Design of computer video games

5. User interface and experience

Boyan Bontchev

Agenda

- User interface in computer games - definitions and basic concepts.
- Interaction models
- Camera models
- Perspectives
- Input devices
- User experience
- Playing styles
- Examples

References

- Adams, E. Fundamentals of Game Design, Third Edition, Pearson Education, Inc., ISBN-13: 978-0-321-92967-9, 2014
- Smith, M., Queiroz, C. Unity 4.x Cookbook, Packt Publishing, Birmingham B3 2PB, UK, ISBN 978-1-84969-042-3, 2013.
- Jose Luis González Sánchez, Natalia Padilla Zea, and Francisco L. Gutiérrez. From Usability to Playability: Introduction to Player-Centred Video Game Development Process, HCII 2009, LNCS 5619, pp. 65–74, 2009.
- Other references are cited in the text

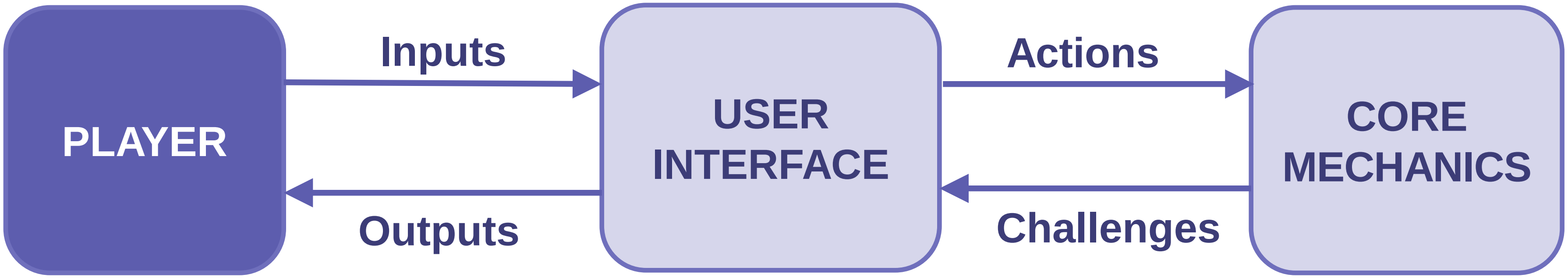
User Interface (UI) definitions

- “means in which a person controls a software application or hardware device” (https://techterms.com/definition/user_interface)
- “UI comprises the screen menus and icons, keyboard shortcuts, mouse and gesture movements, command language and online help, as well as physical buttons, dials and levers. Also included are the physical components, such as the mouse, keyboard, touchscreen, remote and game controllers” (http://www.pcmag.com/encyclopedia/term/53558/user-interface)
- “Visual part of computer application or operating system through which a user interacts with a computer or a software” (http://www.businessdictionary.com/definition/user-interface.html) – *only visual??*

From Module 2

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.



UI in computer games

UI in computer games refers to:

- *feedback elements*
  - visual, audio and textual output elements providing information to the player
- control elements
  - provide player input

UI in games is related to two main concepts:

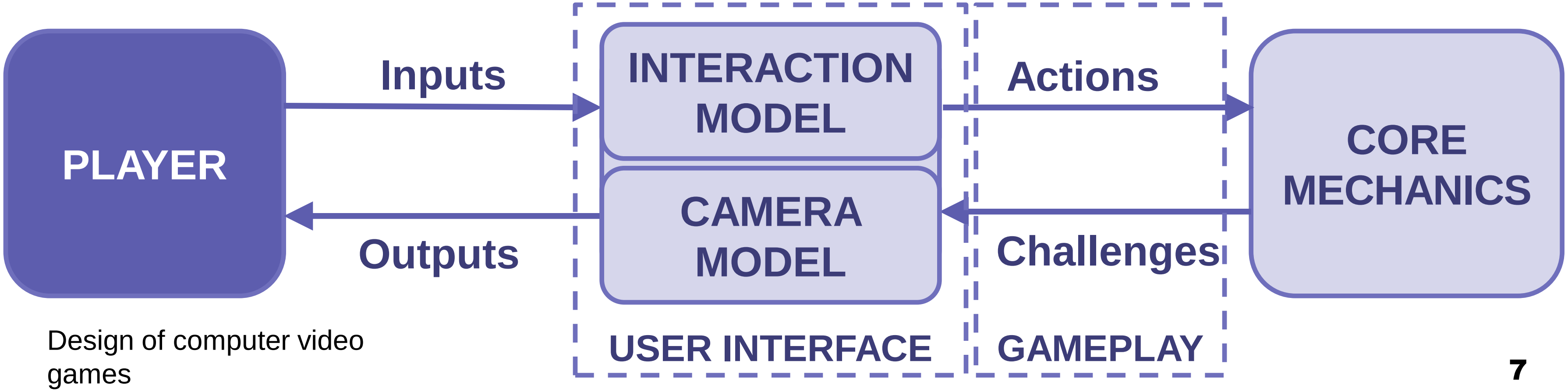
- *interaction models*
- *camera models*

Interaction and camera models

From Module 2

- The relationship between the player's inputs and the resulting actions is dictated by the game's *interaction model* - defines what she may and may not act upon at any given moment

- 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*.



Hint

- When designing the core game mechanics, avoid making choices that depend on the performance characteristics of particular input/output (I/O) devices.
- Let the UI manage the hardware, and keep the internals of the game hardware independent

UI in computer programs and games

- computer programs - tools with UI optimized to enter data, to control processes, and to view results
- video games – have UI to entertain people; must be easy to learn and use, but:
  - neither provide full view of the game,
  - nor give the player maximum control over the game.
- Game UI mediates between the internals and the player
- Create an experience for the player that feels to him like gameplay and storytelling.

UI principles 1/2

- **Be consistent** - applies to both aesthetic and functional issues; your game should be stylistically as well as operationally consistent.
- **Give good and immediate feedback –** when pressing any screen button, the game should produce an audible response even if the button is inactive at the time
- **Limit the number of steps required to take an action**
- **Permit easy reversal of actions** - if a player makes a mistake, allow him to undo the action if possible
- **Minimize physical stress -** video games famously cause tired thumbs, and unfortunately, repetitive stress injuries from overused hands

UI principles 2/2

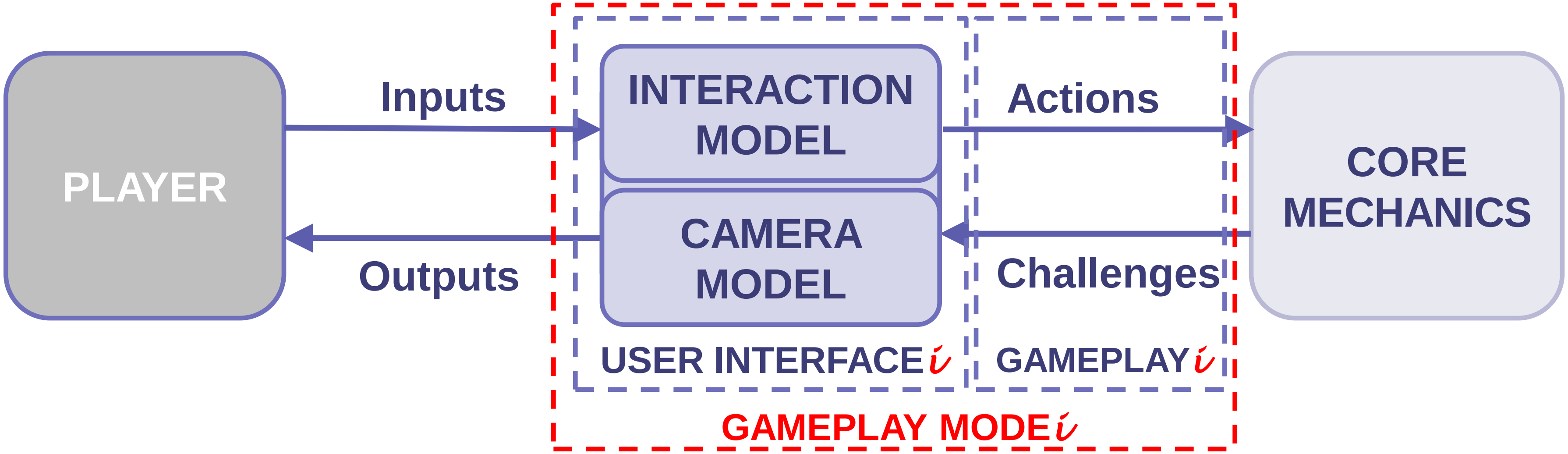
- **Don’t strain the player’s short-term memory** - don’t require the player to remember too many things at once; provide a way for him to look up information
- **Group related screen-based controls and feedback mechanisms on the screen** – thus, the player can take in the information he needs in a single glance
- **Provide shortcuts for experienced players** - experienced players don’t want to go through multiple layers of menus to find the command they need. Provide shortcut keys to perform the most commonly used actions from the menus, and include a key-reassignment customization feature
- Think about ***What the Player Needs to Know (?)***
- Think about ***What the Player Wants to Do (?)***

What the player wants to do?

- Move
- Watch/investigate
- Interact physically with/talk to non-player characters (NPC’s)
- Pick portable objects up and put them down
- Manipulate fixed objects
- Construct, customize and demolish objects
- Conduct negotiations/financial transactions, and set values
- Give orders to units or characters
- Talk to friends in networked multiplayer games
- Set game options
- Pause, save, resume and end the game

What is the first designer job?

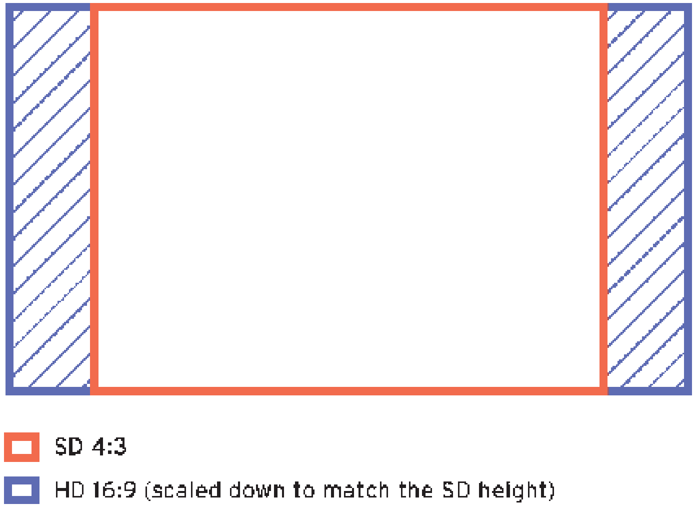
- RULE: First job is to design the *primary* gameplay mode

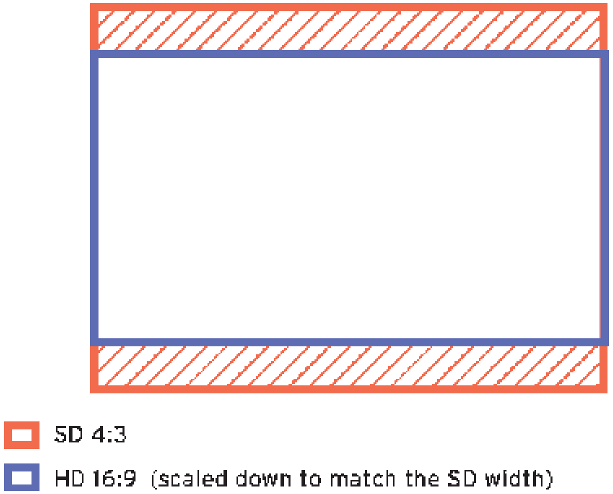


- Only AFTER the *primary* gameplay mode – details of its UI
- Gameplay modes *typically* share a number of UI features, so it's best to define all the modes before you begin UI
- In case of many modes - *wait* until you have planned the game structure and game moves from mode to mode

Choosing a game screen layout

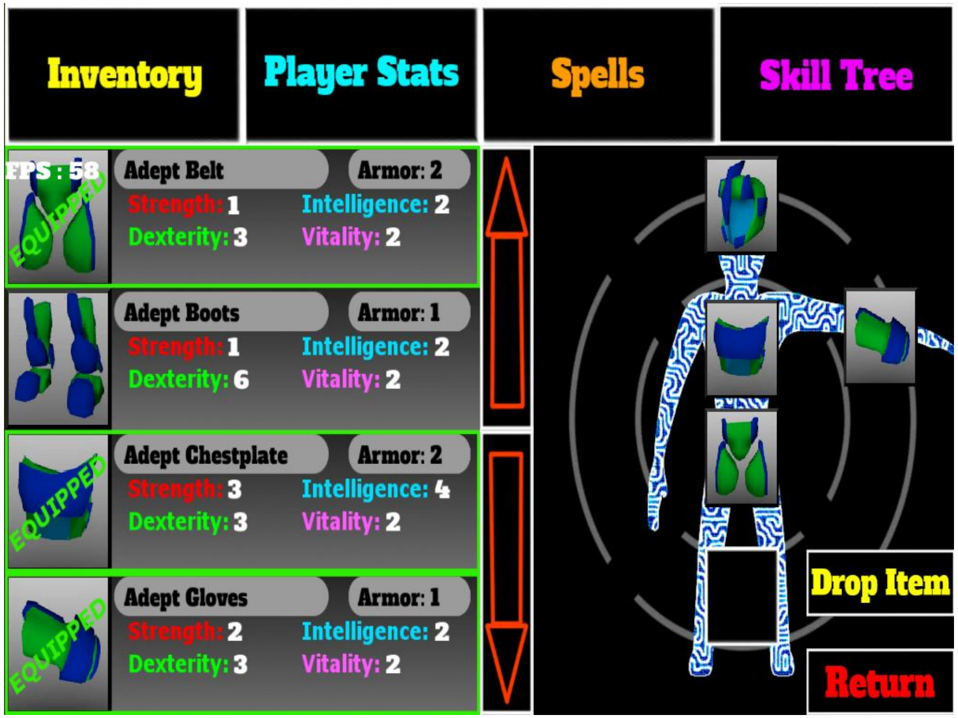
- Proportions
  - □ 4:3
  - □ 16:9
  - □ 1920x1080
  - □ …





Balance your screen layout

- Screen balance between:
  - □ game main view
  - □ on-screen controls
  - □ feedback elements



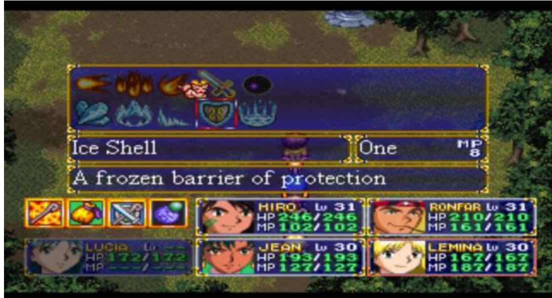
Adept Belt Armor: 2
Strength: 1 Intelligence: 2
Dexterity: 3 Vitality: 2

Adept Boots Armor: 1
Strength: 1 Intelligence: 2
Dexterity: 6 Vitality: 2

Adept Chestplate Armor: 2
Strength: 3 Intelligence: 4
Dexterity: 3 Vitality: 2

Adept Gloves Armor: 1
Strength: 2 Intelligence: 2
Dexterity: 3 Vitality: 2

- Balance b/w *real game playing* and *shell menus*
  - □ shell menus allow the player to start, configure, and manage the operation of the game before and after play - video and audio settings and the game controls
  - □ time and UI of shell menus are to be balanced



Ice Shell One MP 8
A frozen barrier of protection

HIRO Lv 31 HP 246/246 MP 102/102
RONFAR Lv 31 HP 210/210 MP 161/161
LUCIA Lv HP 172/172 MP ---/---
JEAN Lv 30 HP 193/193 MP 127/127
LEMINA Lv 30 HP 167/167 MP 187/187

Managing complexity 1/3

- Try to simplify your game – by using:
  - □ *abstraction* – replace a more accurate and detailed version by a less accurate and detailed one
    - makes the game less realistic but easier to play
  - □ *automation* - remove some actions from the player’s control and let the computer handle it for her
    - the player may choose between automated or manual control over a game feature, resp. in beginner’s mode and expert mode
- Examples?

Managing complexity 2/3

- Depth vs breath
  - *broad* interface - provides a large number of options simultaneously
    - permit the player to search the whole interface (efficient, if he/she has to know the UI location – slower learning curve)
    - finding needed item in that broad array takes time
  - *deep* interface - offers only a few options at a time and require the player to make several selections
    - offers all its choices through a hierarchical series of menus or dialog boxes
    - name and organize them coherently
- Deep (for beginners) and a broad (for experienced players) interface at the same time

Managing complexity 3/3

- Context-sensitive interfaces
  - reduce complexity by showing the player only the options that she may actually use at the moment.
- Avoid obscure UI, especially in cases of:
  - artistic overenthusiasm
  - pressure to reduce UI screen usage – e.g., icons vs buttons
  - developer familiarity with the material

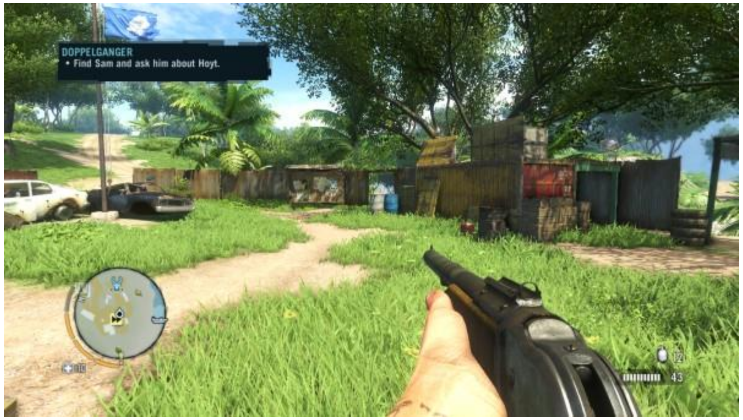
Interaction models

- 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
  - *Party-based interaction model* (in RPG) - small groups of characters remain together, so point-and-click navigation and an aerial perspective are used
  - *Contestant model* - the player answers questions and makes decisions; navigation is not mandatory
  - *Desktop model* - mimics a PC desktop in business simulation games

Camera models

- The system that controls the behavior of this imaginary camera is called the *camera model*. Types:
  - 2D, 2,5D, 3D - good-looking 2D graphics are preferable to bad 3D ones
  - 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 of 3D dynamic camera **perspectives**:
    - FPS (first-person shooter, only in 3D avatar-based gameplay) – the camera takes the position of the avatar's own eyes, and is fixed with respect to the avatar. The player doesn't usually see the avatar's body, though the game may display handheld items in avatar's hand
    - TPS (third-person shooter, only in 3D) - the camera normally follows the avatar at a fixed distance, remaining behind and slightly above her. To permit the player to see both the avatar and the enemies, the camera must crane up and tilt down to show the fight from a raised perspective.

FPS perspective – pro’s and con’s

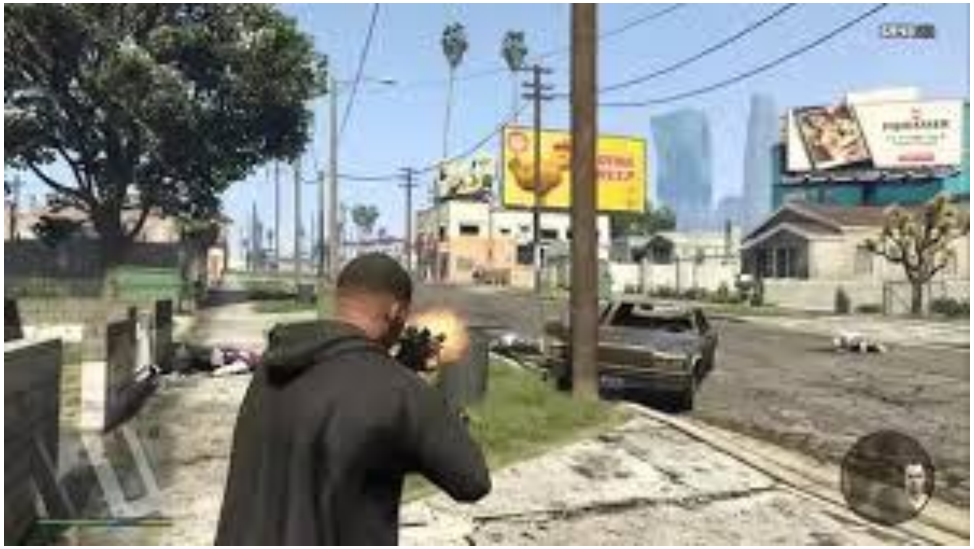


DOPPELGANGER
• Find Sam and ask him about Hoyt.

- Simple design
  - □ No animated avatars
  - □ No AI for controlling the camera
  - □ Easier interactions and positioning
- Less pleasure – the player cannot see and customize the avatar
- Less immersion – no avatar’s body language and facial expressions
- No opportunity to use cinematic camera angles for dramatic effect, like in Resident Evil
- Rapid camera movements may provoke sickness in the viewer. Difficult to judge distance when jumping.

TPS perspective – pro’s

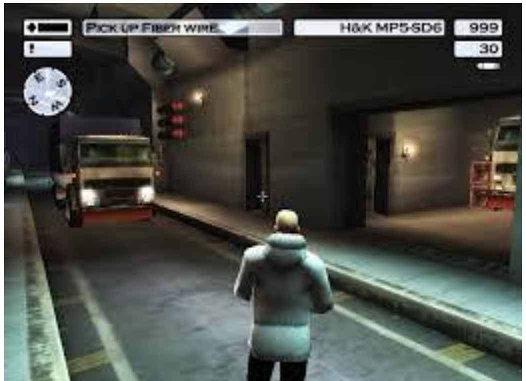
- Most commonly used in modern 3D action/adventure games
- Better user experience (UX):
  - □ the player can choose/customize the avatar
  - □ the camera normally follows the avatar at a fixed distance, remaining behind and slightly above



TPS perspective –con’s

- More complex implementation:
  - when the avatar turns, the camera points in the direction in which the avatar looks
  - treatment of intruding landscape objects - objects in the landscape (e.g., a wall) can intrude between the avatar and the camera. Possible solutions:
    - Place the camera immediately behind the avatar, between her and the wall, but

      crane it upward somewhat and tilt it down, so the player sees the area immediately in front of the avatar from a raised point of view
    - Rendering the wall as semitransparent

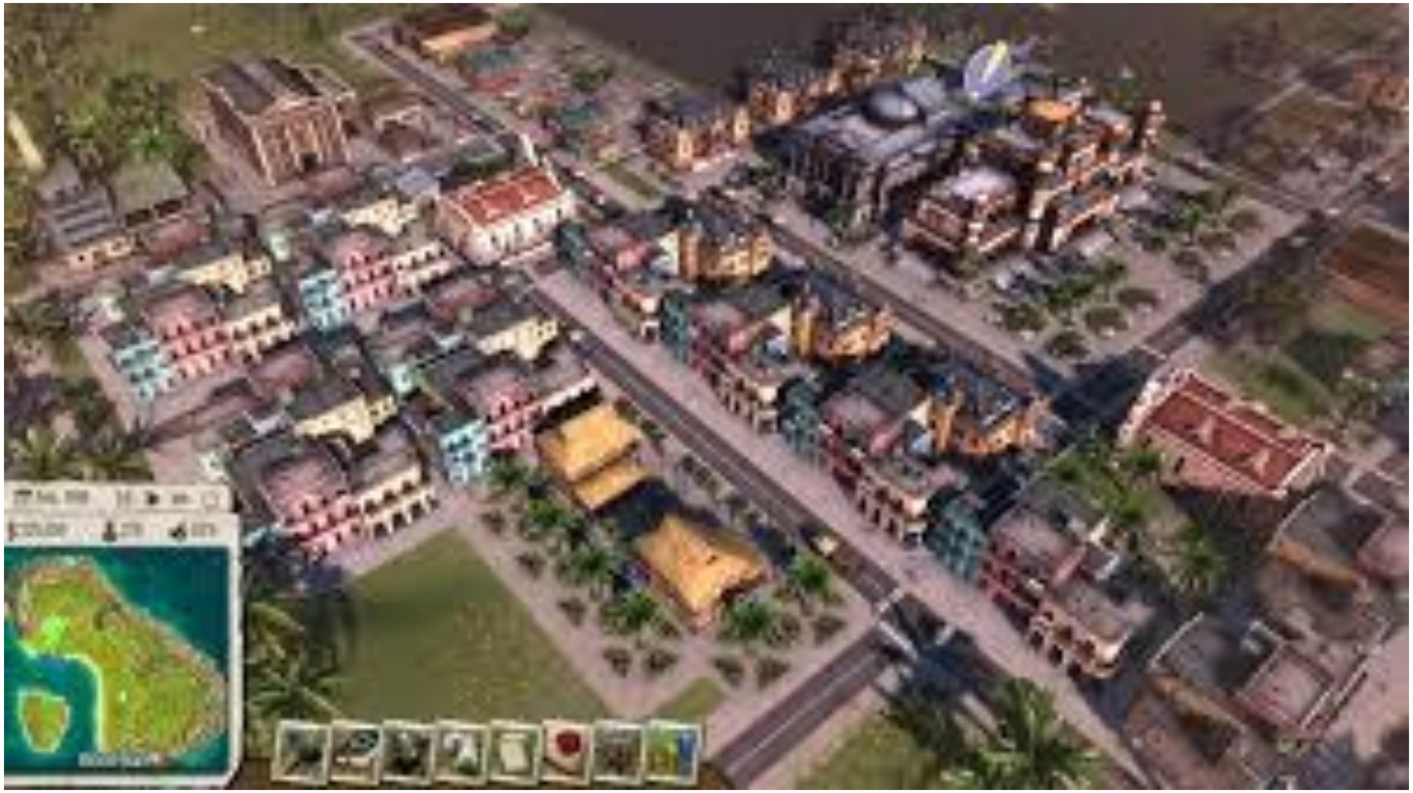


FPS vs TPS

- What Are the Differences Between First-Person Shooter and Third-Person Shooter Games? http://www.ebay.com/gds/What-Are-the-Differences-Between-First-Person-Shooter-and-Third-Person-Shooter-Games-/10000000177589743/g.html

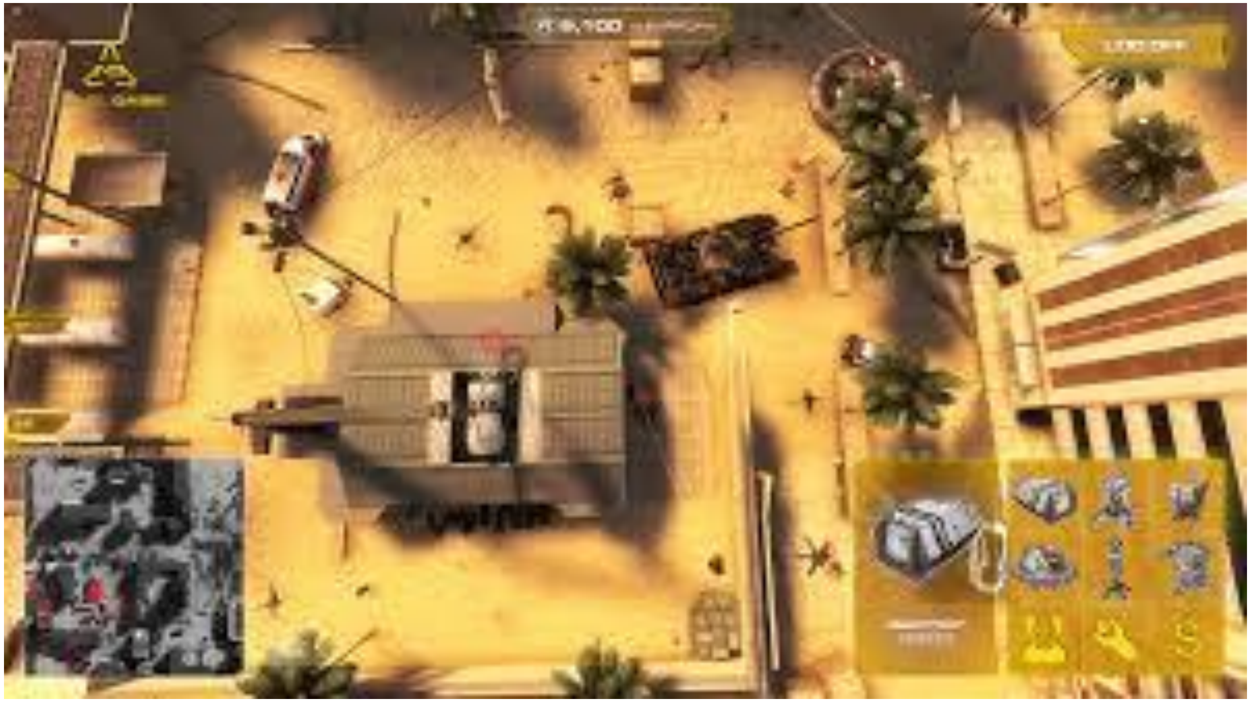
Other camera perspectives 1/4

- *Aerial perspective* – suitable for games with party-based or multi-present interaction models



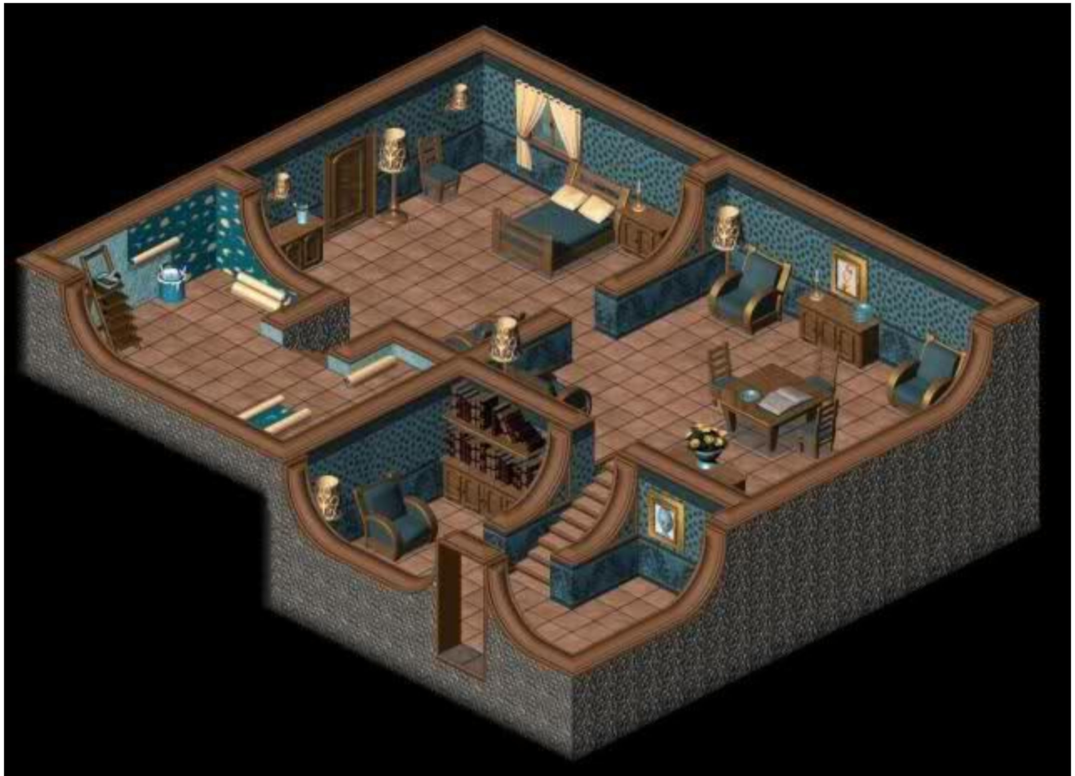
Other camera perspectives 2/4

- *Top-down perspective* - shows the game world from directly overhead with the camera pointing straight down (resembles a map)



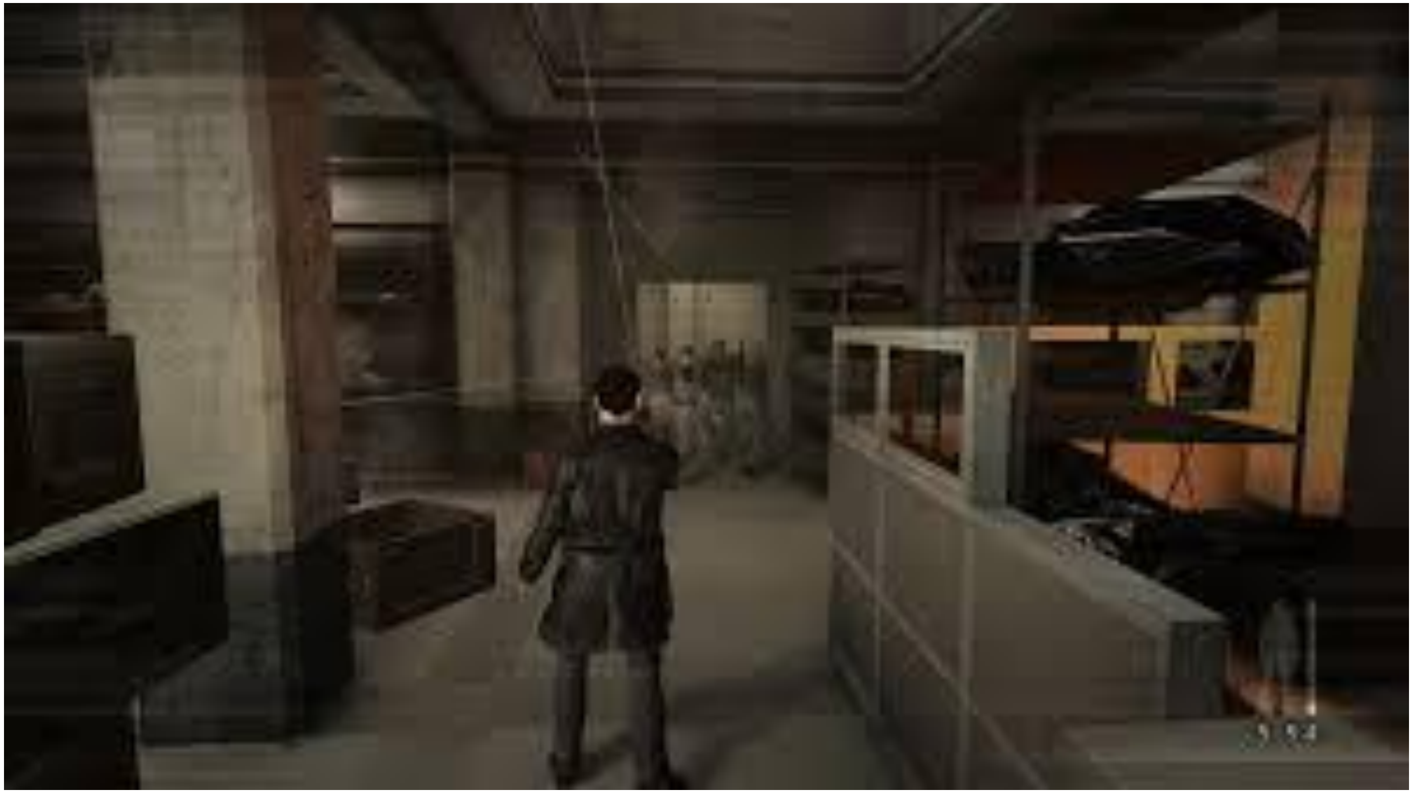
Other camera perspectives 3/4

- *Isometric perspective* - brings the player closer to the action than the top-down perspective and allows him to see the sides of buildings as well as the roofs, so the player feels more involved with the world, with the camera tilted down about 30 (45) degrees from the horizontal



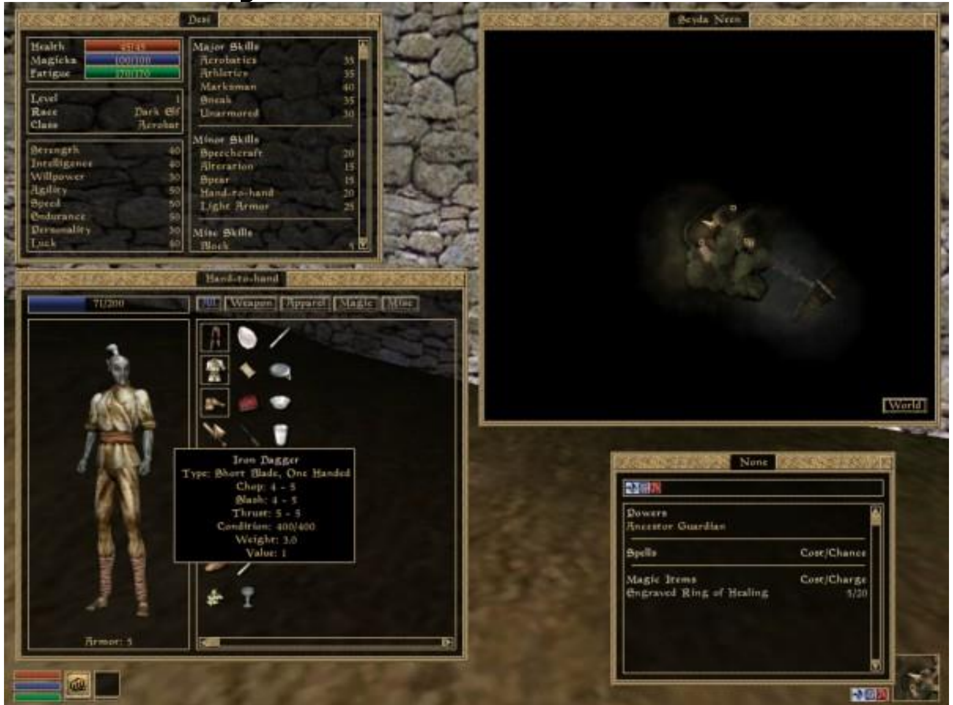
Other camera perspectives 4/4

- *3D perspective (free-roaming camera)* – objects farther away seem smaller



Visual elements - Main view 1/2

- Main view may appear:
  - in a window within the screen with other UI elements around it – called *windowed view*, used in RPG and construction games because they require so many on-screen controls



Dasi

Health 45/45
Magicka 100/100
Fatigue 170/170

Level 1
Race Dark Elf
Class Acrobat

Strength 40
Intelligence 40
Willpower 30
Agility 50
Speed 50
Endurance 50
Personality 30
Luck 40

Major Skills
Acrobatics 35
Athletics 35
Marksman 40
Sneak 35
Unarmored 30

Minor Skills
Speechcraft 20
Alteration 15
Spear 15
Hand-to-hand 20
Light Armor 25

Misc Skills
Block 5

Hand-to-hand

71/200

Iron Dagger
Type: Short Blade, One Handed
Chop: 4 - 5
Slash: 4 - 5
Thrust: 5 - 5
Condition: 400/400
Weight: 3.0
Value: 1

Armor: 5

Seyda Neen

None

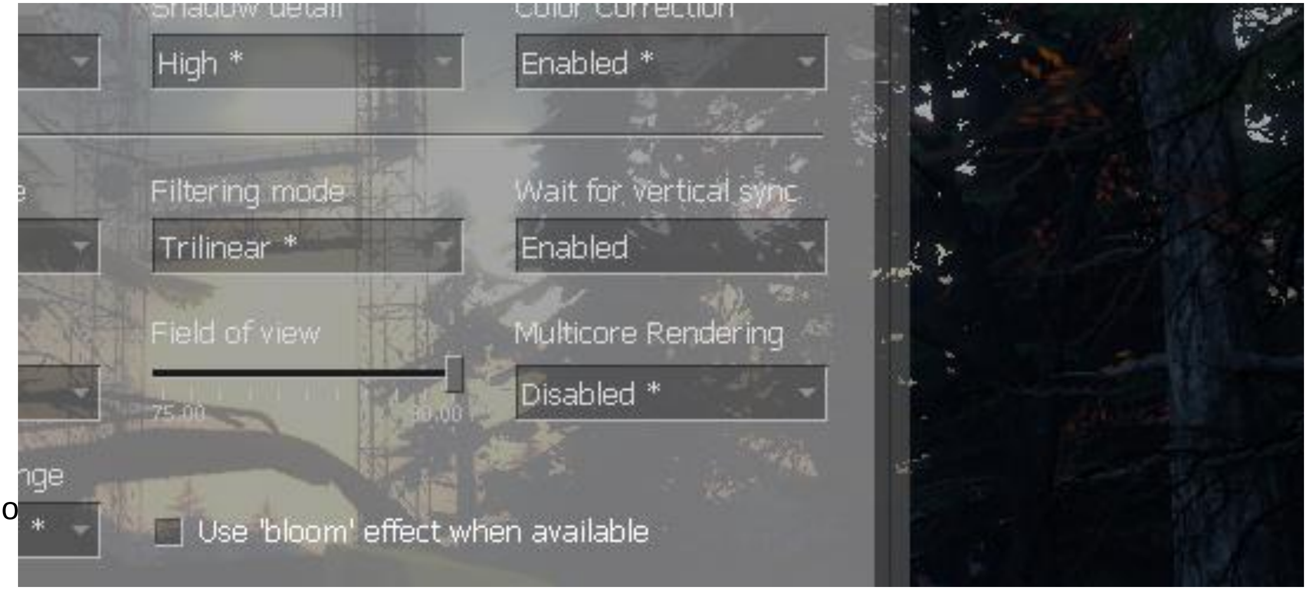
Powers
Ancestor Guardian

Spells Cost/Chance

Magic Items Cost/Charge
Engraved Ring of Healing 5/20

Visual elements - Main view 2/2

- Main view:
  - may occupy the whole screen and the other UI elements will appear superimposed on top of it in *overlays*
    - Non-transparent overlays
    - Semi-transparent overlays



Shadow detail: High *

Color Correction: Enabled *

Filtering mode: Trilinear *

Wait for vertical sync: Enabled

Field of view: 75.00 – 90.00

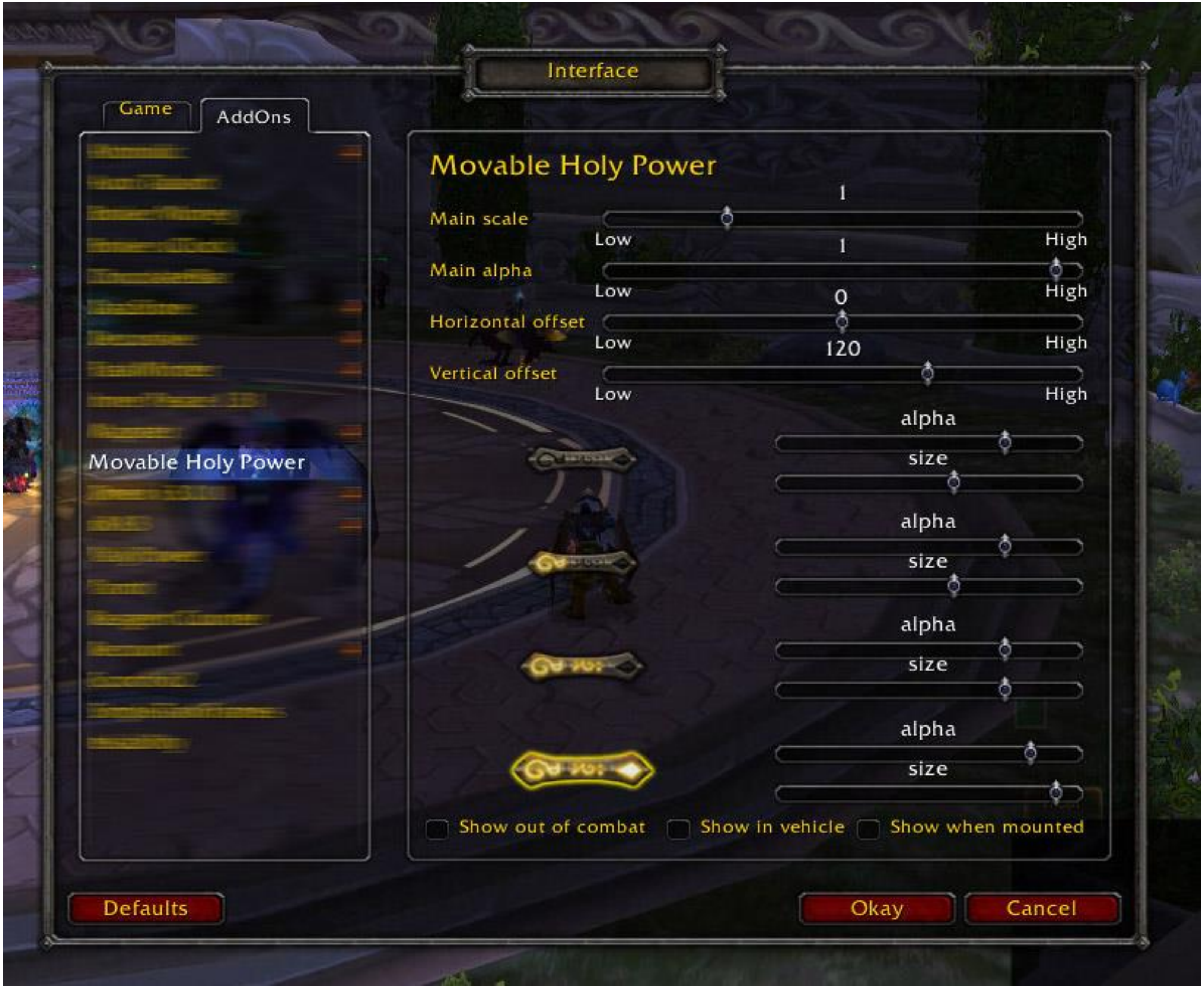
Multicore Rendering: Disabled *

□ Use 'bloom' effect when available

Visual elements - Feedbacks 1/3

Feedback elements communicate details about the game’s inner states (its core mechanics) to the player

- *Indicators* - inform the player about the status of a resource, graphically and at a glance.
  - □ Digits
  - □ Needle gauges (a car’s speedometer)
  - □ Power bar (an analog thermometer)
  - □ Small multiples (the bars indicating signal strength on a mobile phone)
  - □ Colored lights, icons, and text indicators (symbolic)



Interface

Game

AddOns

[redacted]

Movable Holy Power

[redacted]

Movable Holy Power

Main scale: 1 (Low – High)

Main alpha: 1 (Low – High)

Horizontal offset: 0 (Low – High)

Vertical offset: 120 (Low – High)

alpha

size

alpha

size

alpha

size

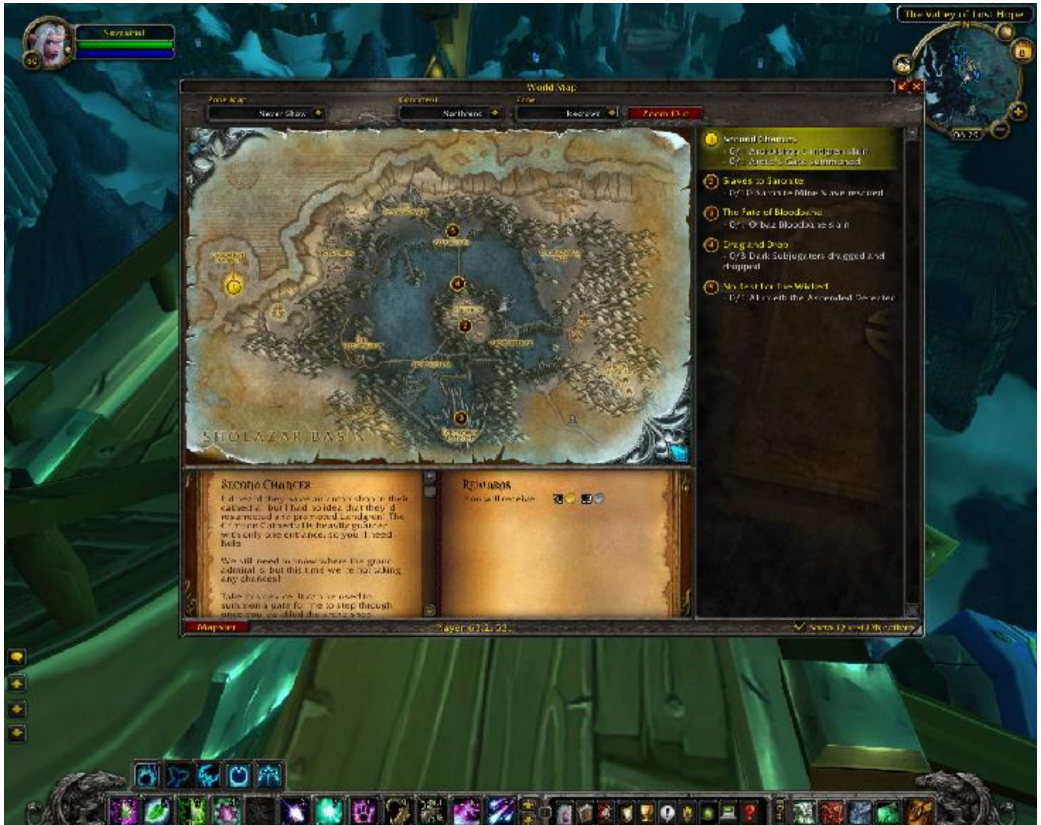
alpha

size

☐ Show out of combat ☐ Show in vehicle ☐ Show when mounted

Visual elements - Feedbacks 2/3

- *Mini-maps* - miniature version of the game world, or a portion of it, from a top-down perspective
  - □ World-oriented
  - □ Character-oriented



Visual elements - Feedbacks 3/3

- *Screen buttons*
- *Menus*
- *Texts (localization issues)* - Keep Text Separate from Other Content
- *Typefaces and formatting*
- *Videos*
- *Music*
- *Sound effects*
- Vibration
  - Intensity
  - Duration
- *Dialogs and narrative*

Input devices 1/2

- Keyboards
- Pointing Devices
  - Based on rolling a ball – mouse, trackball
  - Based on touching a surface – touchpad, graphics tablet (digitizer, digitizing tablet, graphics pad), **touchscreen**, light pen, stylus
  - Based on moving stick
    - **Joystick**
    - **Gamepad / video game controller**

Input devices 2/2

- Graphic and Video Input Devices
  - Scanner (2D, 3D)
  - Webcam

- Audio input devices
- Motion sensing input devices

UI and UX

- UI, or *User Interface*, refers to the methods (keyboard control, mouse control) and interfaces (inventory screen, map screen) through which a user interacts with your game.
- UX, or *User Experience*, refers to how intuitive and enjoyable those interactions are.

Source: https://gamedevelopment.tutsplus.com/tutorials/game-ui-by-example-a-crash-course-in-the-good-and-the-bad--gamedev-3943

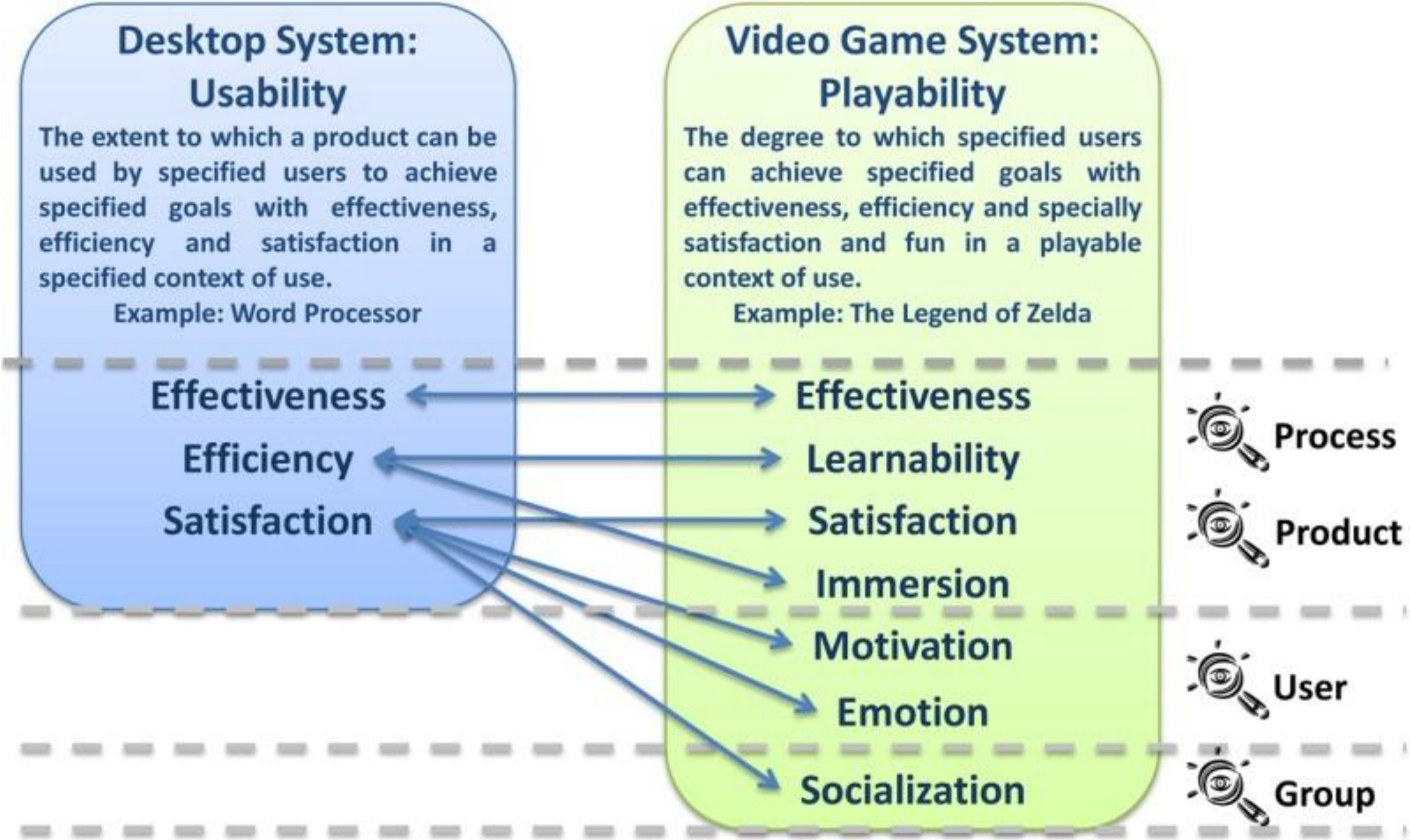
From usability to playability

- For desktop software, *usability* is a measure of product use whereby users achieve concrete objectives in varying degrees of
  - *effectiveness*
  - *efficiency*
  - *satisfaction*
- For video games, *playability* is: *'a set of properties that describe the Player Experience using a specific game system whose main objective is to provide enjoyment and entertainment, by being credible and satisfying, when the player plays alone or in company'*

Slides 29-34 are from: Jose Luis González Sánchez, Natalia Padilla Zea, and Francisco L. Gutiérrez. From Usability to Playability: Introduction to Player-Centred Video Game Development Process, HCII 2009, LNCS 5619, pp. 65–74, 2009

7 attributes of playability

- *Satisfaction* - gratification or pleasure derived from playing a video game
- *Learnability* - the player’s capacity to understand and master the game’s system and mechanics
- *Effectiveness*
- *Immersion* - the capacity of the video game contents to be believable, such that the player becomes directly involved in the virtual game world
- *Motivation*
- *Emotion*
- *Socialization*



Desktop System: Usability

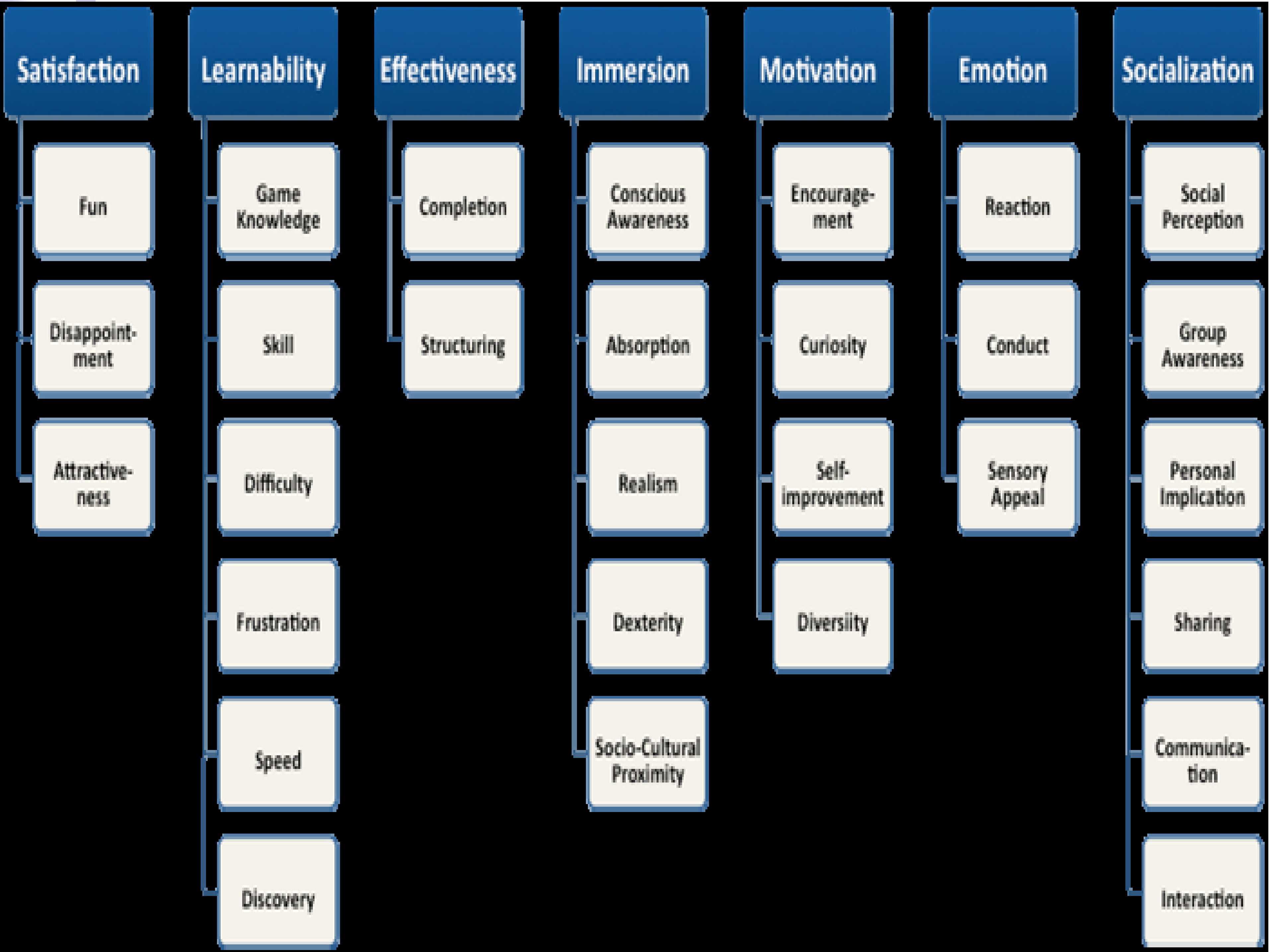
The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

Example: Word Processor

Video Game System: Playability

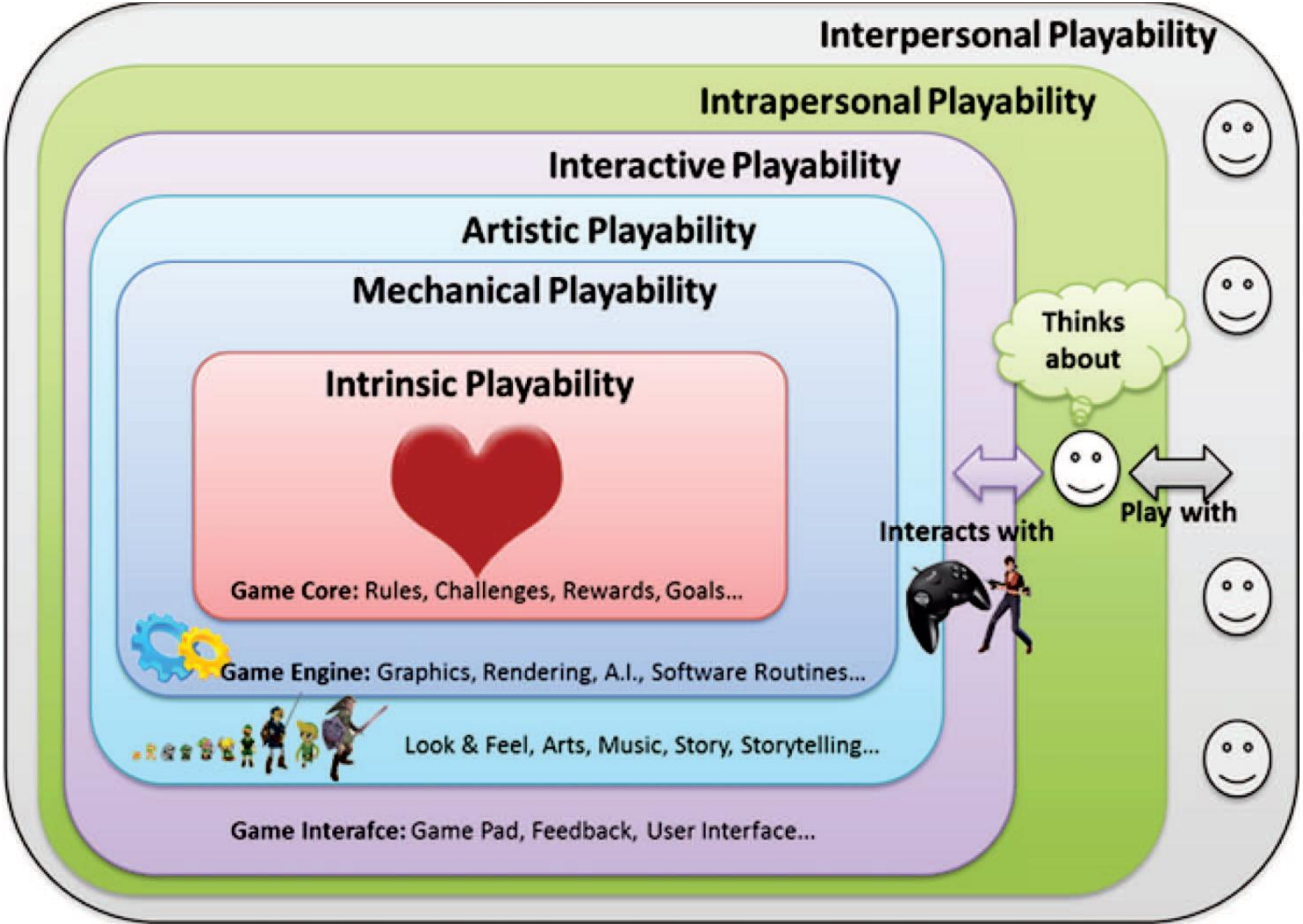
The degree to which specified users can achieve specified goals with effectiveness, efficiency and specially satisfaction and fun in a playable context of use.

Example: The Legend of Zelda



- Satisfaction
  - Fun
  - Disappointment
  - Attractiveness
- Learnability
  - Game Knowledge
  - Skill
  - Difficulty
  - Frustration
  - Speed
  - Discovery
- Effectiveness
  - Completion
  - Structuring
- Immersion
  - Conscious Awareness
  - Absorption
  - Realism
  - Dexterity
  - Socio-Cultural Proximity
- Motivation
  - Encouragement
  - Curiosity
  - Self-improvement
  - Diversiity
- Emotion
  - Reaction
  - Conduct
  - Sensory Appeal
- Socialization
  - Social Perception
  - Group Awareness
  - Personal Implication
  - Sharing
  - Communication
  - Interaction

Facets of Playability and video game elements


Interpersonal Playability
Intrapersonal Playability
Interactive Playability
Artistic Playability
Mechanical Playability
Intrinsic Playability
Game Core: Rules, Challenges, Rewards, Goals...
Game Engine: Graphics, Rendering, A.I., Software Routines...
Look & Feel, Arts, Music, Story, Storytelling...
Game Interafce: Game Pad, Feedback, User Interface...
Thinks about
Interacts with
Play with

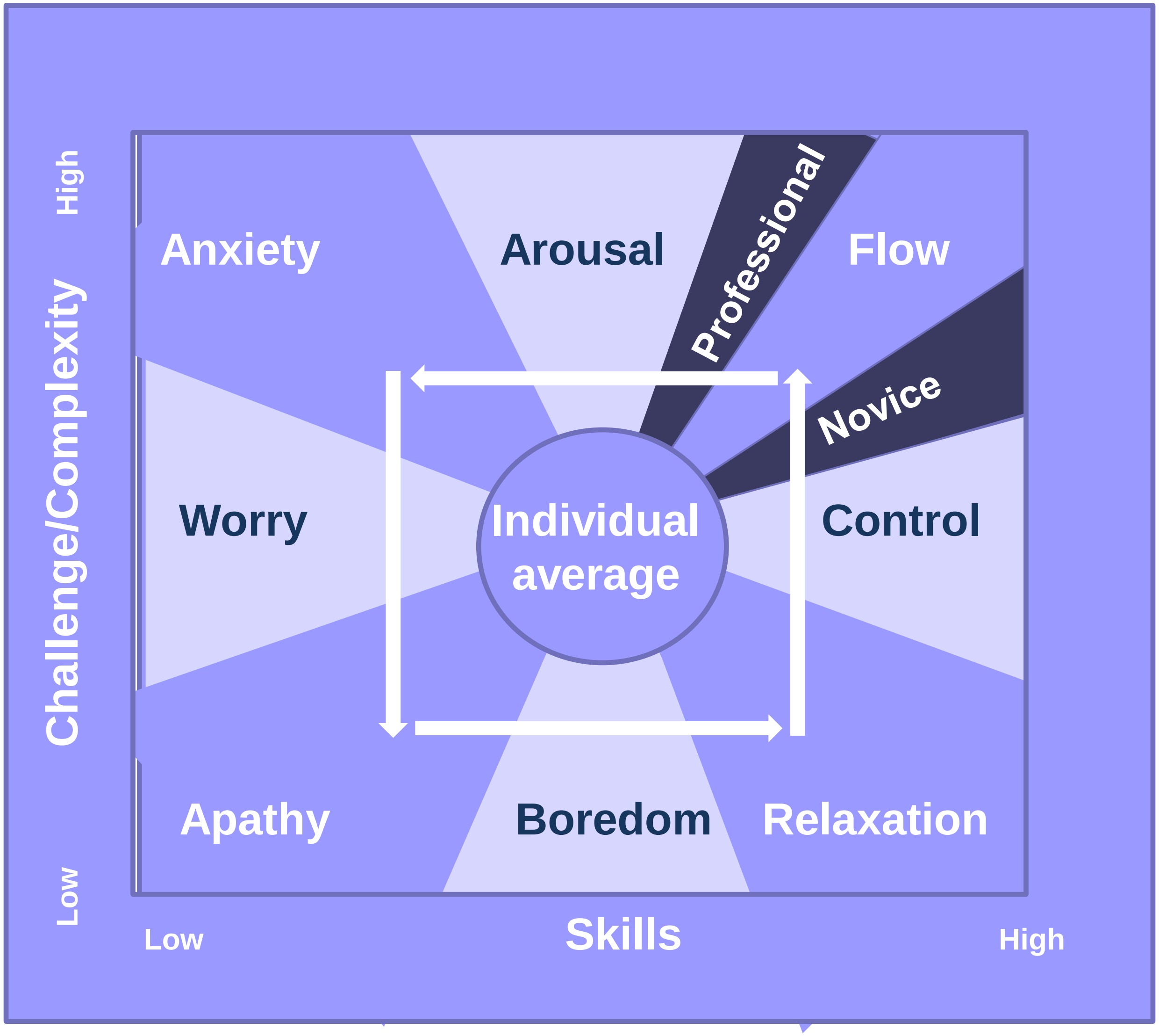
User Experience (UX)

UX is confused with usability. But UX is described by 7 factors:

- Useful
- Usable
- Findable
- Credible
- Desirable
- Accessible
- Valuable

Source: https://www.interaction-design.org/literature/article/the-7-factors-that-influence-user-experience

Flow mental state as function of challenge and player's ability/skills (Csikszentmihalyi, 1997)



UX and Playing styles

Playing style vs learning style

- **Playing style** - persistent traits of the player, whereas defined play styles as treating “motivations as a more temporary state, with an implication that players may adopt different play styles in different games or at different times”.

  (Magerko et al., 2008)

- **Learning style** – “characteristic cognitive, effective, and psychosocial behaviors that serve as relatively stable indicators of how learners perceive, interact with, and respond to the learning environment”.

  (Curry, 1981)

Bartle types

Richard Bartle in 1996 was the first researcher who developed a family of playing styles (types) for multiplayer gameplay:

- Killers: force other players to accept their goals and desires by an active and predominant play experience
- Achievers: overcome gameplay challenges in order to collect more game items (such as tokens, munitions, etc.)
- Explorers: adore to discover and explore the game world
- Socializers: predominantly set closer relationships with other players

Keirsey playing style

Keirsey(1998) developed another playing style categorization system by using the names of basic temperaments:

- artisan (concrete and adaptable)
- guardian (concrete and organized)
- idealist (abstract and compassionate), and
- rational (abstract and objective).

Keirsey divided the four temperaments into two categories (roles) – proactive and reactive, each of them with two types (role variants) – attentive and expressive. Thus, the categorization system of Keirsey resulted in 16 styles.

Bateman and Boon

Bateman and Boon (2005) developed the demographic game design (DGD) player typology in four categories:

- The conqueror - stays for goal-oriented players ready to pay anything for winning the game; very competitive and dominant in multi-player gameplays;
- The manager - represents process-oriented players enjoying enhancing and practicing their mastery.
- The wanderer - typical for players looking for fun and new experiences; prefers enjoyment than challenges in gameplay.
- The participant - looks mainly for social contacts and interactions with the other players.

Relationships between Honey and Mumford (1982) learning styles and playing styles of ADOPTA (Aleksieva et al., 2011) and Bartle (1996)

Honey and Mumford	Learns/plays by:	ADOPTA	Bartle
Activist	Hand–eye coordination, planning and strategizing, problem-solving, teamwork and the ability to think quickly	**Competitor**	**Killer**
Theorist	Logically entering problems step-by-step, with spatial awareness and verbal & numeracy skills	**Logician**	**Achiever**
Pragmatist	Planning, decision-making, testing hypotheses, strategic thinking, management skills	**Strategist**	**Explorer**
Reflector	Observing and watching reflectively	**Dreamer**	**Socializer**

Homework

- José Luis González Sánchez , Francisco Luis Gutiérrez Vela, Francisco Montero Simarro & Natalia Padilla-Zea (2012) Playability: analysing user experience in video games, Behaviour & Information Technology, 31:10, 1033-1054, DOI: 10.1080/0144929X.2012.710648

Video
games
Keywords
presentations
colloquium
Studies
presenters
University
game
videogames
play
Game
abstract
April
work
less
sessions
study
medium
Washington
group
provide
Participants
imagine
invites
program
make
performances
demonstrations
Humanities
Colloquium
programs
lightning
uses
design
HASTAC
interest
Friday
Long
organized
A/V
Things
broader
email
notified
Advanced
Project
equipment
objects
walk-throughs
leisure
videogame
event
keywords
unable
names
development
Critical
Send
developed
Idea
takes
Center
project
understanding
discussion
suggested
demonstration
Unfortunately
Communication
many
designers
Short
performance
enthusiasts
space
scholars
Collaboratory
argues
research/design
need
critical
granted
titles
conference
gold
working
suggest
hopes
introductions
conversation
affiliations
extended
collaboration
practices
question
engage
provocations
fun
expenses
present
individuals
spectrum
activism/politics
Instead
GIG
brief
series
accepted
institutions
RESEARCH/DESIGN
artists
graduate
long
short
Saturday
groups
discuss
continues
introduction
insufficient
limited
fandom/community
year-long
valid
art
code
across
digital
talk
along
include
CALL
themed
PM
6-8
style
roundtable
time
interventions
play/work
academia
answer
everything
collaborate
take
fully
1-3
Simpson
transportation

Въпроси