypes and prototype simulators, expectations maps, interviews, focus groups

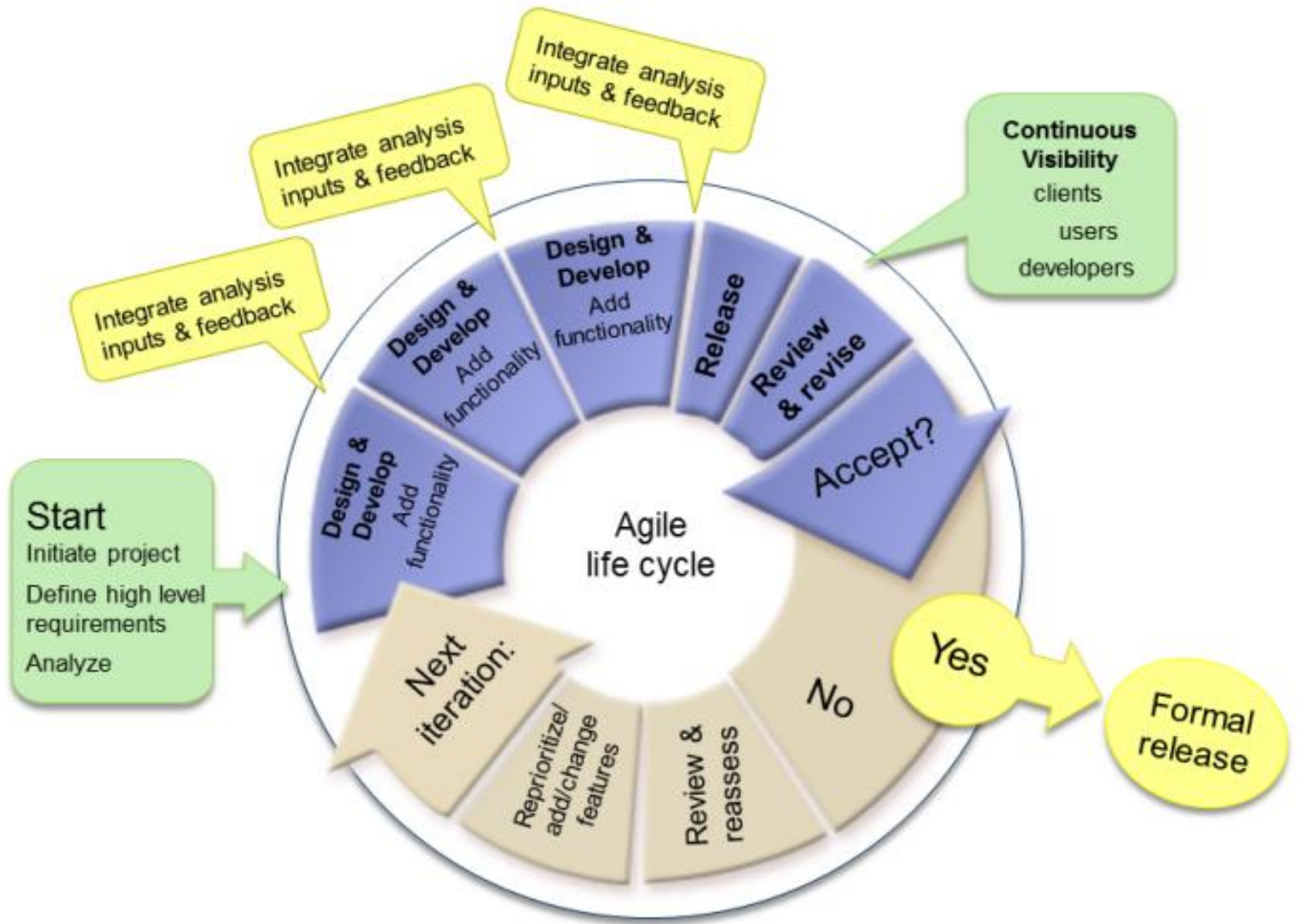
E. Adams's game design process 3/4

Design Stages	Methods and Tools
e. Creating additional modes	Game blueprint, expectations maps, and player lifecycle maps
f. Design levels	Game blueprint, player lifecycle maps
g. Writing the story	Game blueprint, player lifecycle maps
h. Build, test, and iterate	Paper prototype, simulation prototypes, net promoting scores and semantic scale, player storytelling
3. Tuning – No new features, only small adjustments to polish the game “before the game launch.”	Beta test prototype, net promoting scores, semantic scales, heuristic evaluation, task analyses, grid and usability testing, UX questionnaires, interviews, storytelling

E. Adams's game design process 4/4

Design Stages	Methods and Tools
4. Agile Development – Continuously scanning user behaviors, feedback, engagement, and satisfaction.	Game analytics, focus group, interviews, design labs, ethnographic research, futures thinking, collaborative and co-creative tools (such as forums, communities, templates, design patterns, APIs, and a fan art page)





An illustration of an Agile design process, adapted from Unger and Novak (2011)

Game mod's

- mod, or modification, is the alteration of content from a video game in order to make it operate in a manner different from its original version.
- Types:
 - *total conversion* - replaces virtually all of the artistic assets in the original game, and sometimes core aspects of gameplay
 - *total overhaul* mod changes or redefines the gameplay style of the original game
 - *add-on* or *addon* - a typically small mod which adds to the original content of a specific game
 - *unofficial patch* can be a mod of an existing game that fixes bugs not fixed by an official patch or that unlocks hidden content
 - *art mod* is a mod that is created for artistic effect

Source: [https://en.wikipedia.org/wiki/Mod_\(video_gaming\)](https://en.wikipedia.org/wiki/Mod_(video_gaming))

Game engine

- A game engine is a software framework specially created for development of 2D or 3D video games (for consoles, mobile devices, Web, iTV, and personal computers).
- Core functionality:
 - rendering engine (“renderer”) for 2D or 3D graphics
 - a physics engine or collision detection and response
 - sound
 - scripting
 - animation
 - AI, networking, streaming, memory management, threading, localization support, scene graph, video support for cinematics....

Modern game engines

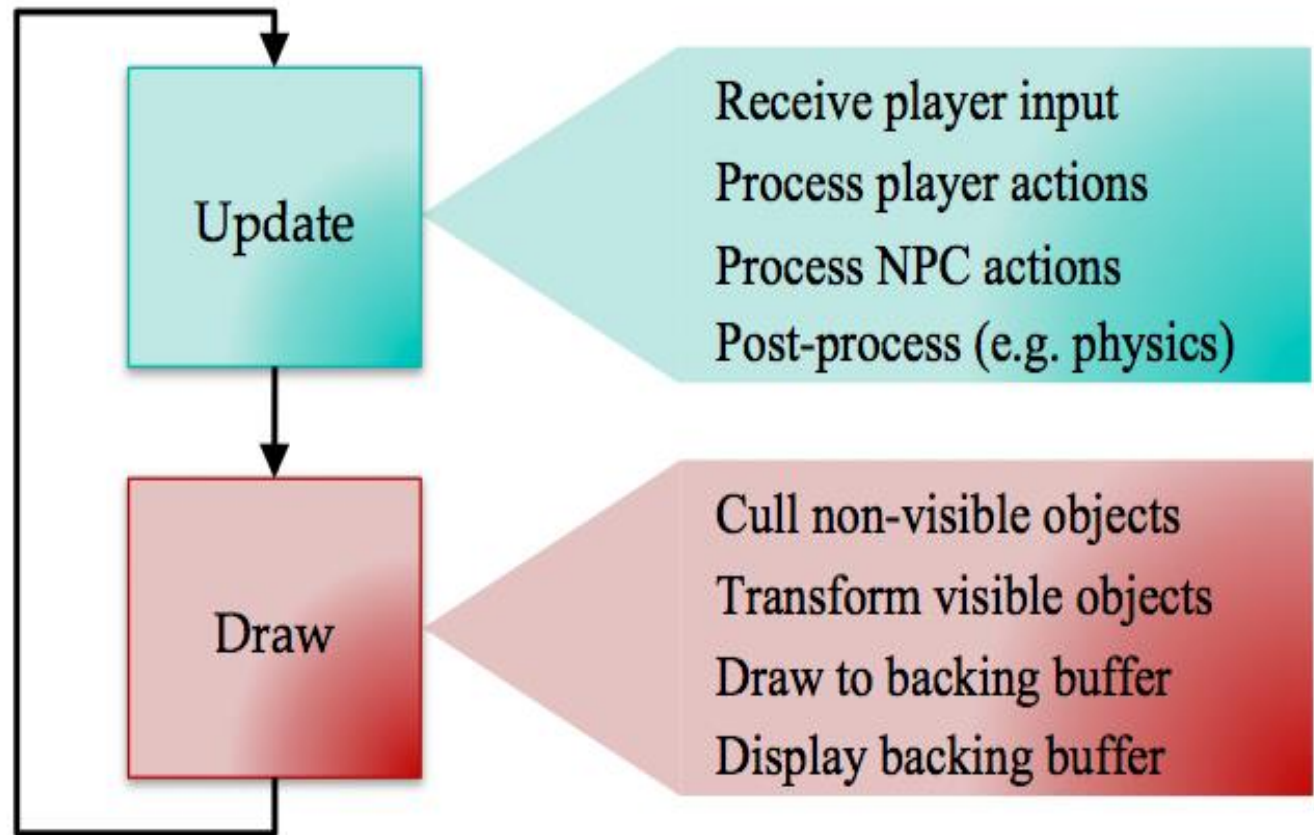
Name	Desktop targets	Dev platforms	Mobile targets
#1 AppGameKit 2.0 ...	Windows, Mac, Linux	Windows, Mac, Linux	iOS, Android
#2 Unreal Engine 4 ...	Windows, Mac OS X, Linux, SteamOS, HTML5	Windows, Mac OS X, Linux	iOS, Android
#3 Unity ...	Windows; OSX; Linux;	Windows; OSX; eucalyptus	Windows Phone; iOS; Android; BlackBerry 10; Tizen
#4 Panda3D ...	Windows; OSX; Linux	Windows; OSX; Linux; FreeBSD	-
#5 Godot ...	Windows; OSX; Linux; HTML5	Windows; OSX; Linux	iOS, Android, BB10
#6 GameGuru ...	Windows	Windows	none

Source: <https://www.slant.co/topics/1495/~3d-game-engines>

See: top 10 engines - <http://venturebeat.com/2014/08/20/the-top-10-engines-that-can-help-you-make-your-game/>

The Game Engine Loop

60 times/s
=
16.7 ms

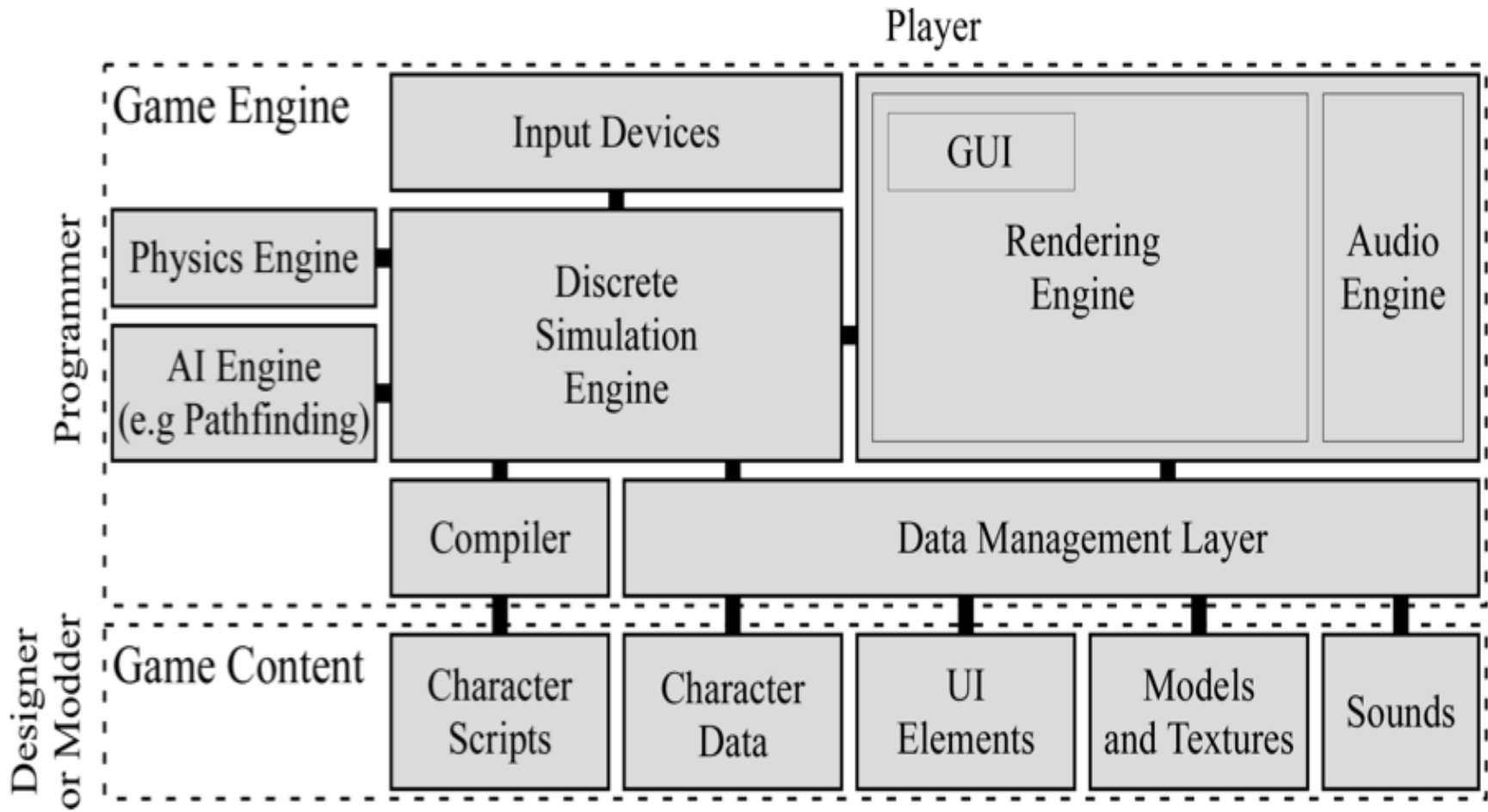


Credit: Walker White (Cornell Univ.)

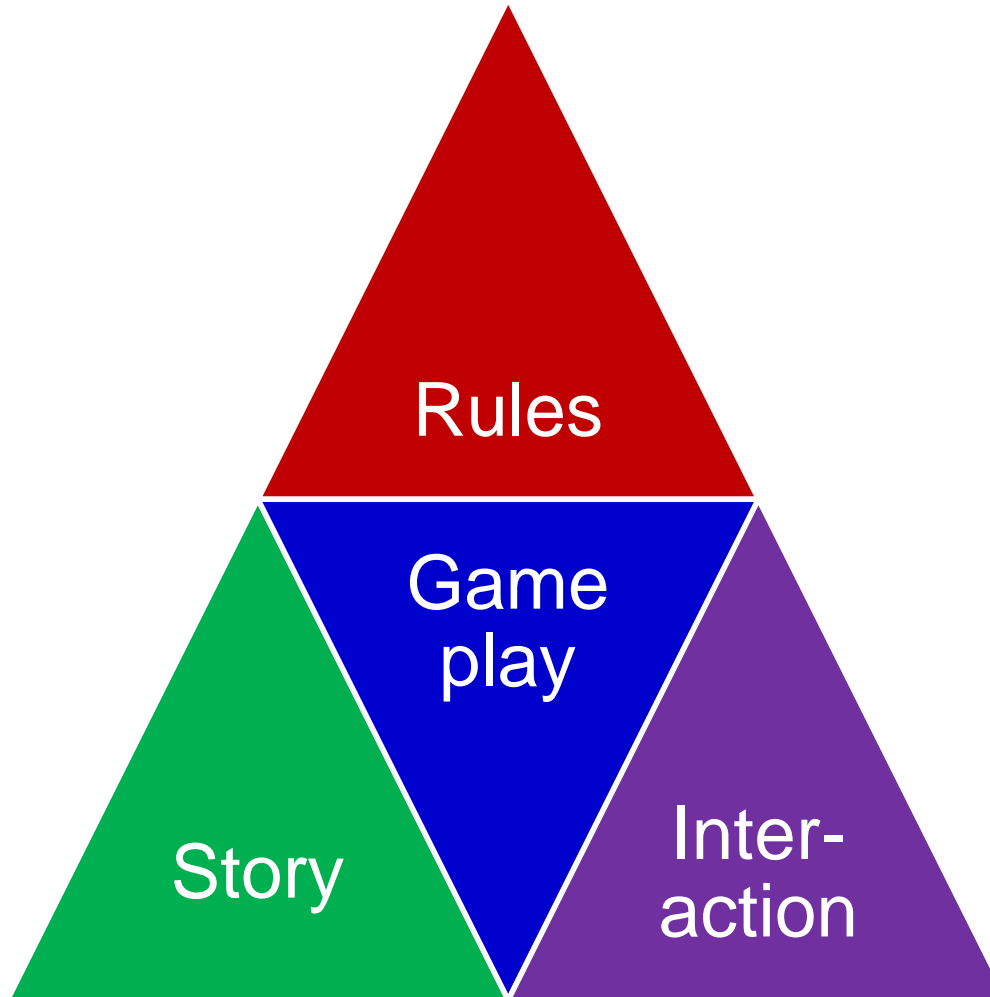
Remarks on processing NPCs

- An NPC (Non-Player Character) is anything that has volition in the world that isn't you
- NPCs take input from the AI engine (maybe!) but not from a direct controller
- Work on the idea of Sense-Think-Act:
 - **Sense** (recognize) the state of the world around it (how can we “cheat” here to make an NPC “harder”?)
 - **Think** about what action to perform (usually limited choices). Thinking is hard!
 - Can take more than one frame to decide what to do!
 - Act without thinking?
 - What if one acts, then the next acts on that action?
 - **Act** in the world

Game architecture

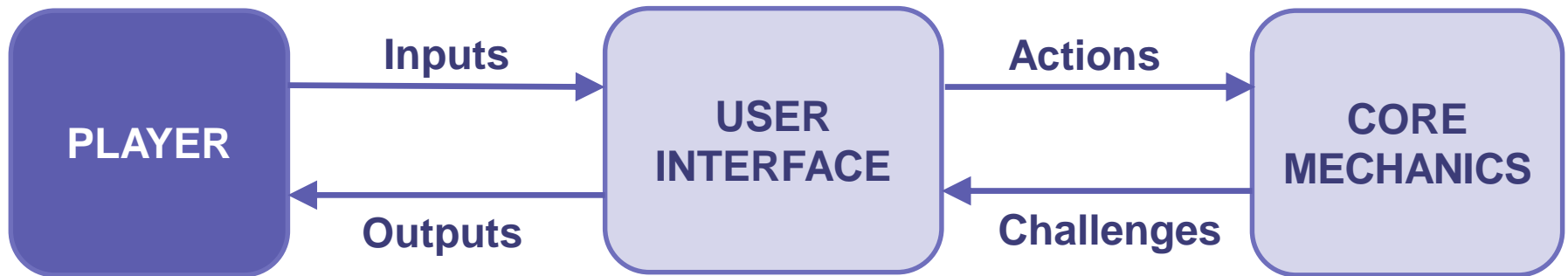


Core issues in game design



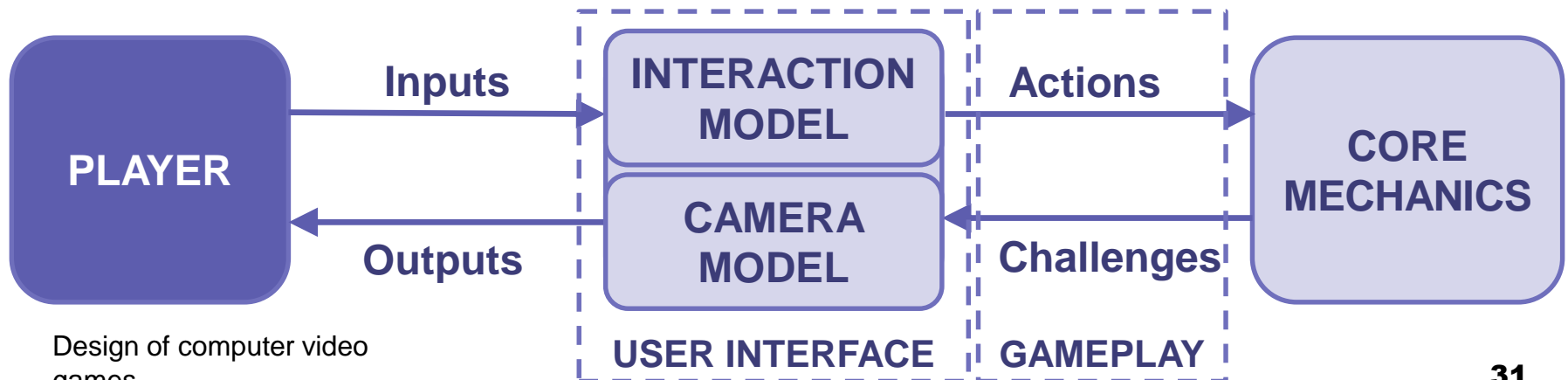
Game user interface

- Most computer (editors, web-browsers, painting tools, etc.) are designed to be as efficient as possible and to present the user's work clearly.
- Games are different because the player's actions are *not* supposed to be as efficient as possible; they are obstructed by the challenges of the game.
- Most games also hide information from the player, revealing it only as the player advances. A game's user interface is supposed to entertain as well as to facilitate.



Interaction models

- The relationship between the player's inputs and the resulting actions is dictated by the game's *interaction model*.
- It defines what she may and may not act upon at any given moment.
- Video games use standard interaction models, such as:
 - *Multi-present model* – the player can act on different parts of the game world whenever she wants to, reaching “into” it from the “outside.”
 - *Avatar-based model* - the player is represented by a character who already *is* inside the game world, and the player acts on the world through that character.



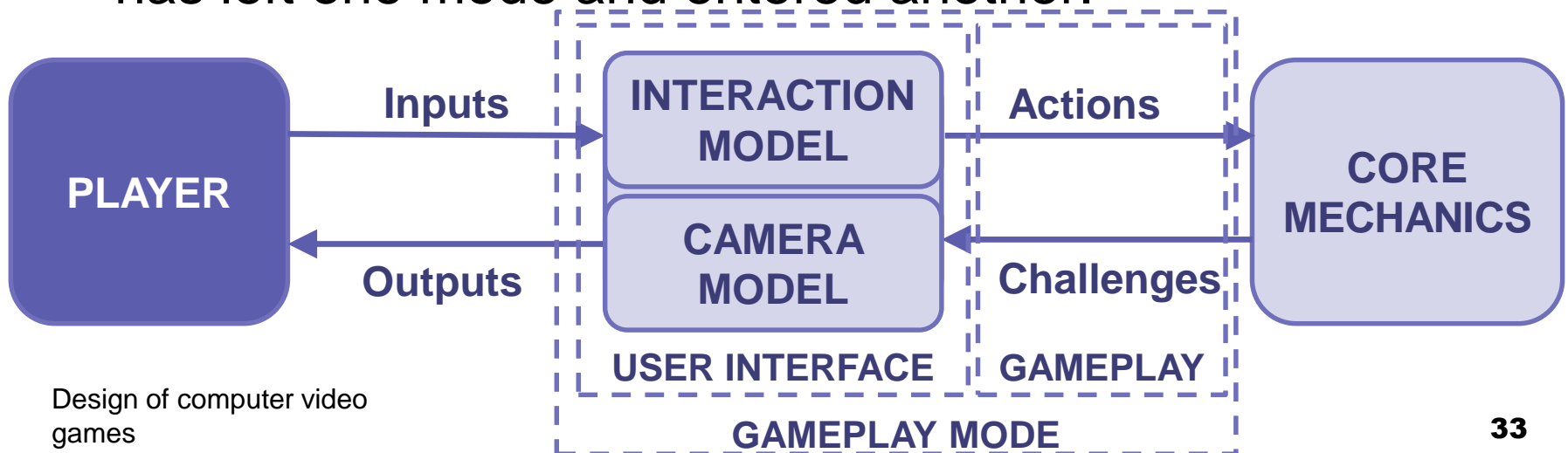
Camera model

- The *game world* is graphics-based simulated physical space
- The user interface must display the space from a particular angle or point of view.
- Designers usually imagine that a hypothetical camera is pointed at the virtual space, creating the image that the player sees.
- The system that controls the behavior of this imaginary camera is called the *camera model*. Types:
 - static camera model - the camera always shows the virtual space from a fixed perspective
 - dynamic camera models - the camera moves in response to player actions or events; requires more effort to design and implement, but they make the player's experience more cinematic. Types:
 - FPS (first-person shooter) – started by DOOM in the 90's
 - TPS (third-person shooter)



Gameplay mode

- A *gameplay mode* consists of the particular subset of a game's total gameplay that is available at any one time in the game, plus the user interface that presents that subset of the gameplay to the player.
- A game can be in only one gameplay mode at a time.
- When either the gameplay available to the player or the user interface (or both) changes significantly, the game has left one mode and entered another.



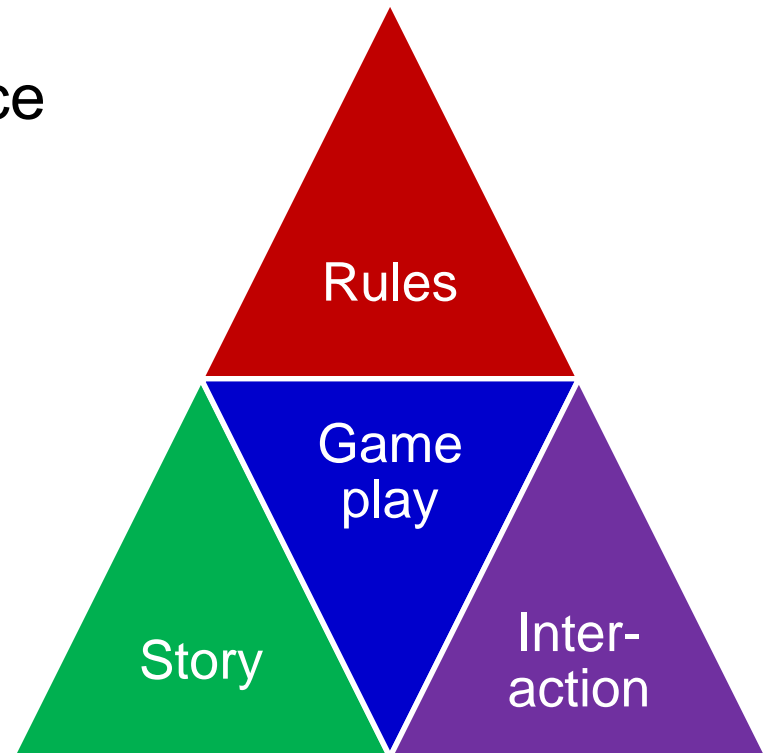
Contents of Design Documents

■ Basic information

- The premise of the game
- The game's intended audience
- The target game platforms
- The genre of the game
- The game's unique selling points

■ Gameplay elements

- Depending on the type of document it will contain more or less information on the **story, rules, and interactions**





Documenting Game Design

- Types of Game Design Docs
 - Concept Document
 - Proposal Document
 - Technical Specification
 - Game Design Document

Documenting Game Designs

■ Concept Document

- Used to explore game idea in more detail
- Often used as a proposal within an organization
- Developed by designer or visionary
- A short sales pitch - 1-3 pages
- May have no art, or amateur art
- Many ideas never get farther than this

Concept Doc

- Must include:
 - Intro
 - Description
 - Key features
 - Genre, spin, flavor
 - Platform(s) / market data
- May also include:
 - Background / License info
 - Concept art

Technical Specification

■ Presents:

- The 'How' of game design
- Contains the architectural vision; technology to be used
- Engineering detail
- Production detail
- Owned by tech director or chief engineer
- Can be exhaustive (and exhausting): 10-100 pages

■ Must include:

- Tooling
- Art / Music / Sound / Production pipeline
- Technology details: platform & portability issues, networking or special tech, server details, software engineering info

■ Key areas of technical risk

Game Design Document (GDD)

■ Game Design Doc

- Functional spec: The 'What' of the design
- Describes the player's experience and interactions in detail
- Could be quite long -- several hundred pages
- but "enough" is the goal.
- Artistic feel
- Owned by the game designer

■ Game Design Doc Must Haves

- Game mechanics
- User Interface
- Visuals
- Audio
- Story (if any)
- Level Specs

Game design document templates

Useful links

- https://www.google.bg/url?sa=t&rct=j&q=&esrc=s&source=web&cd=10&cad=rja&uact=8&sqi=2&ved=0ahUKEwiUkb3Lq6fSAhVICMAKHuZDQoQFghcMAk&url=http%3A%2F%2Fwww.photonstorm.com%2Fdownloads%2FCSC_GDD.pdf&usg=AFQjCNFzCzehlbwYX3PDNkfm0N4fVGUrqA&bvm=bv.147448319,d.bGs
- https://www.google.bg/url?sa=t&rct=j&q=&esrc=s&source=web&cd=8&cad=rja&uact=8&sqi=2&ved=0ahUKEwiUkb3Lq6fSAhVICMAKHuZDQoQFghLMAc&url=http%3A%2F%2Fwww-personal.engin.umd.umich.edu%2F~bmaxim%2Fcis488%2FBaldwinGameDesignDocumentTemplate.doc&usg=AFQjCNEKwxb0SDkdvJYhGQbBj8T_ZBFrdQ&bvm=bv.147448319,d.bGs
- <https://s3-eu-west-1.amazonaws.com/kurapov/file/191.doc>



Game studios – vertical structure

- Developers
 - Publishers
 - Distributors
 - Retailers
-
- Much like a mini-Hollywood

Developers

- *Design and implement games*
 - Including: **programming**, **art**, **sound effects**, and **music**
 - Historically, small groups
 - Analogous to book authors
- Structure varies
 - May exist as part of a Publisher
 - May be “full-service” developers or may outsource some
 - Motion Capture (to replicate realistic movement)
 - Art and Animation (can be done by art house/studio)
- Many started on PC games (console development harder to break into)
- Typically work for royalties & funded by advances
 - Do not have the capital, distribution channels, or marketing resources to publish their games
 - Often seen that developers don't get equitable share of profits
 - Can be unstable

Publishers

- *Fund development of games*, including manufacturing, marketing/PR, distribution, and customer support
- If developers are the “**geeks**”, publishers are the “**suits**”
- Various specialties: PC only, PC + console, mobile, import, web
- Publishers assume most of the risk, but they also take most of the profits
- Console/PC publishers handle:
 - Production process
 - Quality assurance
 - Licensing
 - Manufacturing and shipping to retail
 - Sales
 - Consumer marketing and PR
 - HR, finance, investor relations, legal

Distributors

- *Get software from publisher to retailer*
- Originally modeled on book distribution
- May resell to smaller independent stores and chains
- Compete on price, speed and availability
- Earn profit margin of around 3%
- Becoming less important as the retail market changes

Retailers

- *Sell software*
- Started with mail-order and computer specialty stores
- Shift in 80's to game specialty stores, especially chains (Today 25%)
 - *EB Games, GameStop*
- Shift in 90's to mass market retailers (Today 70%) (ask)
 - *Target, WalMart, Best Buy*
- Retailers generally earn 30% margin on a \$50 game
- Electronic download of games via Internet



Developer and Publisher Relationship

The Pitching Process: Prototype

- Key game prototype features:
 - Core gameplay mechanic
 - Game engine / technological proficiency
 - Artistic / styling guide
 - Demonstration of control / camera system
 - Example gameplay goals



The Pitching Process: Pitch Presentation

- Key pitch presentation content:
 - Concept overview & genre profile
 - Unique selling points
 - What makes it stand out from its competitors
 - Proposed technology & target platform/s
 - Team biographies & heritage
 - Outline marketing information, including potential licensing opportunities

The Pitching Process: Design

- Game Design - focuses on intimate detail such as:
 - Storyline
 - Control dynamics
 - Camera system
 - Level progression
 - Game features and functionality
 - Score systems etc.
- Technical Design - covers technical topics:
 - Graphics engine
 - AI routines
 - Audio system
 - Online capability and requirements
 - Peripherals/controllers
 - Development asset management/backup



The Pitching Process: Project Schedule & Budget

- Schedule & budget must:
 - Be detailed and transparent
 - Allow for contingency scenarios
 - Have several sets of outcomes for different size publishers
 - Be realistic

Deal Dynamics: IPR

- Intellectual Property Rights include:
 - Game name
 - Logos
 - Unique game mechanics & storyline
 - Unique characters, objects & settings
 - Game Source Code including artwork & associated assets
 - Unique sounds and music

Payment Negotiation: Overview

- Current approximate development costs:
 - \$4-7 million for AAA multi-platform
 - \$2-3 million for AAA PlayStation only
 - \$1 million for A-quality single platform

In the early days, a publisher graded the games in their portfolio with simple a code.

“A” titles were made by an in-house team, under direct control of an experienced producer, with high quality standards and often a new IP or sequel to a previous hit. The majors.

“B” titles were lesser titles, made by 3rd party teams, elsewhere, supervised by associate producers. Farm league.

“C” products were add-on content like new maps, editors, soundtracks, cluebooks, ports or compilations.

Marketers could see a list of games in progress and plan ad budgets. As marketers often do, they began amplifying so the bigger of two “A” titles became “double A”. The idea spread and eventually grew to include triple A.

Moving Projects Forward

- Most Publishers have a “Greenlight Process”
 - Use to determine which projects go forward
- Developers submit to committee at five, mostly independent stages:
 1. Concept
 2. Assessment
 3. Prototype
 4. First Playable
 5. Alpha
- At each stage, committee reviews:
 - Decides whether or not to continue funding
 - Evaluates market potential
 - Adjusts unit forecasts accordingly

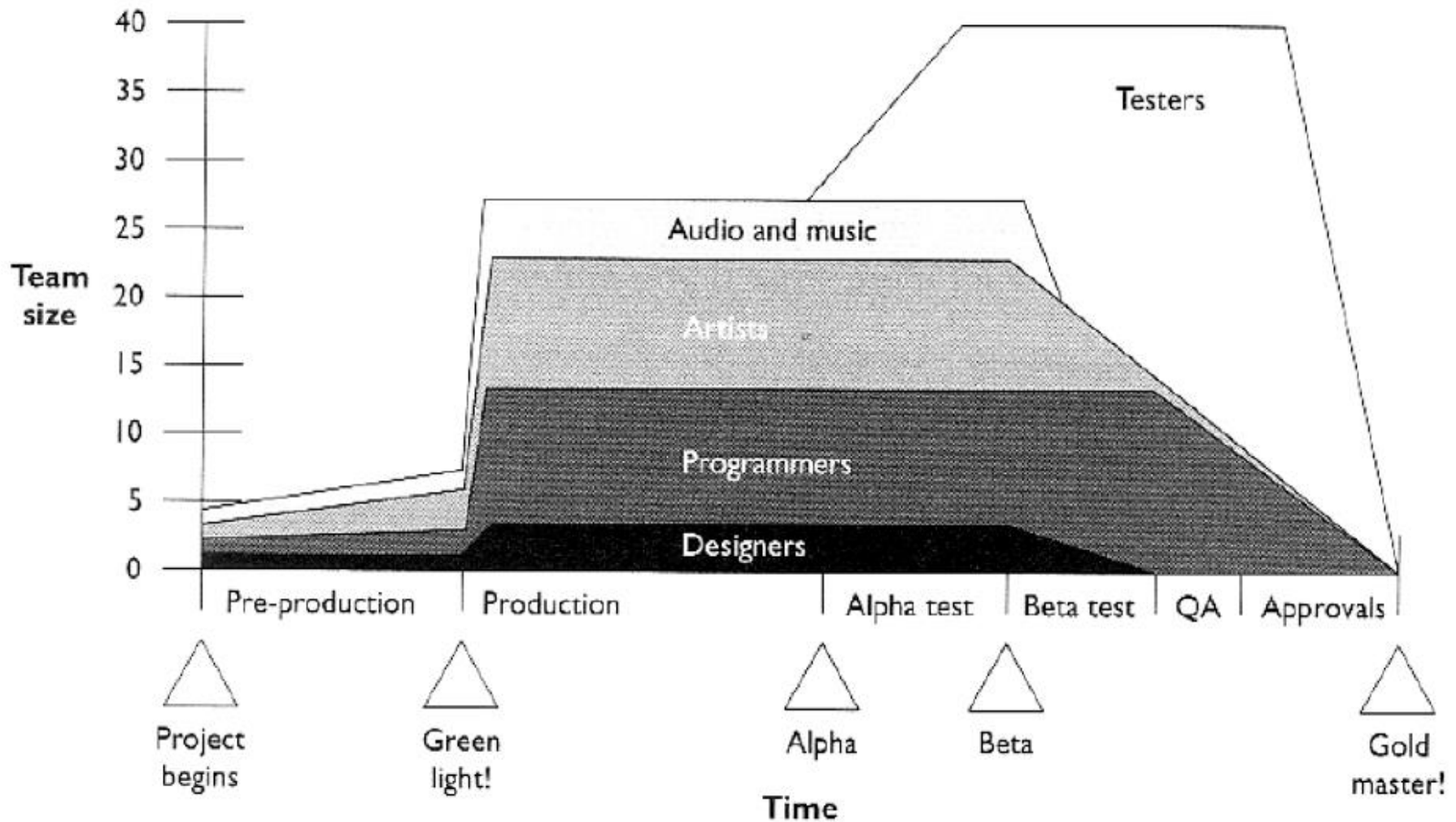
Development Milestones: Development Timeline

- Here are some example development periods for different platforms:
 - 4-6 months for a high-end mobile game
 - 18-24 months for an original console game
 - 10-14 months for a license / port
 - 16-36 months for an original PC Game

What's Involved?

- People involved
 - lead designer
 - project leader
 - software planner
 - architectural lead
 - programmers
 - artists, modellers, animators
 - musicians, sound engineers, actors
 - level designers
 - testers

Game Development Process



Game Development Timeline 1/5

■ Inspiration

- getting the global idea of the game
- duration: **1 month** (for a professional game)
- people: lead designer
- result: treatment document, decision to continue

■ Conceptualization

- preparing the "complete" design of the game
- duration: **3 months**
- people: lead designer
- result: complete design document
 - defines Game Concept
 - defines Core Game Features
- find/assign developers
- estimate budget & due date

Game Development Timeline 2/5

■ Prototypes

- Build prototypes (from 1 up to 5, 2-3 months) as proof of concept
- In particular to test game play
- Throw them away afterwards

■ While prototyping:

- Game Design Document (GDD) & Technical Design Document (TDD) = “The Bibles”
- Production Budget & Detailed Schedule
- Submit Concept to publisher
- Working Prototype, with Game Mechanics
- Focus Test
- Pitch to Publisher

Game Development Timeline 3/5

■ Blueprint

- separate the project into different tiers
- duration: 2 months
- people: lead designer, software planner
- result: several mini-specification

■ Architecture

- creating a technical design that specifies tools and technology used
- duration: 2 months
- people: project leader, software planner, lead architect
- result: full technical specification

Game Development Timeline 4/5

■ Tool building

- create a number of (preferably reusable) tools, like 3D graphics engine, level builder, or unit builder
- duration: 4 months
- people: project leader and 4 (tool) programmers
- result: set of functionally tools (maybe not yet feature complete)

■ Assembly

- create the game based on the design document using the tools; update design document and tools as required (consulting the lead designer)
- duration: 12 months
- people: project leader, 4 programmers, 4 artists
- result: the complete game software and toolset

Other Development Milestones: Alpha Definition

- At Alpha stage, a game should:
 - Have all of the required features of the design implemented, but not necessarily working correctly
 - Be tested thoroughly by QA to eliminate any critical gameplay flaws
 - Still likely contain a certain amount of placeholder assets
 - “Localization” Begins
 - Focus Test
 - Play Testing
 - Marketing Continues

Game Development Timeline 5/5

■ Level design

- create the levels for the game
- duration: 4 months
- people: project leader, 3 level designers
- result: finished game with all levels, in-game tutorials, manuals

■ Review

- testing the code, the gameplay, and the levels
- duration: 3 months (partially overlapping level design)
- people: 4 testers
- result: the gold master

Other Development Milestones: Beta Definition

- At Beta stage, a game should:
 - Have all content complete
 - Be tested thoroughly for bugs and gameplay tweaks
 - Be shown to press for preview features
- Beta tasks:
 - Polish, Polish, Polish
 - Game Balancing
 - Localization Continues
 - Demo Versions

Other Development Milestones: Gold Master Definition

- At Gold Master stage, a game should:
 - Be sent to the platform holder/s (where applicable) for Technical Requirements Checklist (**TRC**) testing
 - Be sent to press for review
 - Be sent to duplication for production
 - Be backed up and stored
- Gold Master tasks:
 - The Game is “Done”
 - Testing, Testing, Testing
 - Intense Pressure
 - Submit to Console developers

Post-Mortem analysis

- Analysis of PR and marketing

- Analysis of production / source code

- Archive all game assets

- What went **right**, what went **wrong**

- Kick-off the game sequel!

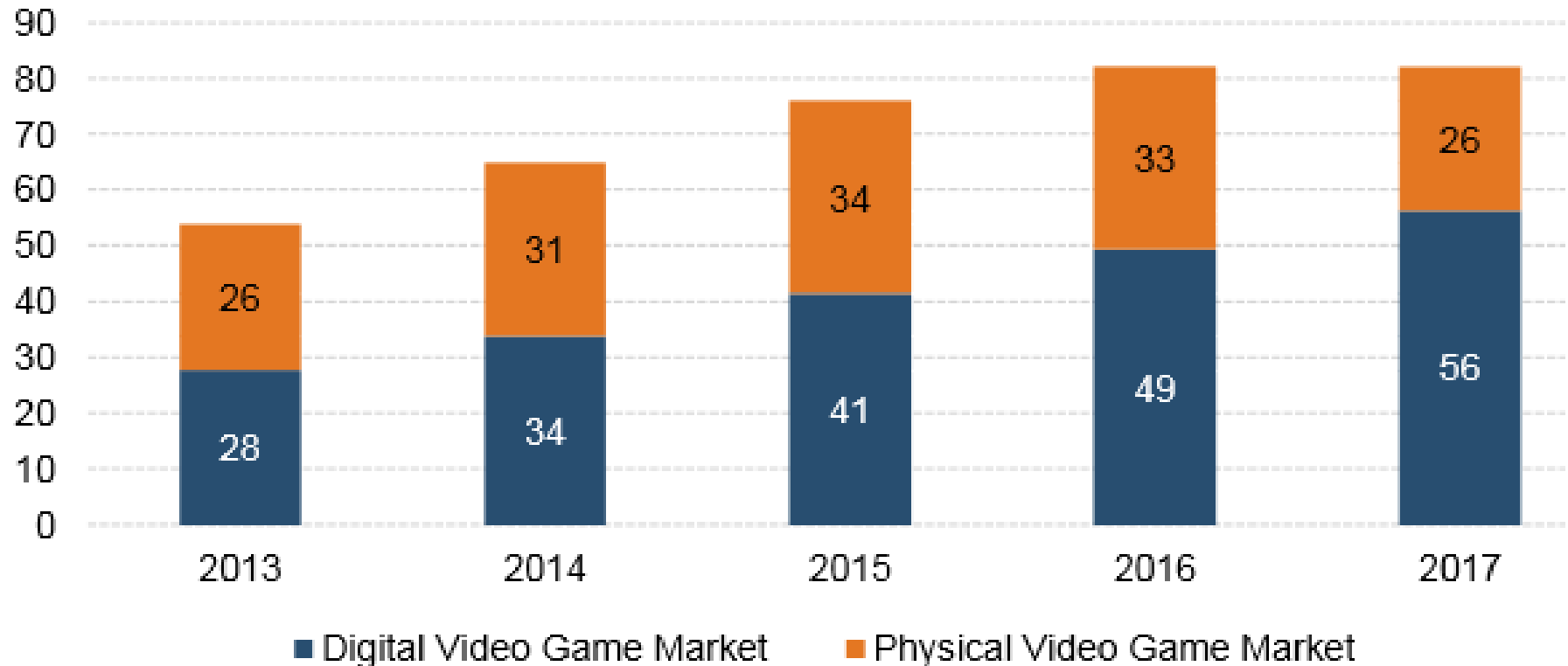


Game industry trends 1/5

- The arrival of next home consoles will boost the market up to 2017
 - Lower price
 - Simple multi-platform, cloud-based access to the play environment
 - Community character:
 - social ties
 - sharing content and experience
 - Visual and renewable off-game content – via TV, VOD services, web browsing, messaging services, and video capture

Game industry trends 2/5

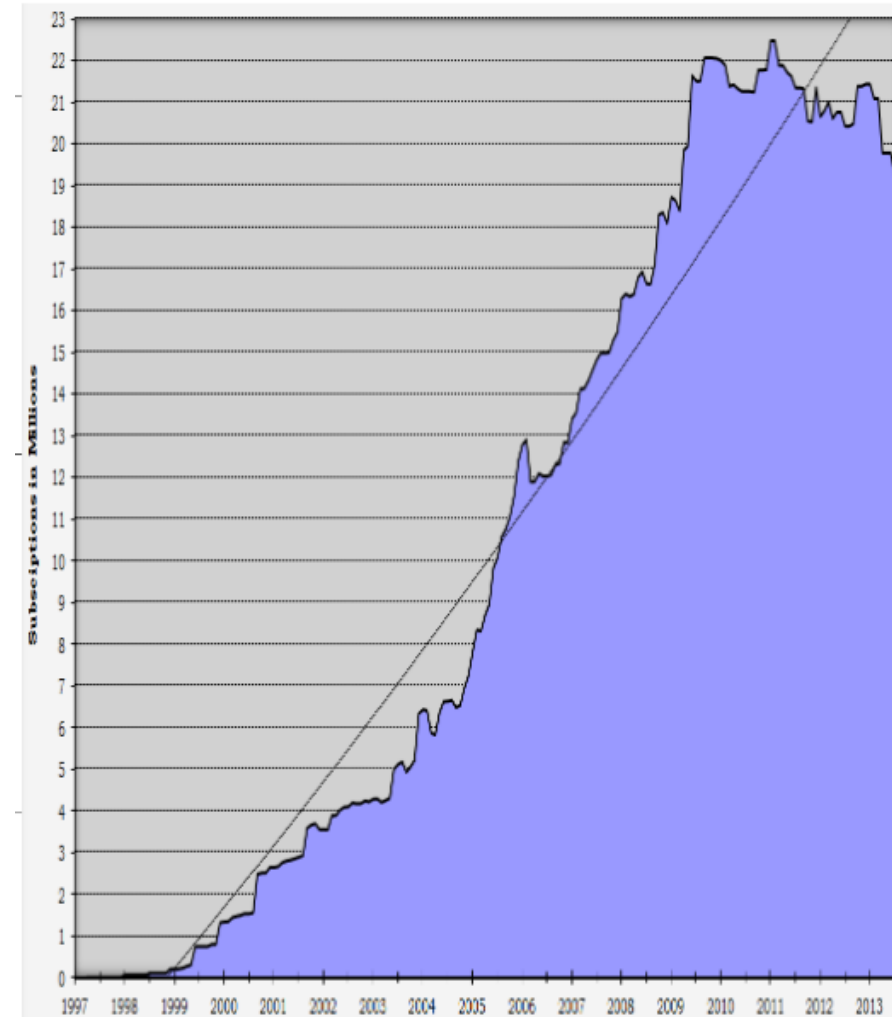
- From 1/2 up to 2/3 video game market revenue generated by online software sales and paid gaming



Game industry trends 3/5

Total MMORPG subscriptions and active accounts listed in mmodata.net

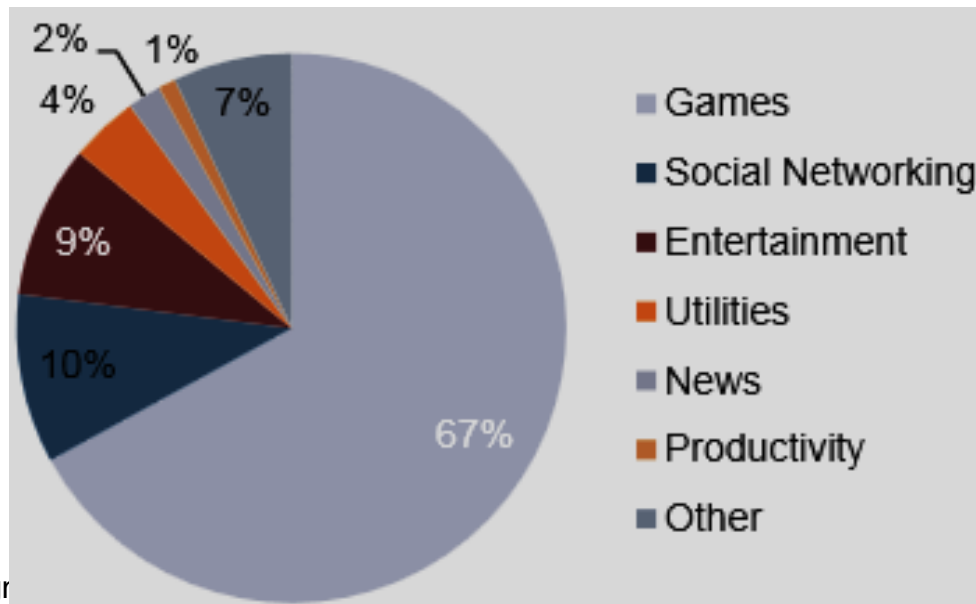
- Transition from paid subscription model to Free2Play gaming:
 - Clash of Clans and Hay Day (Supercell) : 8.5 million gamers per day, a daily turnover of 2.4 million USD
 - Candy Crush Saga (King.com): more than 700 million game sessions per day, a daily turnover of 630 million USD



Source: IDATE. November 2013

Game industry trends 4/5

- Intense competition in the segment of mobile games - Smartphones are getting to compete with handheld/home consoles
- Increasing attraction of tablets for gamers



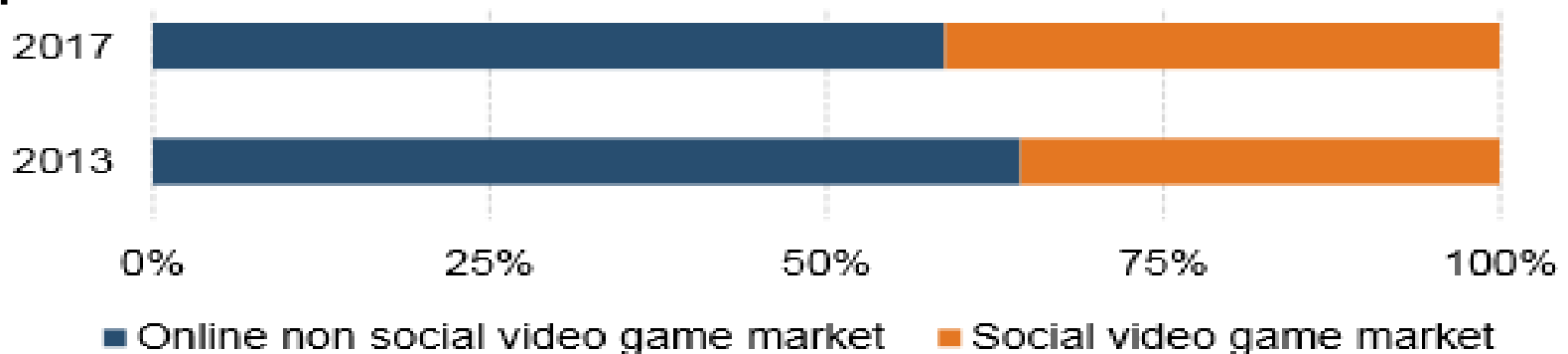
Design
games

Game development

Source: IDATE. November 2013

Game industry trends 5/5

- Close-up on social gaming - big players spend more than 25 USD per month and are less than 15% of all gamers but generate over 50% of publishers' revenues

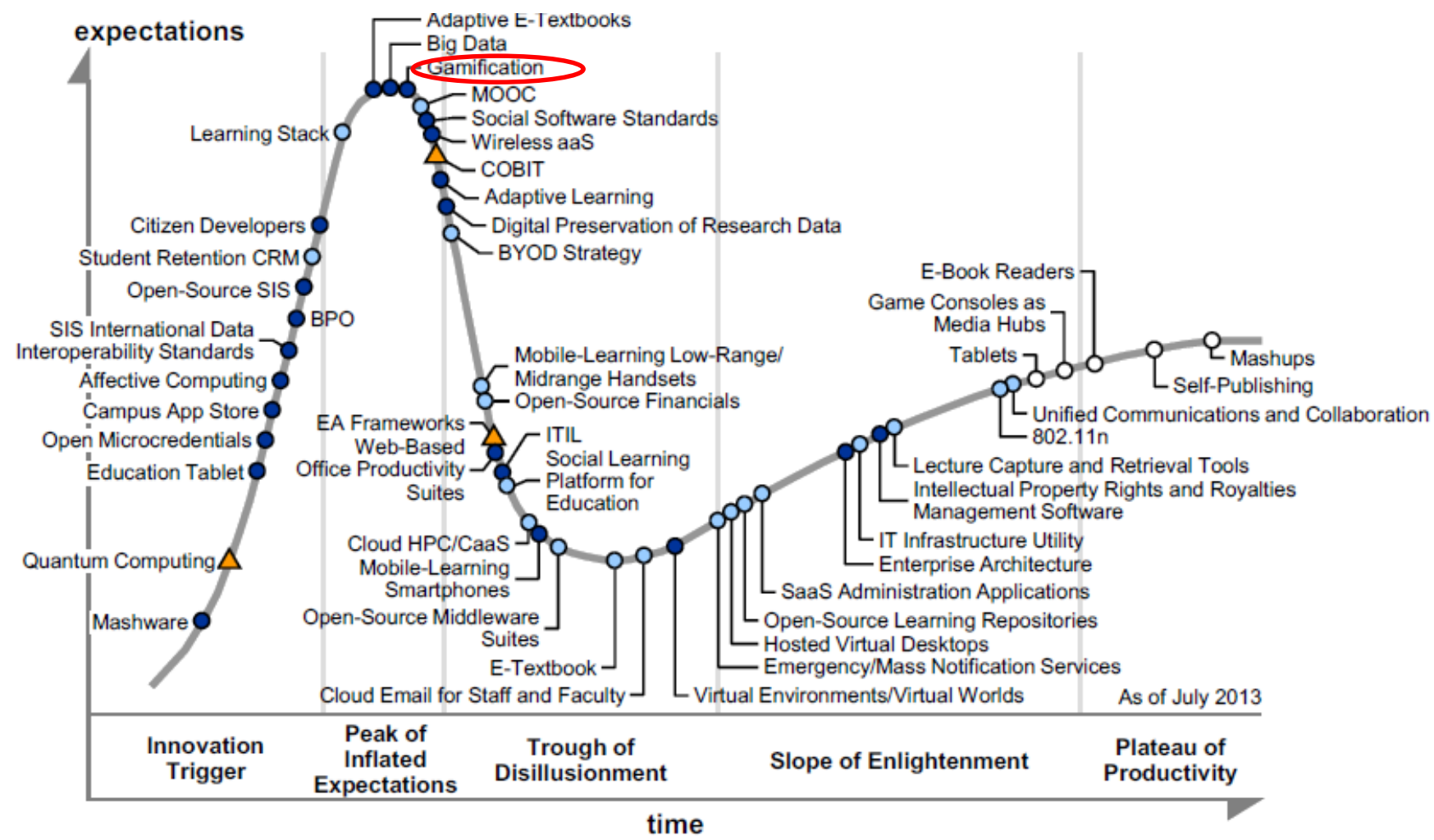


- Irreversible ubiquity - most of the games ranked in the top 20 of the titles sold on the Apple Store are ubiquitous and accessible either on iPad and

Gamification 1/2

- Gamification - game design techniques, thinking and mechanics applied to enhance non-game contexts
- Used for non-game applications and processes
- Gamification is using game mechanics outside of game – it is not an *advergame* (a casual game to advertise a brand) nor another serious game
- Used to change behavior, develop skills or drive innovation, in customer engagement, education, employee performance, innovation management and health care

Gamification in Gartner hype cycle (2013)



Plateau will be reached in:

- less than 2 years
- 2 to 5 years
- 5 to 10 years
- ▲ more than 10 years
- ⊗ obsolete before plateau

games

Game development

