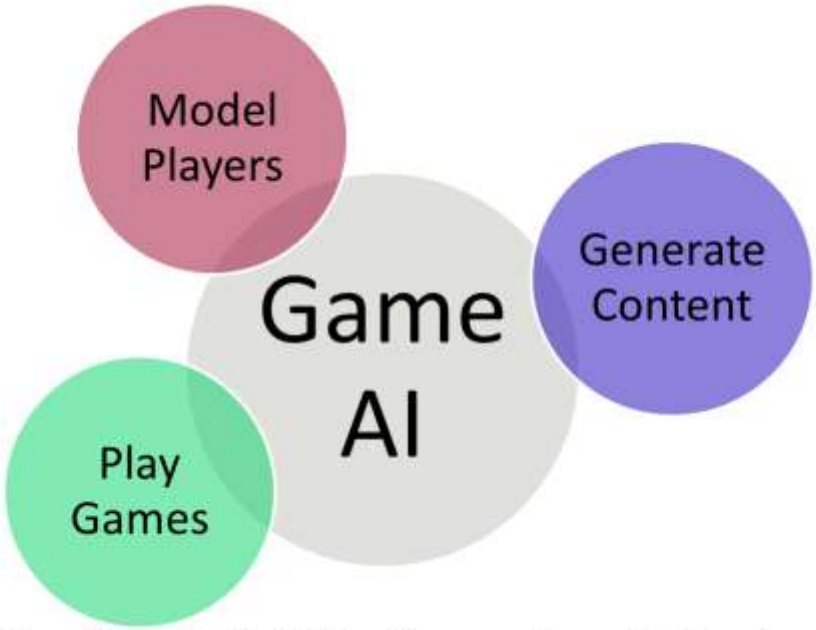
# Artificial Intelligence and Games
## Modeling Players

Georgios N. Yannakakis
**@yannakakis**

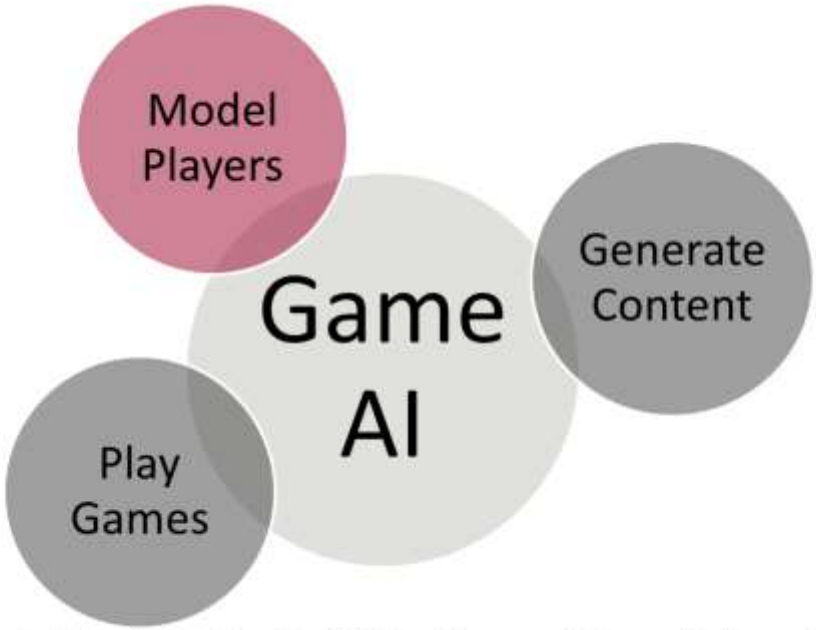
Julian Togelius
**@togelius**


These are the slides accompanying the book Artificial Intelligence and Games through the gameaibook.org website




G. N. Yannakakis and J. Togelius, "**Artificial Intelligence and Games**," Springer Nature, 2018.

As a reminder: "Model Players" is identified as one of the three major roles of AI in games in this book.




G. N. Yannakakis and J. Togelius, "**Artificial Intelligence and Games**," Springer, 2018.

The focus of this slide deck is on the role of AI for modelling players


Your readings from **gameaibook.org**

Chapter: 5

# Overview

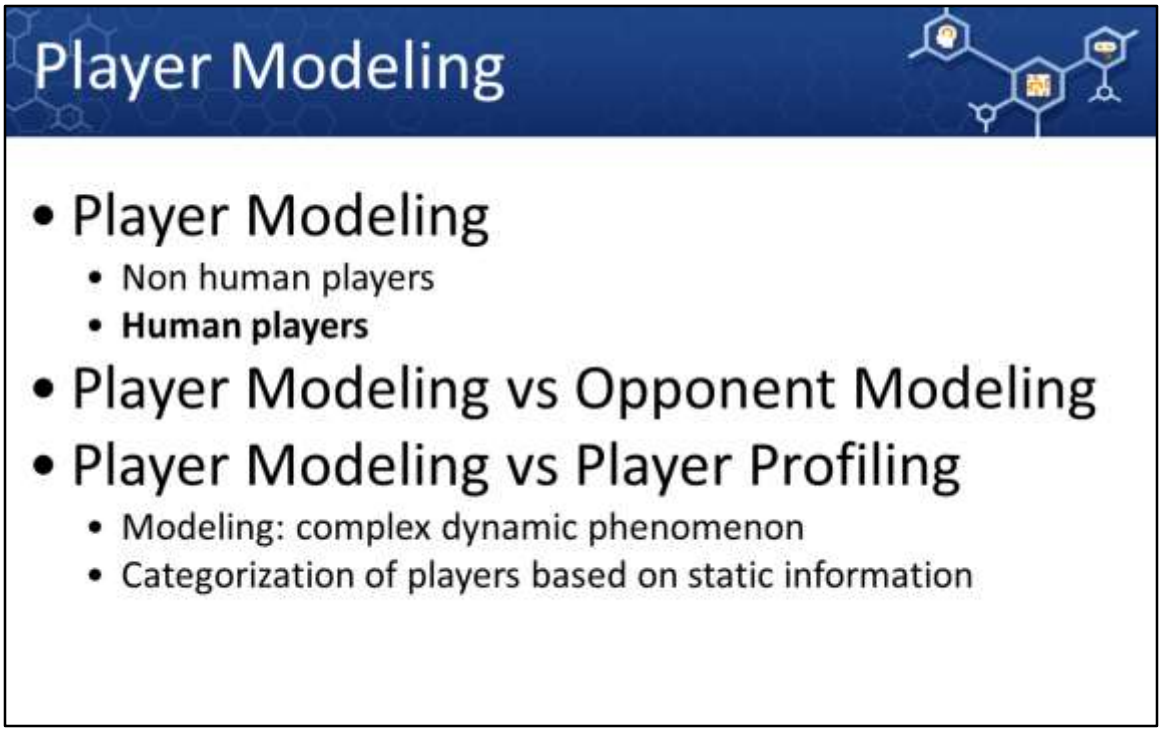
- Player Modeling
  - Why?
  - How?
- Main tasks for AI/Machine Learning
- Holistic view
  - From Affective Computing to Game Studies to Game Analytics
- Examples
  - Player Behavior
  - Player Experience

The overview of the slide-deck

# What is Player Modeling?

[see Section 5.1 for more details]

Let us first attempt to define what player modeling is within the focus of this book




We could detect behavioral, emotional or cognitive aspects of both human players and non-human players, or non-player characters. However, in this book we focus on aspects that can be detected from, modeled from, and expressed in games wi**th human players**.

Sometimes the terms player modeling and *opponent modeling* are used interchangeably when a human player is modeled. However, opponent modelling is a more narrow concept referring to predicting behavior of an adversarial player when playing to win in an imperfect information game like Poker or StarCraft (Blizzard Entertainment, 1988)

We also make a distinction between player modeling and *player profiling*. The former refers to modeling complex dynamic phenomena during gameplay interaction, whereas the latter refers to the categorization of players based on static information that does not alter during gameplay. Information of static nature includes personality, cultural background, gender and age. We put an emphasis on the former, but will not ignore the latter, as the availability of a good player profile may contribute to the construction of reliable player model.

# Player Modeling

The study of *computational* means for the modeling of a player's **experience** or **behavior** which is based on *theoretical frameworks* about player experience and/or *data* derived from the interaction of the player with a game

Our definition in this book

# Why Model Players?

[see Section 5.2 for more details]

Let us first attempt to define what player modeling is within the focus of this book

## Goals and Aims

**Goal**

"*…understand how the interaction with a game is experienced by players.*"

Why use AI for Player modelling?

- Understanding player experience
- Understanding player behavior

[see Section 5.2 for more details]

The primary goal of player modeling is to understand how the interaction with a game is experienced by individual players. Thus, while games can be utilized as an arena for eliciting, evaluating, expressing and even synthesizing experience, we argue that the main aim of the study of players in games is the understanding of players' cognitive, affective and behavioral patterns. Indeed, by the very nature of games, one cannot dissociate games from player experience.

# Why Model Players?

- Why not?
- Machines understand numbers
- Player Experience is the holy grail for design and development
- But most importantly because…

[see Section 5.2 for more details]

# Why Model Players?

- The perfect game is tailored to you!
- We are different (and many more than before)
- If you learn to play…. it is only fair that the game **learns you**

[see Section 5.2 for more details]

Image: Player Modeling example in the stealth horror game *Hello Neighbor.* In that game the AI learns from each of your moves. The neighbor keeps track of the player all the time and learns from your mistakes. Your neighbour will, in turn, teach AI how you can be defeated.

## Core Player Modeling Tasks

**Supervised/Reinforcement Learning**

Imitation

Prediction

**Unsupervised Learning**

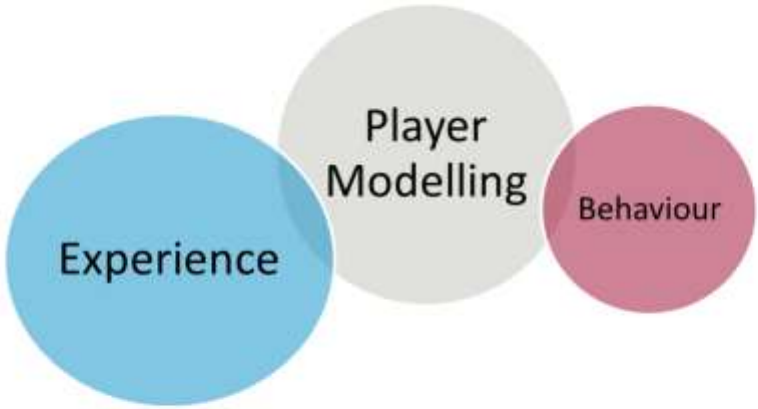
Clustering

Association mining

[see Section 5.2 for more details]

# Player Modeling Examples


- Arousal-driven appearance of non-player characters (NPCs) *Left 4 Dead* 2
- The fearful combat skills of the opponent NPCs in *F.E.A.R.*,
- The avatars' emotion expression in the *Sims* series and *Black and White*
- The emotional play-through for characters in *Psychonauts*
- The emotional responses of game characters in *Prom Week* and *Façade*
- The personality-based adaptation in *Silent Hill: Shattered Memories*
- The affect-based cinematographic representation of multiple cameras in *Heavy Rain*
- The aesthetically pleasing locations of *World of Warcraft*
- Affect-centred game narratives such as the one of *Final Fantasy VII*
- …




G. N. Yannakakis and J. Togelius, "**Artificial Intelligence and Games**," Springer Nature, 2018.

[see Section 5.2 for more details]

While player behavior and player experience are interwoven notions there is a subtle difference between them.

## Player Experience vs Player Behavior

**Experience**: how you **feel during play**

- A set (a synthesis) of affective, cognitive and behavioral states
- Or else *user states*
- Emotions: Appraisal theory, …
- Cognition/Behavior: several models (e.g. BDI,…)

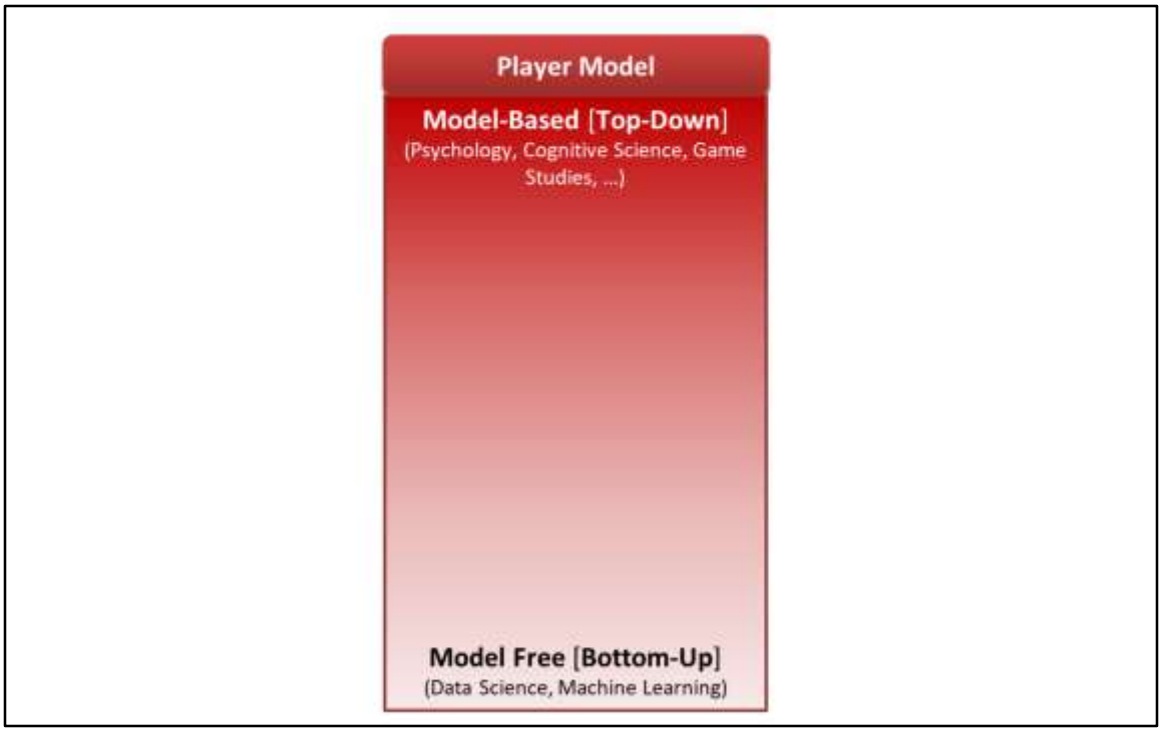
**Behavior**: what you **do during play**


[see Section 5.2 for more details]

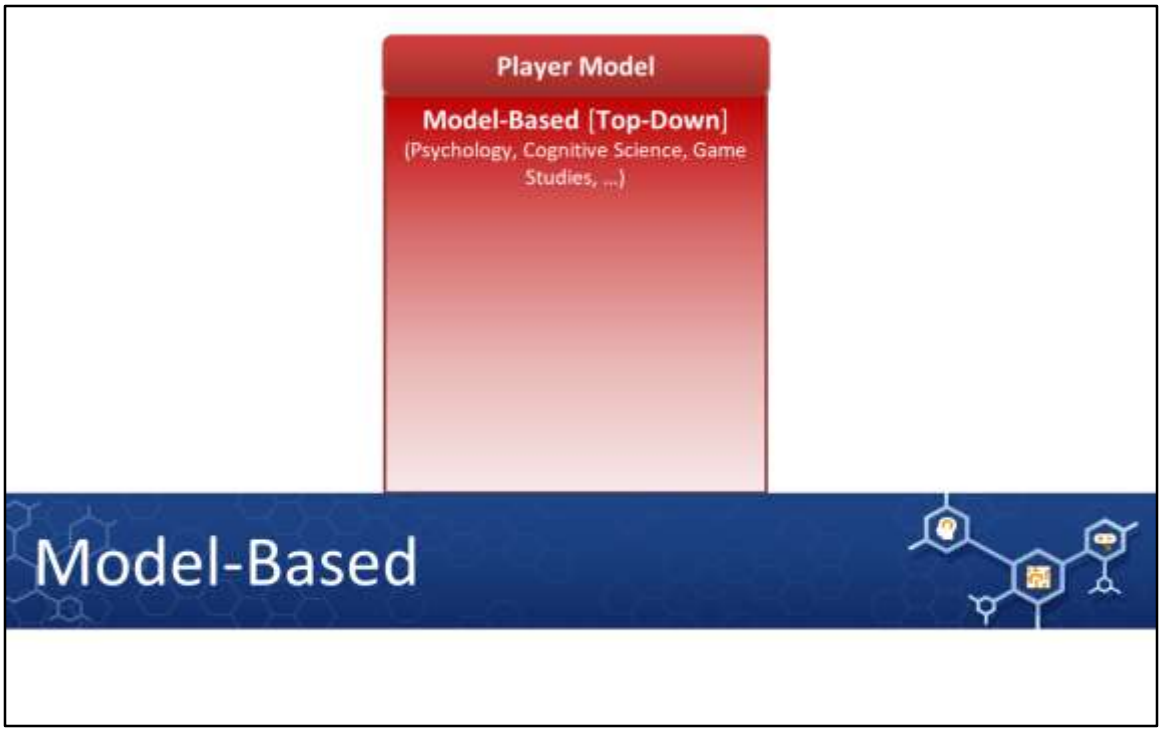
Player behavior points to what a player **does** in a game whereas player experience refers to **how** a player **feels** during play. The feeling of one's gameplay experience is clearly associated with what one does in the game; player experience, however, is primarily concerned with affective and cognitive aspects of play as opposed to mere reactions of gameplay which refer to player behavior.

# A High-Level Taxonomy

[see Section 5.3 for more details]




A high-level classification of the available approaches for player modeling can be made between model-based (or top-down) and model-free (or bottom-up) approaches. The above definitions are inspired by the analogous classification in RL by which a world model is available (i.e., model-based) or not (i.e., model-free). Given the two ends of this continuum hybrid approaches between them can naturally exist. The gradient red color of the player model box in this Figure illustrates the continuum between top-down and bottom-up approaches.




In a model-based or top-down approach a player model is built on a theoretical framework. As such, researchers follow the modus operandi of the humanities and social sciences, which hypothesize models to explain phenomena. Such hypotheses are usually followed by an empirical phase in which it is experimentally determined to what extent the hypothesized models fit observations; however, such a practice is not the norm within player experience research. While user experience has been studied extensively across several disciplines, in this book we identify three main disciplines we can borrow theoretical frameworks from and build models of player experience:

- **Psychology** and **affective sciences**
- **Neuroscience**
- **Game studies** and **game research**


[see Section 5.3.1.1 for more details]

Within top-down player modeling we will first cover a number of key notions and definitions within psychology and affective sciences

## Psychology and Affective Sciences

"**Affective computing** is the study and development of systems and devices that can recognize, interpret, process, and simulate human **affect**" (Wiki)

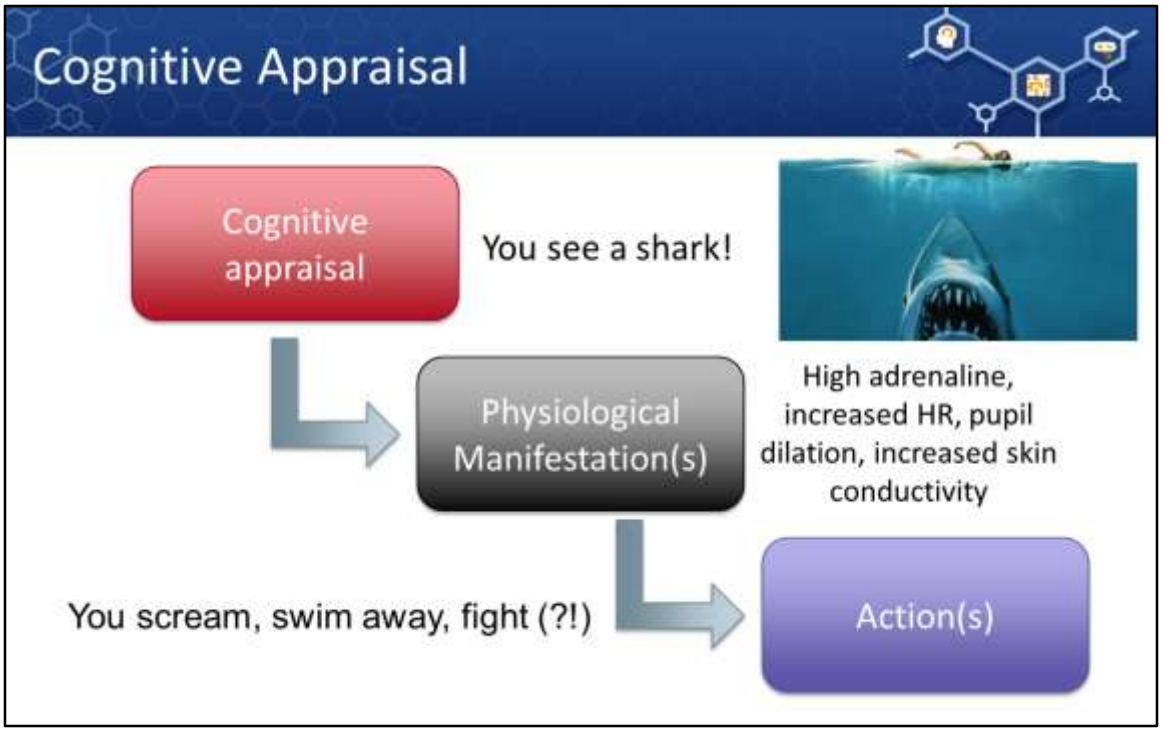
Term coined by R. Picard (1999)


[see Section 5.3.1.1 for more details]

# Emotion

**Emotion** is a subjective, conscious experience characterized primarily by psychophysiological expressions, biological reactions, and mental states. (Wikipedia)

NB: Emotion (and affect) is a **construct** with **fuzzy** boundaries!




Emotion occurs in the following order:

1. Cognitive appraisal—The individual assesses the event cognitively, which cues the emotion.
2. Physiological changes—The cognitive reaction starts biological changes such as increased heart rate or pituitary adrenal response.
3. Action—The individual feels the emotion and chooses how to react.

For example:

- Jenny sees a shark.
- Jenny cognitively assesses the shark in her presence. Cognition allows her to understand it as a danger.
- Her brain activates adrenaline gland which pumps adrenaline through her blood stream resulting in increased heartbeat.
- Jenny screams and runs away.

# Emotion Representation – Eckman

Emotions

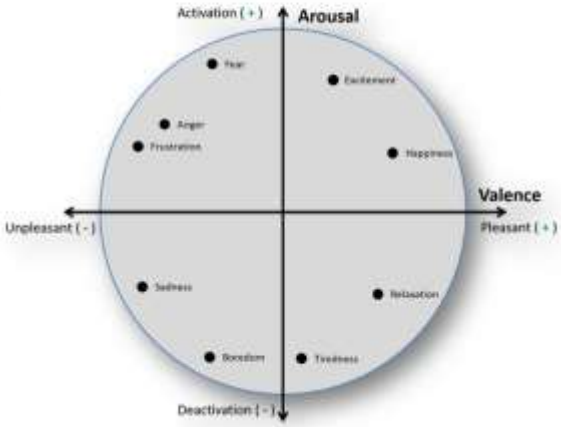
- Are **discrete** (states)
- Are measurable
- Can be identified through their physiological or bodily manifestations

The 6 basic emotions are culture-independent

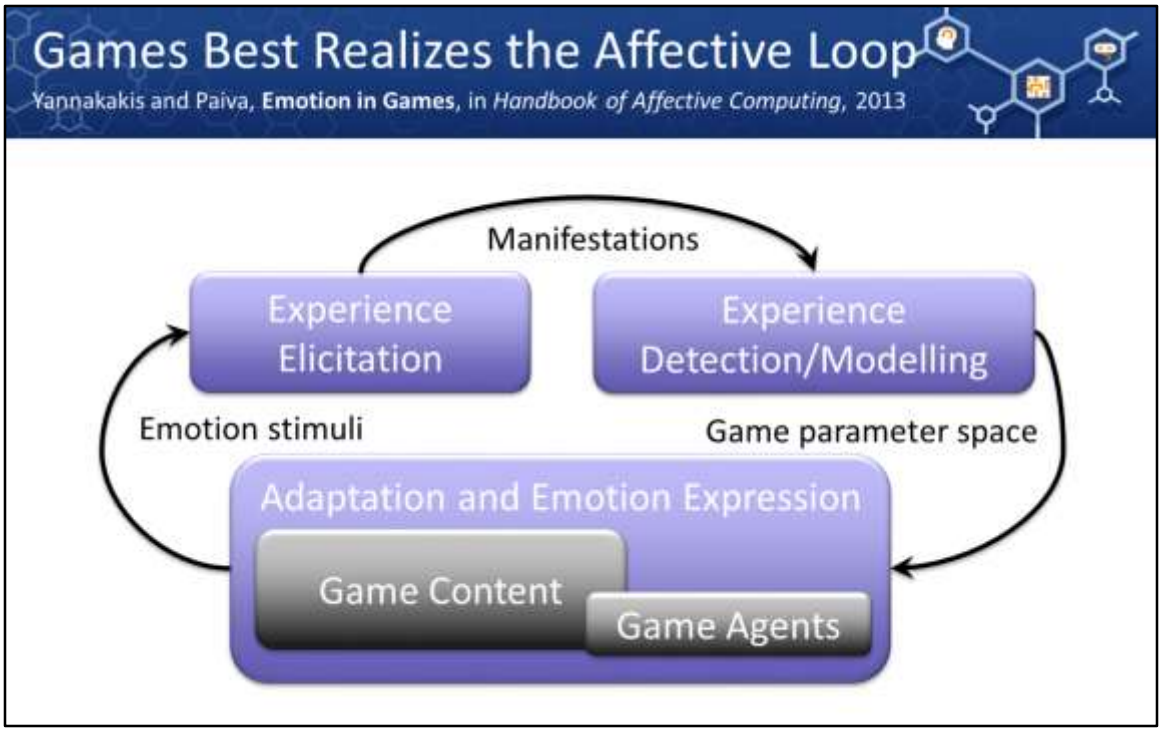
# Emotion Representation – Russell

Emotions

- Are **scalar values**
- Represented in a 2-d plane: **arousal** vs. **valence**
- Are measurable
- Can be identified through physiological or bodily manifestations




The Circuplex Model by Russell




This is a reminder slide about affective Loop and affective interaction (see slides of chapter 4 as well!)

This is an illustration of the *affective loop* as applied to games. The 3 core phases include: emotion elicitation, emotion detection and emotion expressions. Here are some reasons why games offer the best possible realization of the affective loop.

- Emotion elicitation: games offer brilliant contextual building blocks for eliciting emotion as stimuli are variant and come from different sources such as images, sounds, stories etc.
- Emotion Detection: users of games (players) are generally more that willing to provide more input of multimodal nature (via sensors) as long as that would lead to enhanced experience. In a sense, players are the best possible users for affective computing and multimodal interaction studies
- Emotion Adaptation:
    - Users voluntarily go through a spectrum of experiences during play: these vary from the very positive to the very negative ones
    - Affective experiences in games are **affected** by players! As a result a player is used to and largely **open** to affect-based adaptation!

EDPCG realises the game affective loop by offering content that is “appropriate” (based on experience heuristics) for players. EDPCG is a content-based formalization of the game affective loop.

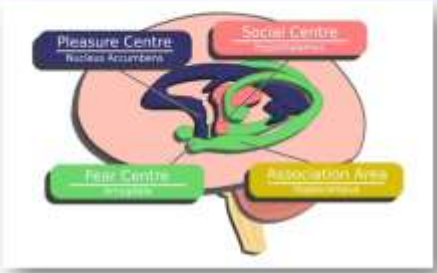

[see Section 5.3.1.2 for more details]

A number of studies have relied on the working hypothesis of an underlying mapping between the brain, its neural activity and player experience. However, this relationship is not well explored and the presumptive mapping is largely unknown. We will now cover briefly some key assumptions and evidence that might be relevant to top-down player modelling.

# Neurobiology of Play

Neuroscience – via fMRI

- Interest (or curiosity): **visual cortex** (hippocampus) and **endomorphin.**
- Explorer (Barter) / Easy Fun (Lazzaro) / Curiosity (Malone) → Wonder: **endomorphin (pleasure center)**
- Achiever (Bartle) → Achievement: **dopamine (pleasure center).**
- Hardcore gamers / Hard Fun (Lazzaro) → Fiero: **dopamine (pleasure center)**
- Serious Fun (Lazzaro) → Excitement: **adrenaline (fear center)**
- Socialiser (Bartle) / People Fun (Lazzaro) → Enjoyment: **oxytocin (social center) and dopamine (pleasure center)**




Bateman, Chris, and Lennart E. Nacke. "**The neurobiology of play**" *Proceedings of the International Academic Conference on the Future of Game Design and Technology*. ACM, 2010.

[see Section 5.3.1.2 for more details]

Please refer to the cited work for more details

## Game Studies and Games Research

[see Section 5.3.1.3 for more details]

Finally we will cover a number of key notions and definitions within game studies and game research relevant to top-down player modelling.

# "Fun"

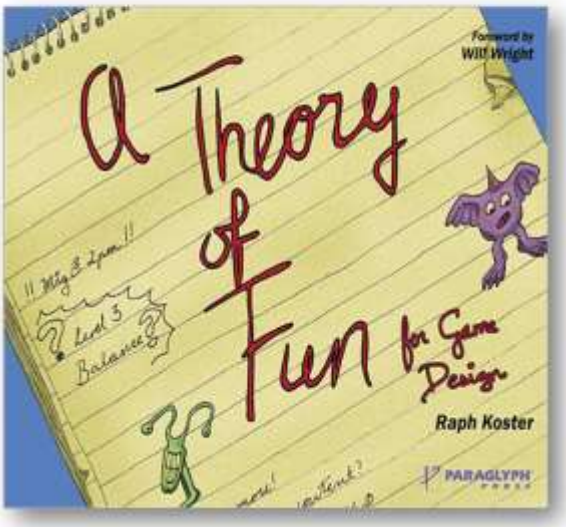
- "Fun" and friends
  - Learnability
  - The right difficulty level
  - Challenge – Curiosity – Fantasy
  - Hard Fun / Easy Fun / **X** Fun / ...
  - Adrenalin / Dopamine
  - Arousal / Positive Emotions
  - Flow
  - ...

# A Theory of Fun (Koster, 2005)

**Fun = Learning**

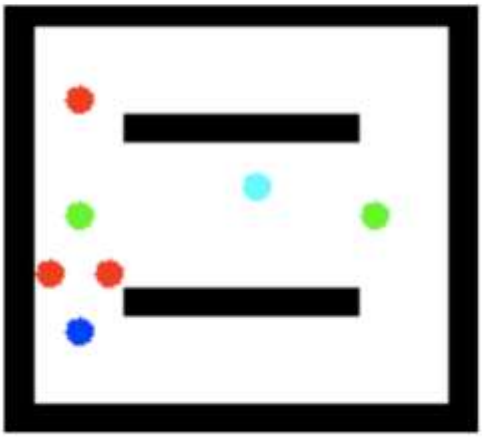
Complex training signals / patterns → Learning Difficulty → Frustration → No Fun!

Seen (and mastered) training signals / patterns → Boredom → No Fun!



# Evolving Game Rules Based on Koster

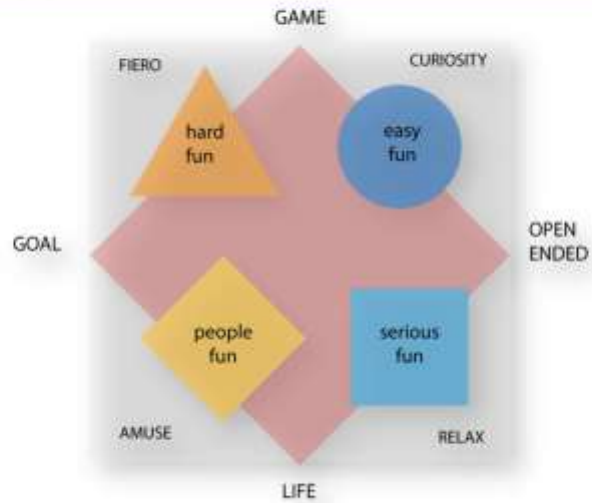
Fun of an AI agent
=
Learnability during the game (dynamic "fun" fitness function)



Togelius, J., & Schmidhuber, J. (2008, December). **An experiment in automatic game design**. In *Computational Intelligence and Games, 2008. CIG'08. IEEE Symposium On* (pp. 111-118). IEEE.

See cited paper for more details

# Lazzaro's 4 Fun-factor Model




(c) Nicole Lazzaro, www.xeodesign.com

HARD FUN
- Playing to see how good I really am
- Playing to beat the game
- Having multiple objectives
- Requiring strategy rather than luck

EASY FUN
- Exploring new worlds with intriguing people
- Excitement and adventure
- Wanting to figure it out
- Seeing what happens in the story, even if I have to use a walk through
- Feeling like me and my character are one

SERIOUS FUN (Altered states)
- Clearing my mind by clearing a level
- Feeling better about myself
- Avoiding boredom
- Being better at something that matters

PEOPLE FUN
- It's the people that are addictive not the game.
- I want an excuse to invite my friends over.
- I don't like playing games, but it's a fun way to spend time with my friends.
- I don't play, but it's fun to watch

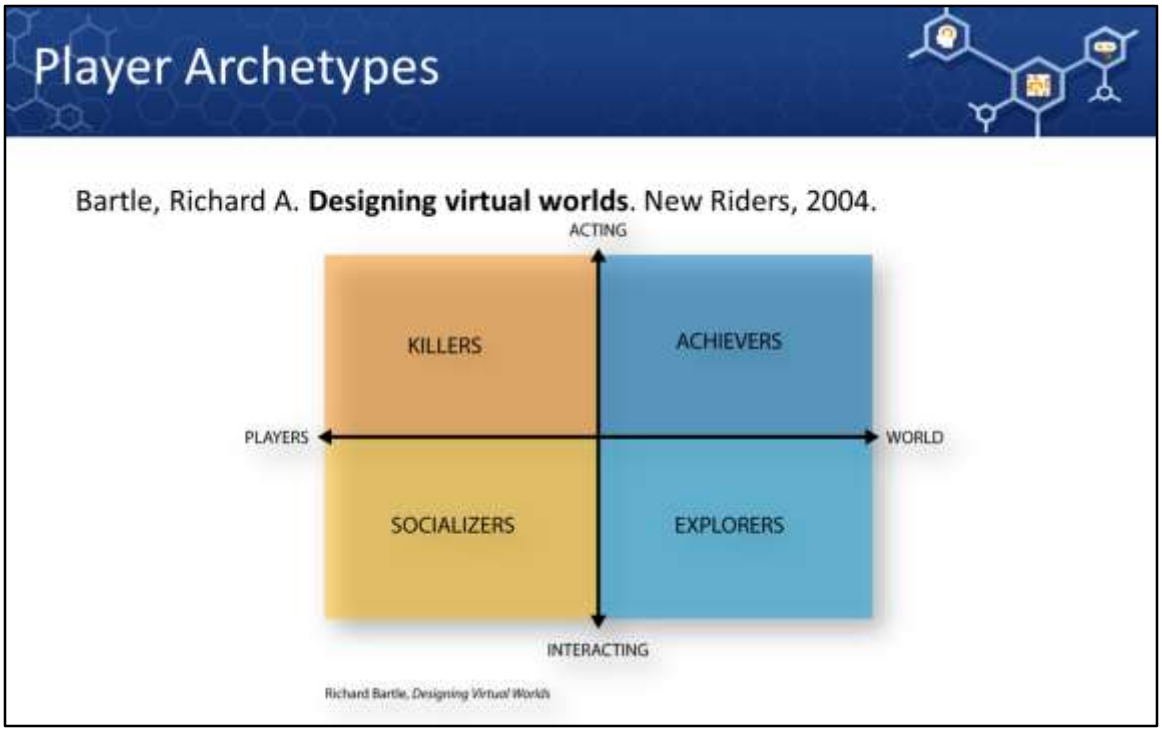
# Problems with "Fun"

Way too many

- Complex – composite of other things…
  - Engagement, interest, immersion, …? Who knows?
- Quasi/Fake Emotion (not real!)
- Nowhere in any model of emotion
- No device can measure "fun"


But

- People can still express it…
- Ask 4-yo children if they are immersed….




[Please refer to the cited book of Bartle for more details]

KILLERS:

- Focus: winning / competition
- Engaged by: ranks, leaderboards

ACHIEVERS

- Focus: attaining status and achieving preset goals quickly/completely
- Engaged by: achievements

SOCIALIZERS

- Focus: Socializing / develop network of friends/contacts
- Engaged by: friend lists/newsfeed/chat

EXPLORERS

- Focus: exploring/discover the unknown
- Engaged by: fuzzy achievements…

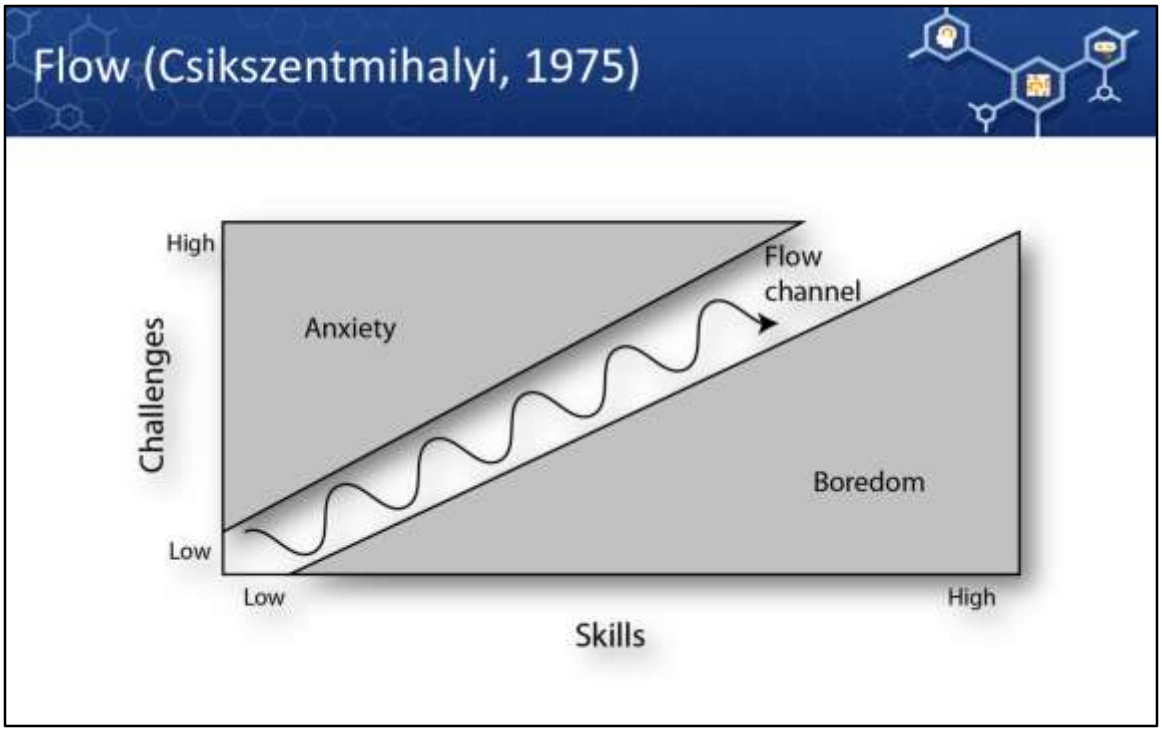
## Flow (Csikszentmihalyi, 1975)

Flow is the state of "happiness"

When in *Flow*:

1. focused concentration on the present moment
2. loss of reflective self-consciousness
3. sense of personal control or agency over the situation or activity
4. a distortion of temporal experience (subjective experience of time is altered)
5. experience of the activity as intrinsically rewarding

Characteristics of the flow state




An illustration of the flow channel

## Malone (1981) : Challenge, Curiosity and Fantasy

**Challenge**

- Uncertainty of achieving a goal due to e.g. variable difficulty level, multiple level goals, hidden information, and randomness.

**Fantasy**

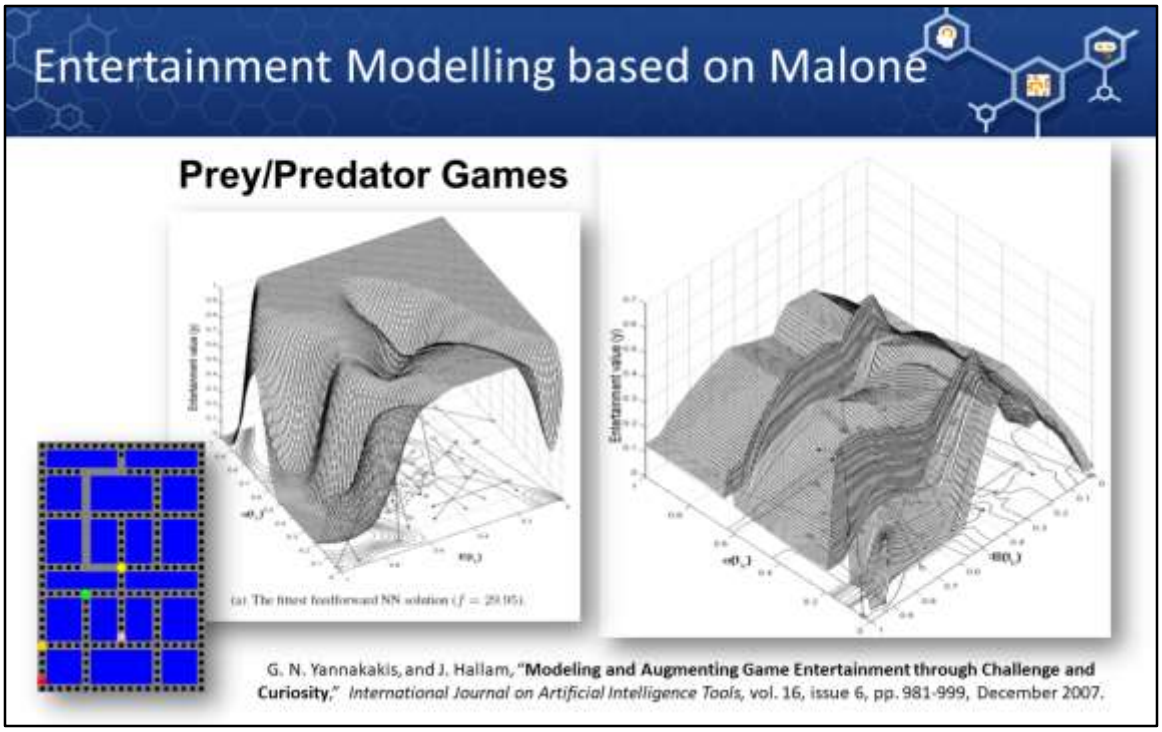
- Show (or evoke) images/situations/context not present
  - Extrinsic: depends on the skill used in a game
  - Intrinsic: related to the use of that skill.

**Curiosity**

- What will happen next to the game?
  - Sensory curiosity
  - Cognitive curiosity – it can be aroused via incomplete/inconsistent knowledge structures

T. Malone, **Toward a Theory of Intrinsically Motivating Instruction**, Cognitive Science, 1981

[Please refer to the cited paper for more details]




[Please refer to the cited paper for more details]

The graphs depict the function between entertainment (y), in-game challenge (E{t_k}) and in-game curiosity (σ{t_k}) in Pac-Man like games. The function is modelled through an artificial neural network (left image) or a neuro-fuzzy model (right image). Both models are trained via preference learning on self-reported pairwise preferences of entertainment from players of the game. Note the inverse U-shaped functions confirming Malone's assumption that average levels of challenge and curiosity yield, on average, higher degrees of entertainment.

# Entertainment Modelling based on Malone

**Playware Playground**

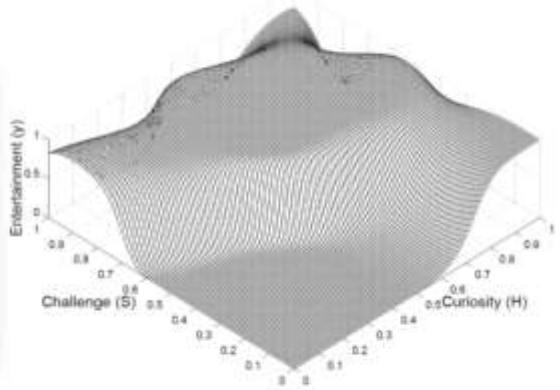


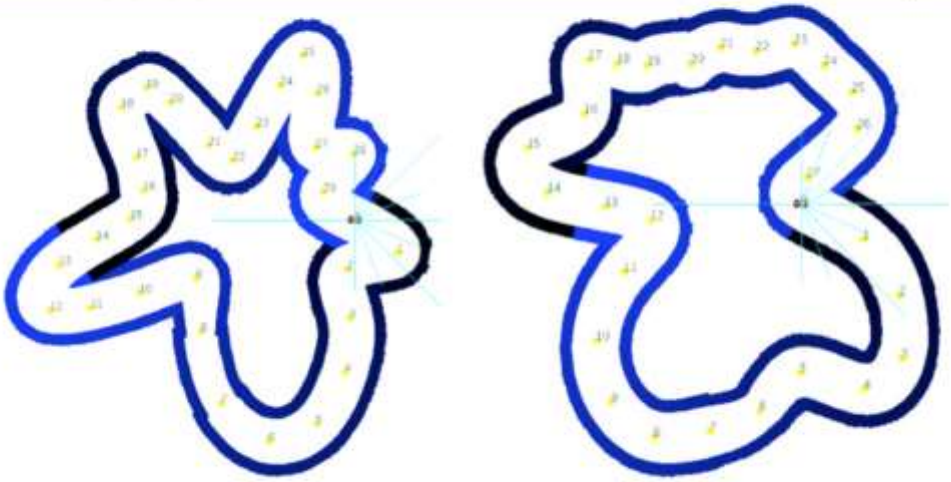

Fig. 4. Fittest feedforward NN ($f = 22.82$).

G. N. Yannakakis, and J. Hallam, "**Modeling and Augmenting Game Entertainment through Challenge and Curiosity**," *International Journal on Artificial Intelligence Tools*, vol. 16, issue 6, pp. 981-999, December 2007.

[Please refer to the cited paper for more details]

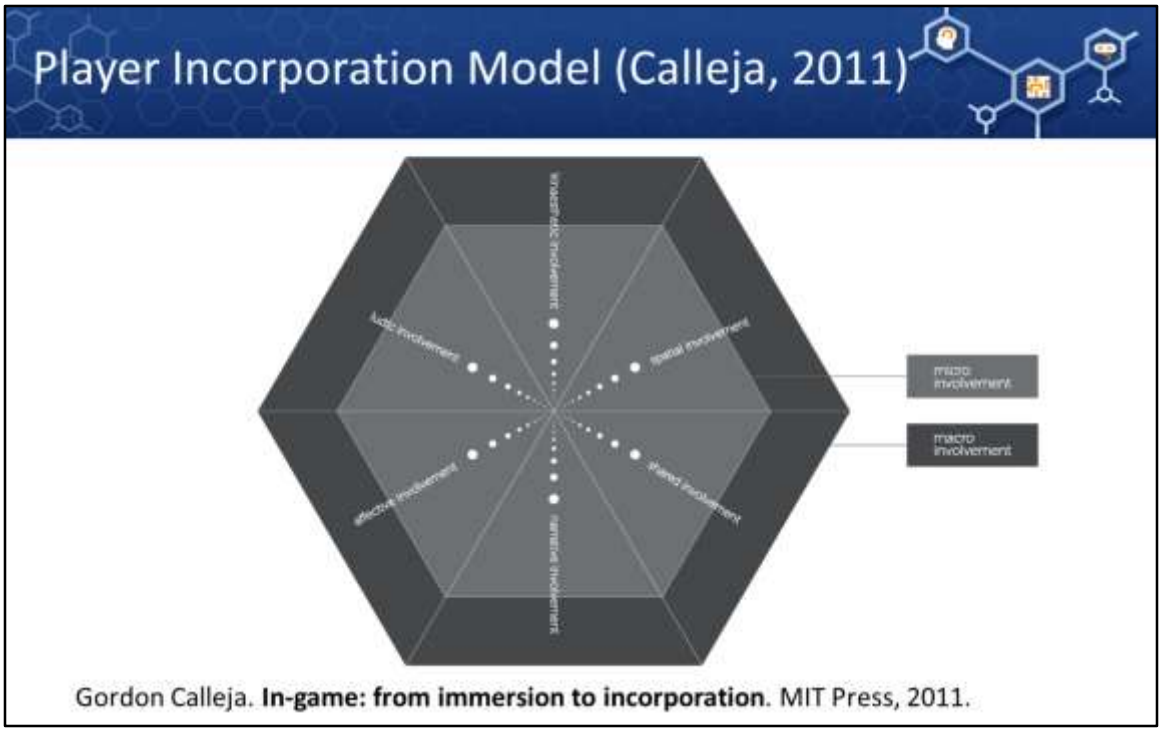
The graphs depict the function between entertainment (y), in-game challenge (S: game speed) and in-game curiosity (H: opponent appearance entropy) in physical interactive games (left image). The function is modelled through an artificial neural network (right image) trained via neuroevolutionary preference learning on self-reported pairwise preferences of entertainment from players of the game. Note the inverse U-shaped function confirming Malone's assumption that average levels of challenge and curiosity yield, on average, higher degrees of entertainment.

# Evolving Racing Tracks Based on Malone



Togelius, J., De Nardi, R., & Lucas, S. M. (2007). **Towards automatic personalised content creation for racing games.** In *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on* (pp. 252-259). IEEE.

[Please refer to the cited paper for more details]



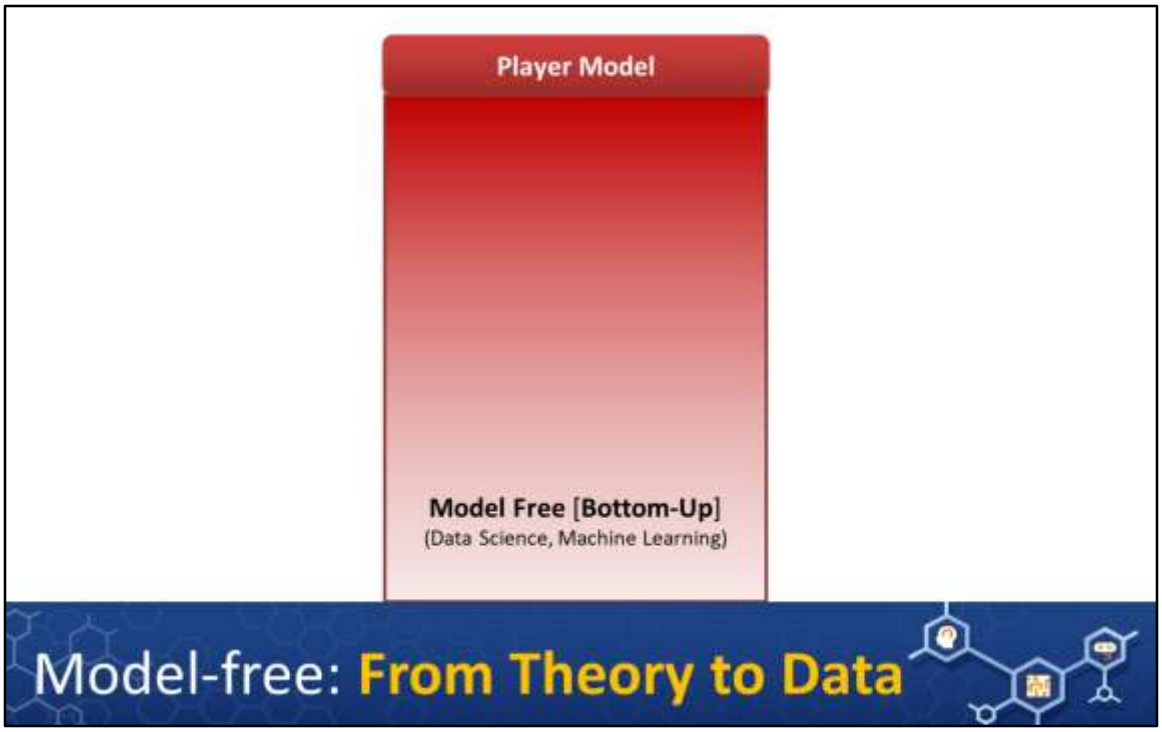

[Please refer to Calleja's book for more details]

Micro: moment to moment aspects of involvement

Macro: long-term aspects


Some observations and concerns about the similarities of developed theories on player experience.

With a careful analysis of the models proposed and their subcomponents one could coherently argue that there is one underlying theoretical model of player experience after all. While it is not the intention of this book to thoroughly discuss the interconnections between the aforementioned models it is worth pointing out a number of indicative examples of our envisaged overarching player experience model. An explorer (Bartle), for instance, can be associated with the easy fun factor of Lazzaro and the curiosity dimension of Malone. Further, the achiever archetype (Bartle) can be linked to the serious fun factor (Lazzaro). Accordingly, a killer archetype (Bartle) maps to the hard fun factor (Lazzaro), the challenge dimension of Malone's model, and a number of flow aspects. Finally, a socializer player profile (Bartle) could be associated to people fun (Lazzaro) and, in turn, to the shared involvement facet of Calleja.




[see Section 5.3.2 for more details]


[please refer to the cited book of Seif El-Nasr et al. for more details]

# Basic Definitions

- **Game analytics:** application of analytics to game development and research.
- **Goal:** support decision making, at operational, tactical and strategic levels for design, art, programming, marketing, user research, etc.
- **Game metrics:** interpretable measures of something related to games
  - User/Player metrics
    - User: customer, gameplay community
    - Gameplay: interface, in-game
  - System metrics

# What's the Problem?

- Understanding players
  - Behaviour
  - Player Experience
- How do people play a game?
- Do they play the game as intended?

# What's the Problem?

- Debugging costs!
  - Code debugging
  - Experience debugging
- Play testing costs!
- Non-linear games
  - Testing is challenge
- Improve next game release costs!
- …

# Game Data Visualization

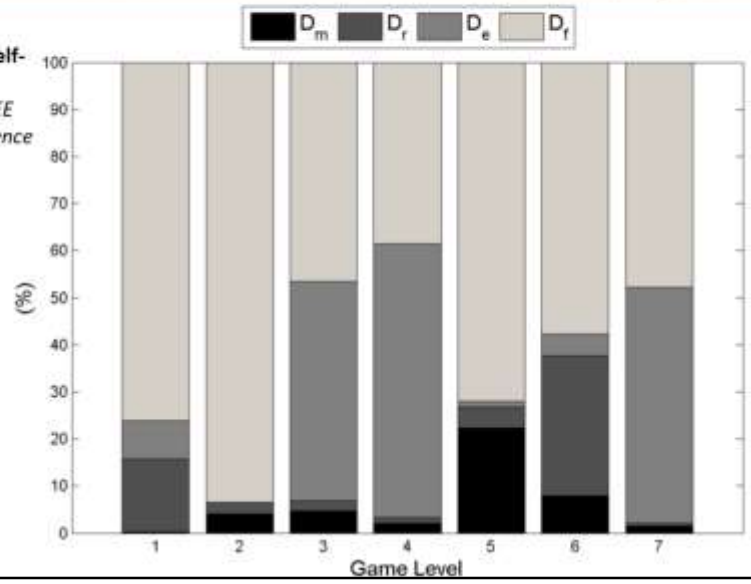
A core aspect of game analytics is the way we visualise data once we obtain it – we will dedicate a few slides on this aspect of data analysis in games

# Descriptive Statistics in Games

- Very important to visualise data prior to further processing
- Lots of information is hidden in basic statistics and analytics

# An Example: Tomb Raider: Underworld

A. Drachen, A. Canossa, and G. N. Yannakakis, "**Player Modeling using Self-Organization in *Tomb Raider: Underworld*,**" in*Proceedings of the IEEE Symposium on Computational Intelligence and Games*, Milan, September, 2




[for more details please refer to the cited paper]

Percentages of the four causes of death in Tomb Raider: Underworld over the game's 7 levels

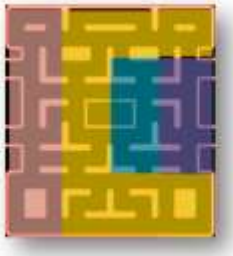
Sample size: 10,000 players

Causes of death:

- D_m: melee death from enemies
- D_r: ranged death from enemies
- D_e: environment
- D_f: falling

# Heatmaps

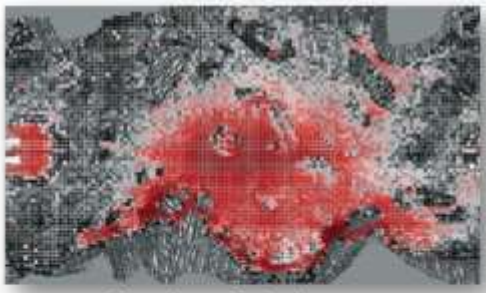
- Plot specific information over the map of a level
- Typically, counts of a particular event in a map location
- The higher the count, the higher the *temperature*



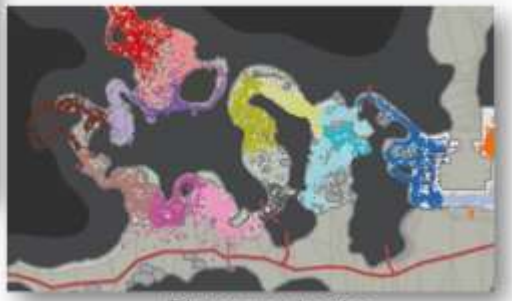
A heatmap is one of the most popular ways of vizualising game/player data. A heatmap reveals the relation between the distribution of the data and the game level.

# Heatmaps in *Halo*

Source: How Microsoft Labs Invented a New Science of Play, Thompson, Wired, 2007



Number of deaths



Player navigation

An example of dissimilar heatmaps in *Halo*


A heatmap example in Just Cause 2 (Sqare Enix Europe)

- More than 22.3 Million events
- Larger is white

Please refer to the cited paper for more details


The image depicts a menu heatmap of user clicks from game *Youda Lewel Shop* using the *Playtomic* heatmap function. *A* heatmpap as such can provide useful information on the use of buttons vs. the non-use of alternate use of buttons or other visual conten. In this example several players clicked on the dress to change it but the dress cannot change in this game.


Another example of a menu heatmap of user clicks from game *Stackwick Legacy* using the *Playtomic* heatmap function


Trajectory analysis is a decisive step from the spatial to the spatiotemporal space!

Example depicted in the image: DNA viewer used by Ubisoft

- While most players would take the parachute path during the tutorial (desired path) some would take the ground path which is far less fun.
- Ubisoft fixed the level so that no player takes the ground path and hence the game is more engaging for everyone.

# Game Analytics as a Service

Can be viewed as a service to

- **Game Developers** (publishers, designers, programmers etc.)
- **Third Parties** (recommendation systems, adverts, etc.)
- **Players** (own performance, tailored challenges, game aids, etc.)

Ben Medler. **Visual Game Analytics**, in El-Nasr et al (Eds.) *Game Analytics: Maximizing the value of Player Data.*

Please refer to the cited paper for more details

# Mining Data in Games

- Tracking data in games is a common practice
- The amount of data is usually huge
- One does not simply spot patterns in the data
- Data mining provides methods for finding **regularities** and **anomalies**

Drachen, Anders, Christian Thurau, Julian Togelius, Georgios N. Yannakakis, and Christian Bauckhage. "**Game data mining**." In *Game analytics*, pp. 205-253. Springer, London, 2013.

## Game Data Mining: What can it do for you?

- Supplementary approach to traditional testing
- Imitate human playing styles
- Identify player profiles
- Spot cheating
- Spot game design flaws (e.g. “sweet spots” and frustrating sections)

## Game Data Mining: What can it do for you?

- Tell you when and why players stop playing
- Enable player-driven adaptive games
- Overall, help you make better games
  - —Project management
  - —Marketing
  - —Customer care

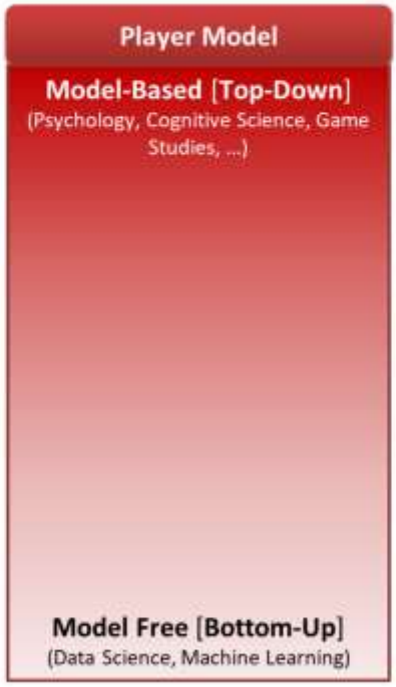
## Game Data Mining: What can it do for you?

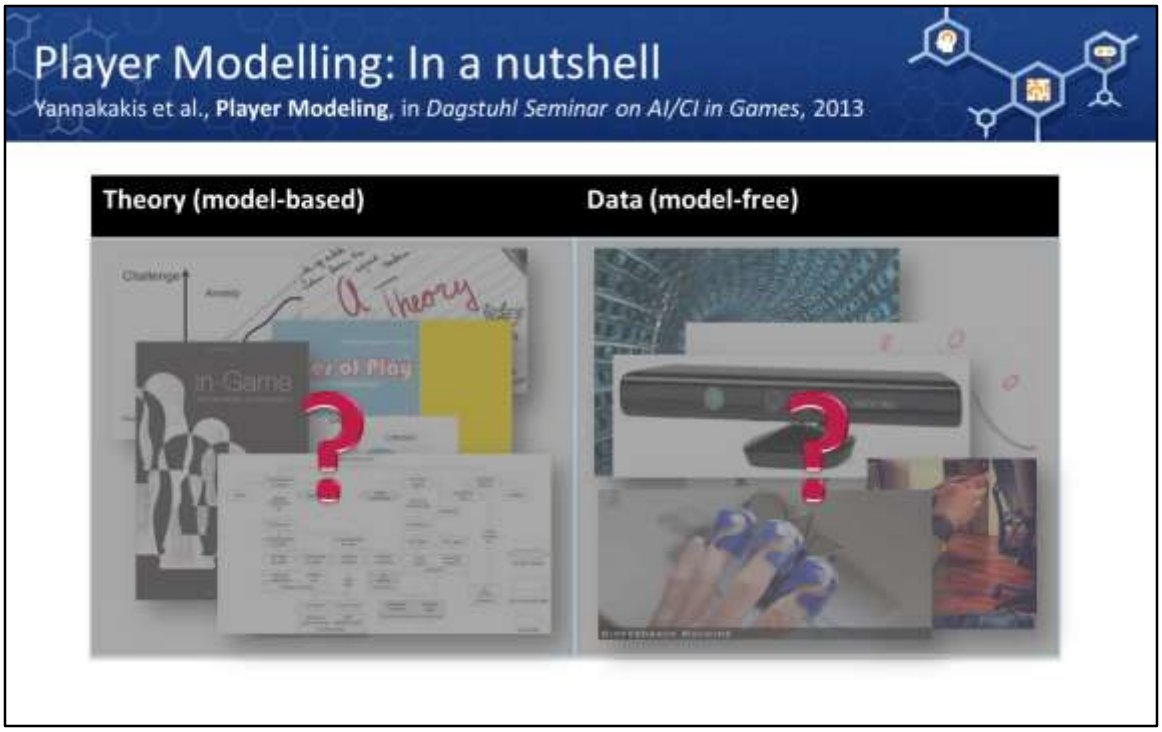
A few examples

- Find weak spots in the design of game
- Which assets that are not getting used
- Figure out how players spend their time when playing
- Predict when they will stop playing
- Predict what they will do while playing
- Discover gold farmers in an MMORPGs
- Explore and use of social grouping
- Figure out how to convert non-paying to paying users

2 main uses: understanding players (behaviour, style, experience) and testing (Players vs game design)

Testers vs alpha players | Code and experience debugging | Nonlinear games


Player Model
Model-Based [Top-Down]
(Psychology, Cognitive Science, Game Studies, ...)
Model Free [Bottom-Up]
(Data Science, Machine Learning)



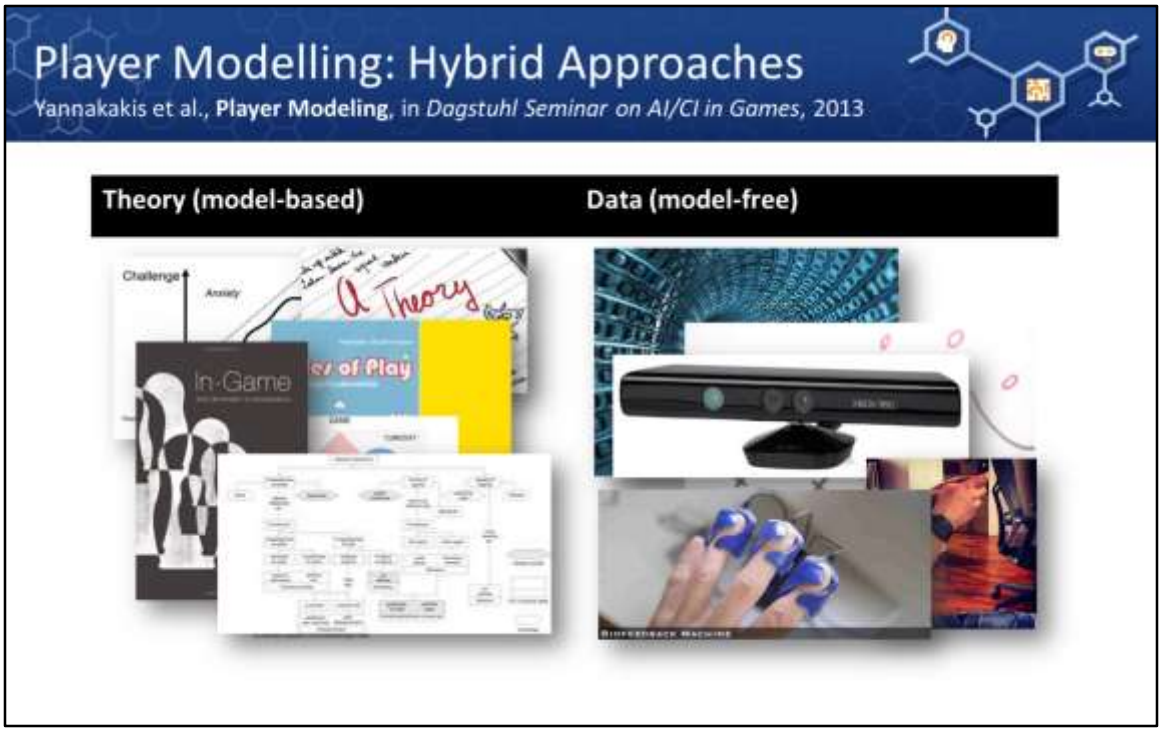

Please refer to the cited paper and section 5.3 for more details

Both Player Modeling approaches have their limitations

# Player Modelling: Limitations

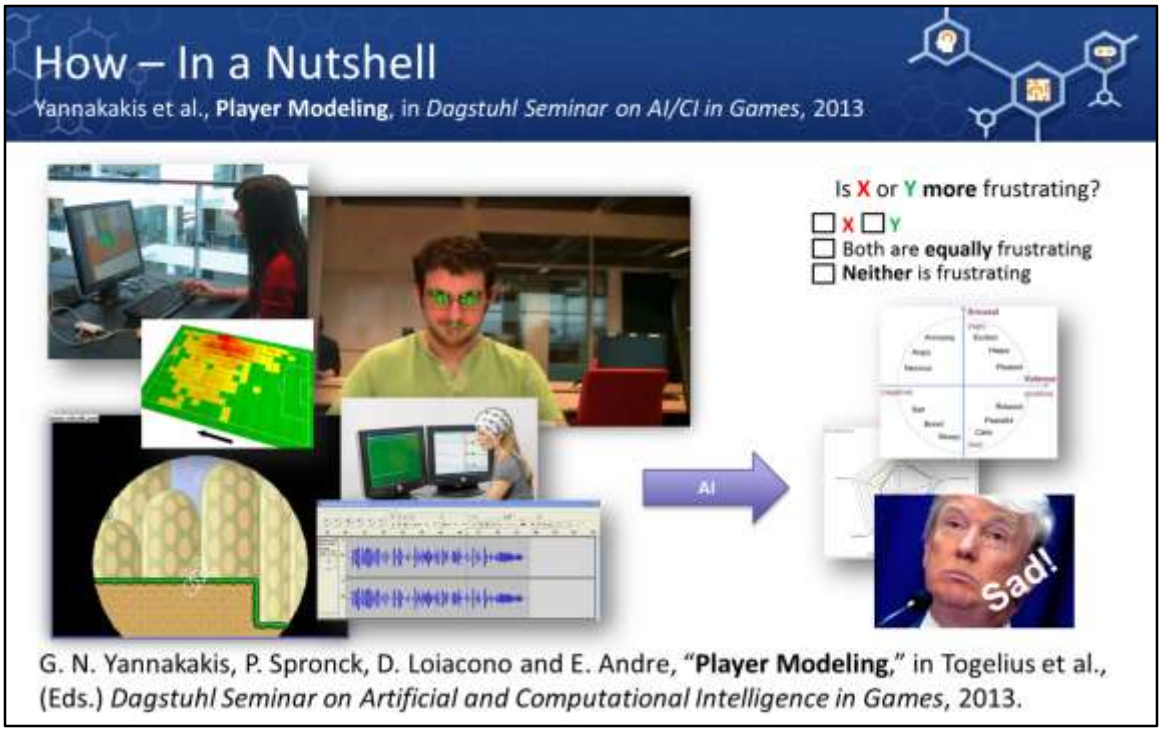
Yannakakis et al., **Player Modeling**, in *Dagstuhl Seminar on AI/CI in Games*, 2013

- **Model-based** (theory-driven):
  - The majority of models have not been tested on or derived from interactive media
  - Models are not cross-validated (they should be!)
  - Agent-based models (OCC, BDI) – rather limited to agent-based interaction (it's limited in games)
- **Model-free** (data-driven):
  - Big data, crowdsourcing, elegant ML (e.g. deep learning / sequence mining), sensor technology
  - Supreme and obvious given the availability of big data and crowdsourcing
  - Problems: quality / quantity of data
  - Problems: garbage in – garbage out



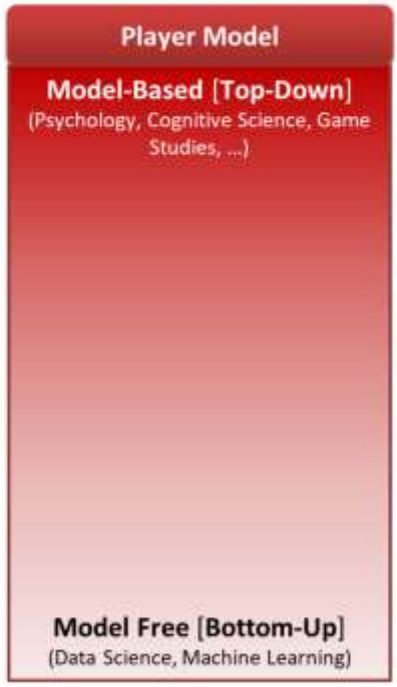

[see Section 5.3. for more details]

The space between a completely model-based and a completely model-free approach can be viewed as a continuum along which any player modeling approach might be placed. While a completely model-based approach relies solely on a theoretical framework that maps a player's responses to game stimuli, a completely model-free approach assumes there is an unknown function between modalities of user input and player states that a machine learner (or a statistical model) may discover, but does not assume anything about the structure of this function. Relative to these extremes, the vast majority of studies in player modeling may be viewed as hybrids that synergistically combine elements of the two approaches.

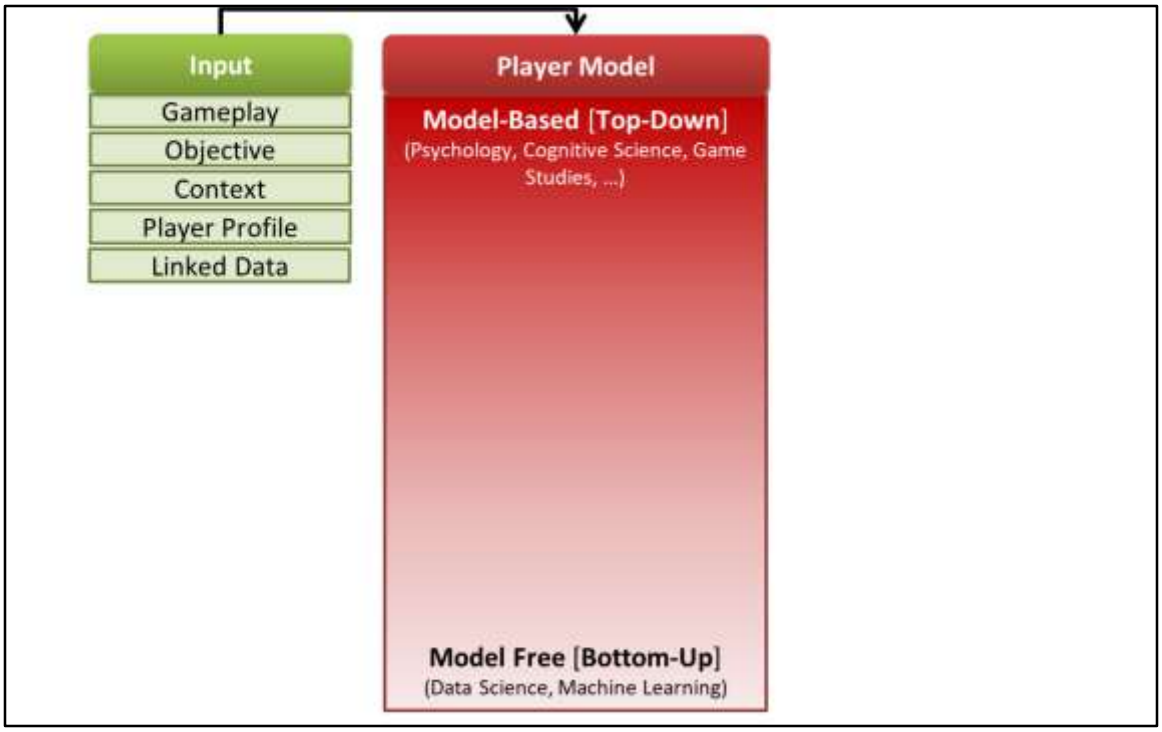



[see section 5.3 for more details]

- On the left (input) there is rich input that comes from multiple modalities of player signals (manifestations of experience) including gameplay data, physiology, gaze tracking etc.
- On the right (output) there are labels of experience (annotators) in various formats
- In between there is a computational model (AI) that maps between the two




Note: The hybrids continuum between top-down and bottom-up player modeling approaches is illustrated with a gradient color in the figure above




[see Section 5.4 for more details]

By now we have covered the various approaches available for modeling players and we will now focus on what the **input** of such a model might be like. The model's input can be of three main types: (1) anything that a player is doing in a game environment gathered from gameplay data—i.e., behavioral data of any type such as user interface selections, preferences, or in-game actions; (2) objective data collected as responses to game stimuli such as physiology, speech and body movements; and (3) the game context which comprises of any player-agent interactions but also any type of game content viewed, played, and/or created.

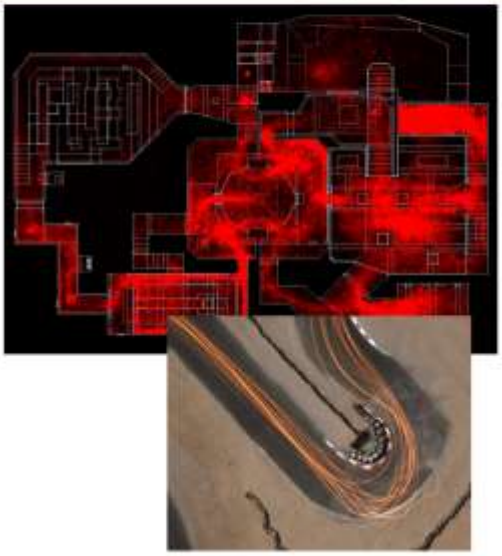
We also discuss static profile information on the player (such as personality) as well as web data beyond games that could feed and enhance the capacity of a player model.

# Gameplay

# Gameplay Input

- Player game preferences, behavioral patterns
- Examples: tactics, strategy, play patterns, clickthroughs, deaths, weapon selection, character selection, etc…

- **Common features:** **Micro** vs **macro** actions
- **Pros:** real-time efficiency
- **Challenge:** we can't tell much beyond player behavior…



[see Section 5.4.1 for more details]

# Objective Input

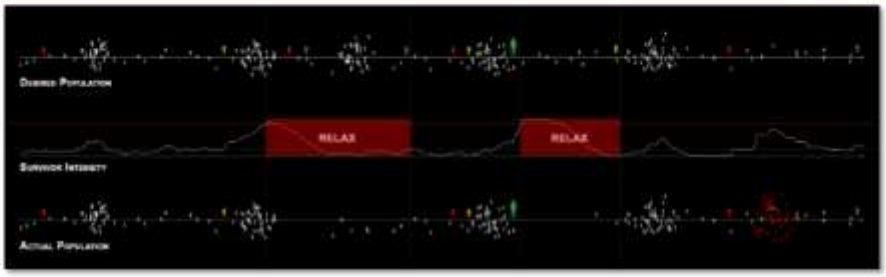
- Bodily and physiological manifestations of gameplay
- **Pros:** reliable measures of user experience
- **Challenges:** many; let's see them in more detail

[see Section 5.4.2 for more details]

# Objective Input – ways

- Cameras
  - Face
  - Body movement
  - Eye movement
- Other sensors/devices
  - Physiology (heart rate, skin conductance, …)
  - Muscle activation (EMG)
  - Brain activity (EEG)
  - Speech

# Physiology




- **Common features:** summarization, time and frequency domain
- **Pros:** directly linked to arousal – immediate response
- **Challenges:** signal denoising/normalization; control for subjectivity of physiological responses, law of initial values, habituation, rebound

Challenges of physiological signals

- Measuring physiology can be obtrusive and noisy
- The less intrusive the sensor, generally, the less accurate
- Law of initial values: response depends of current status
- Habituation: when the stimulus is presented repeatedly, the physiological responsivity decreases (linked to changes in effect response?)
- Rebound: after the stimulus, signals return to lower levels than before the stimulus
- Subject dependent levels: baselines

## Visual Cues

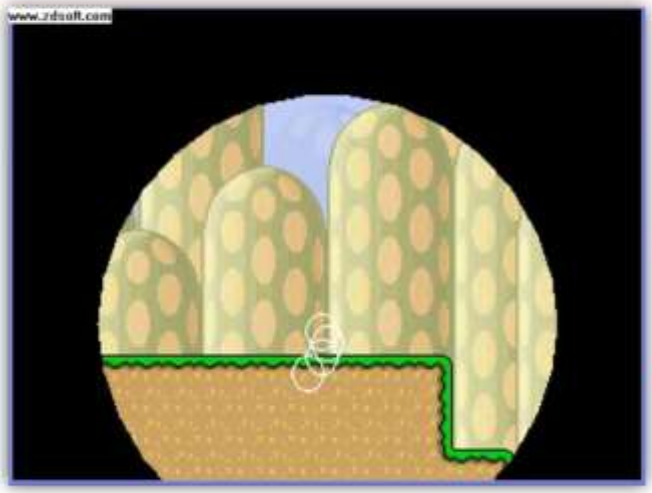
- **Common features:** action unit detection, head pose stats
- **Pros:** every laptop has a camera, off-the-shelf cheap solution, natural interaction
- **Challenges:** do we really express emotions (facially) while playing? Head-pose might be more relevant? Models rely on posed/acted expressions

- Most research on emotion modelling from facial expressions is based on posed expressions
- What are gamers' facial expressions though?
- Research area scarcely explored


The images depict a number of available tools for video-based affect detection. In particular

- Intel realsense – gesture recognition [top right]
- Affectiva's AffDex – facial features detection and emotion detection [left]
- Openface – facial features detection [bottom right]

# Eye-tracking




- **Common features:** blinks and gaze fixation (indicators of attention and engagement); total time spent looking at particular objects
- **Pros:** you know where your player looks at/focuses on/pays attention to
- **Challenges:** practicality, lab conditions (illumination), pupilometry doesn't really work in games

J. Munoz, G. N. Yannakakis, F. Mulvey, D. Witzner, G. Gutierrez and A. Sanchis, "**Towards Gaze-Controlled Platform Games**," in *Proceedings of 2011 IEEE Conference on Computational Intelligence and Games*, 2011.

# Speech

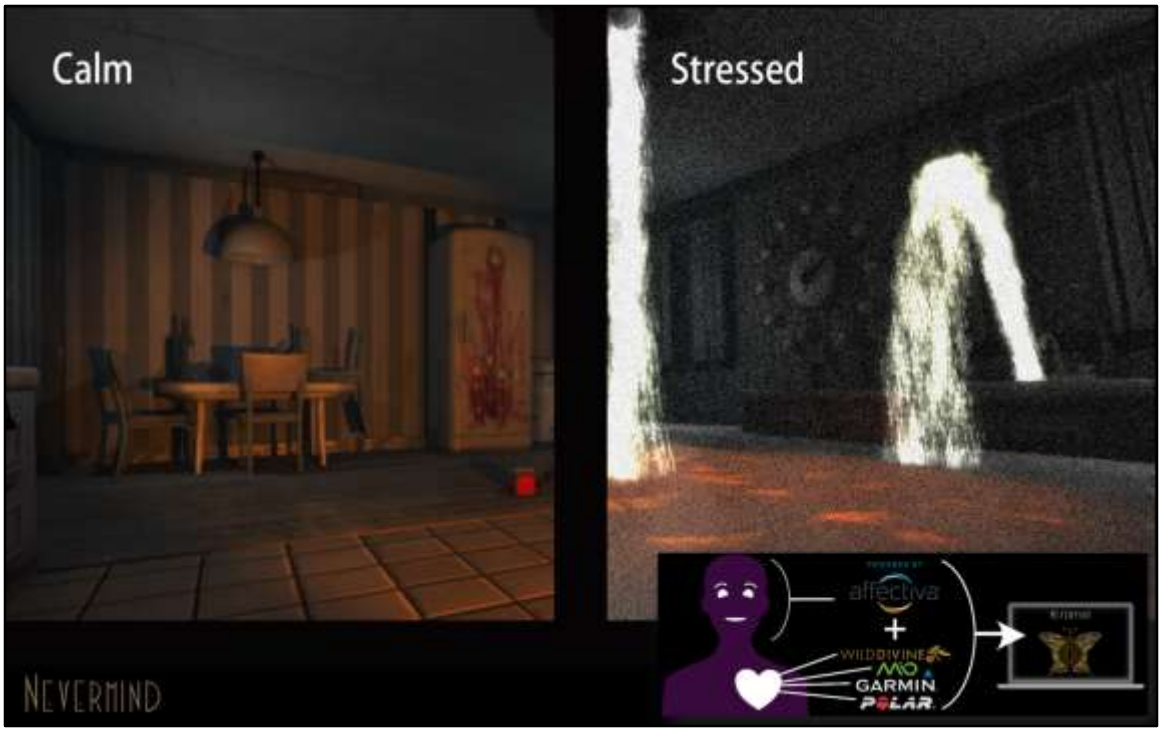
- **Common features:** frequency-based, detection of spoken words and other communication components (e.g. laughter and pauses)
- **Pros:** speech (pitch, loudness, quality) is linked to emotions (arousal/valence); useful in game-child interaction studies
- **Challenges:** verbal cues are rare; environment noise; multi-player games

# Tools for Speech/Music-based Feature Extraction

openSMILE:)
by audEERING™

https://audeering.com/technology/opensmile/

Eyben, Wöllmer and Schuller: "**openSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor**", In Proc. ACM Multimedia (MM), ACM, Florence, Italy, ACM, ISBN 978-1-60558-933-6, pp. 1459-1462, October 2010.

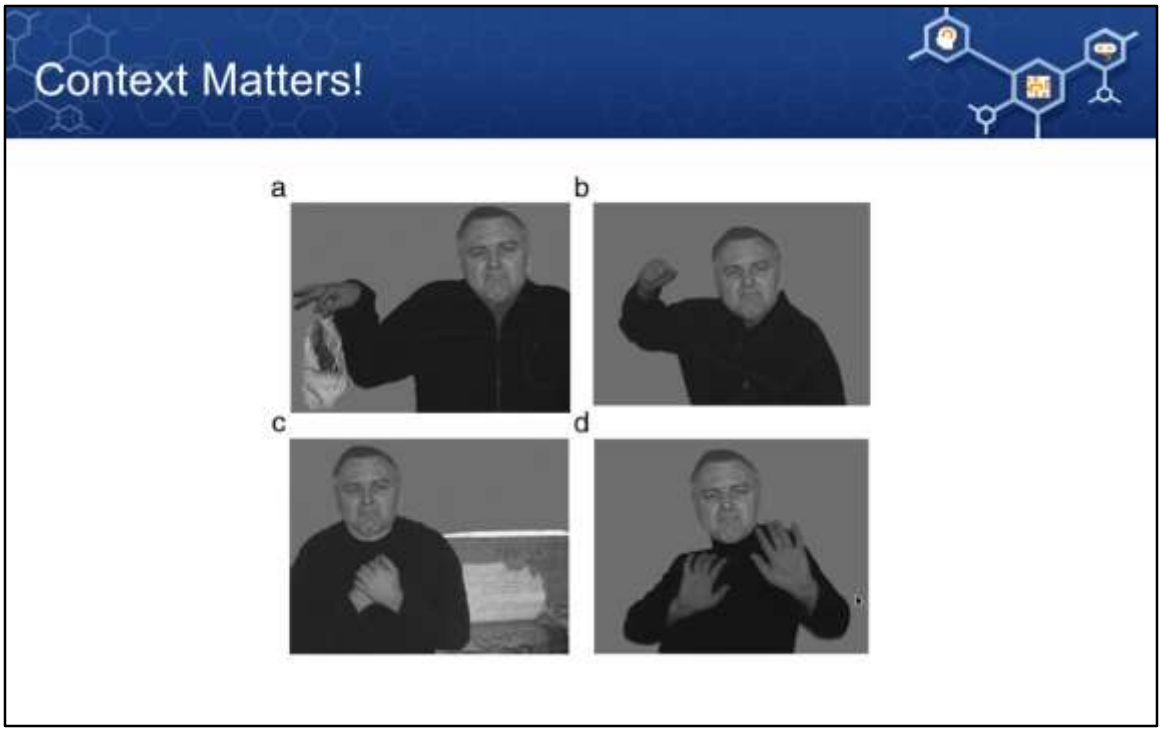



A great example of a biofeedback-adaptive game is *Nevermind* by Erin Reynolds. The game features a number of sensors for the multimodal tracking of players. The sensors detect the level of stress of a player which is used to modify the content of the game.

# Game Context

[see Section 5.4.3 for more details]

Context is very important in player modelling. Let us suppose you detect high arousal (e.g. through the player's skin conductance) at a particular moment in the game. What would high arousal actually mean without knowing what happened in the game? Was it player excitement or player frustration?

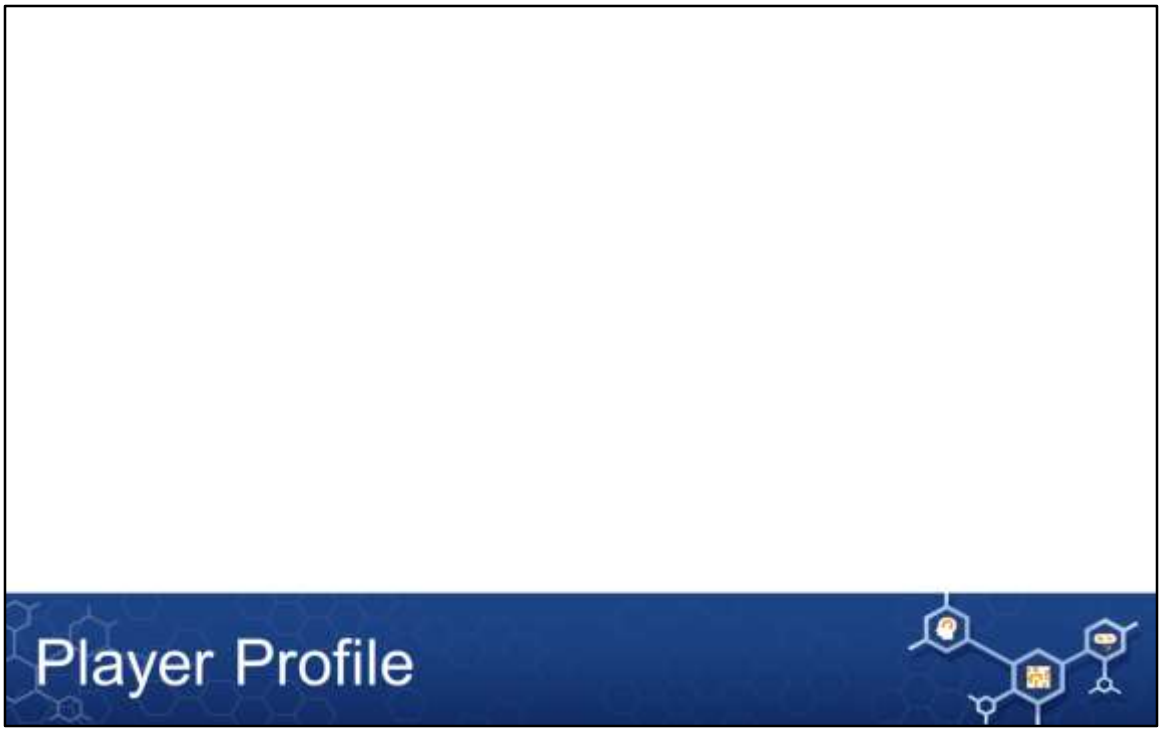



[see Section 5.4.3 for more details]

The images depict a famous example within the affective computing literature: four different emotional states can be identified across the four images even though the facial expression of the person remains the same. What changes is the context!

The world around (and within) us is tied to our emotional reactions. Affect can be seen as appraisal of game events! While very detailed information is easily available we have no standard (or general) way of extracting features for affect detection. In particular those are:

- Game dependent
- Genre dependent
- Player dependent




[see Section 5.4.4 for more details]

# Player Profile

- Player profile
  - Information about ones' personality, demographics, culture, age, gender, experience with games etc…
  - In general information that does not change due (or not altered via) games – at least not that rapidly…
- A player profile can form additional input(s) to a player model

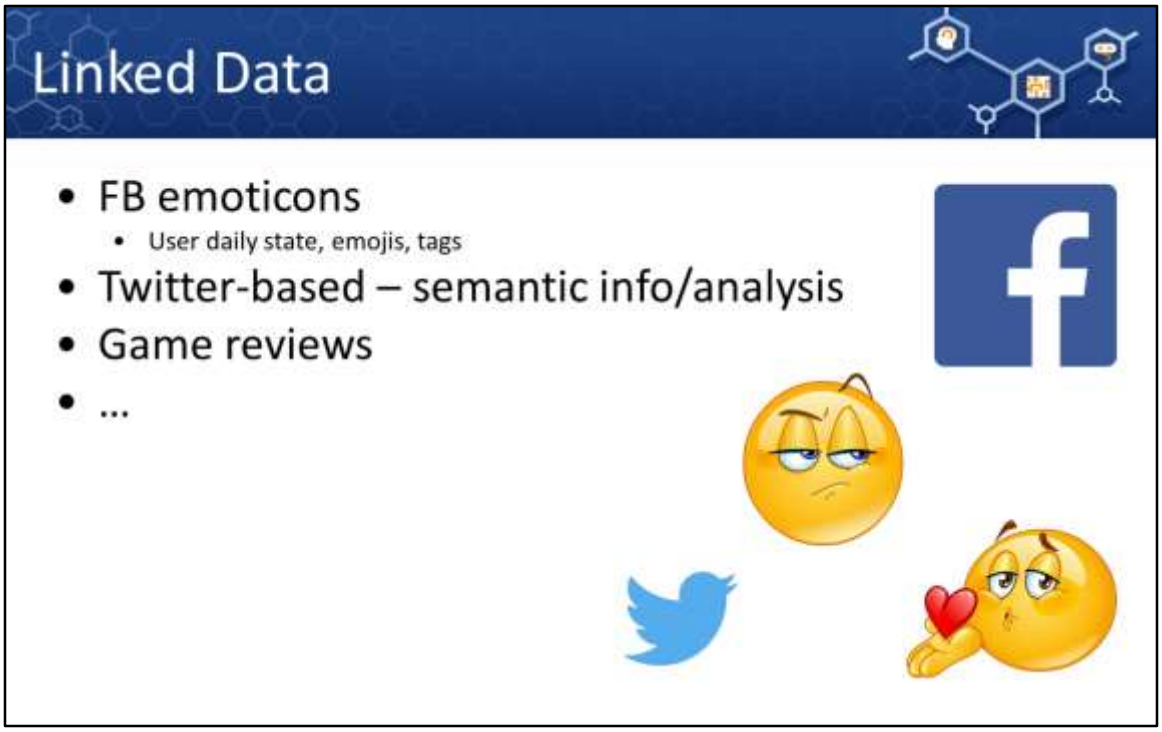
A player profile includes all the information about the player which is static and it is not directly (nor necessarily) linked to gameplay. This may include information on player personality (such as expressed by the Five Factor Model of personality, culture dependent factors, and general demographics such as gender and age. A player's profile may be used as input to the player model to complement the captured in-game behavior with general attributes about the player. Such information may lead to more precise predictive models about players.

# Reminder: Player Profile vs. Player Model

- What are the differences?
  - A **profile** is built on *static* data and not influenced by the game
  - A **model** is built on *dynamic* data from the gaming interaction and is (temporally) influenced by the game

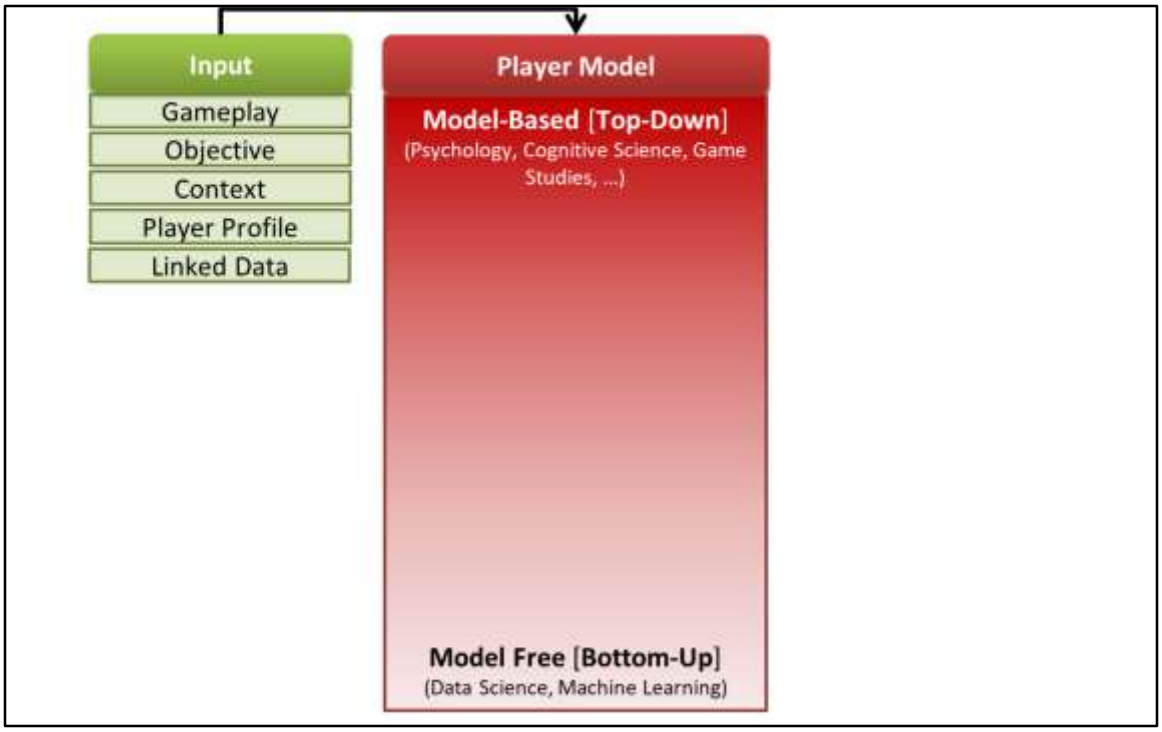
# Linked Data

[see Section 5.4.5 for more details]




[see Section 5.4.5 for more details]

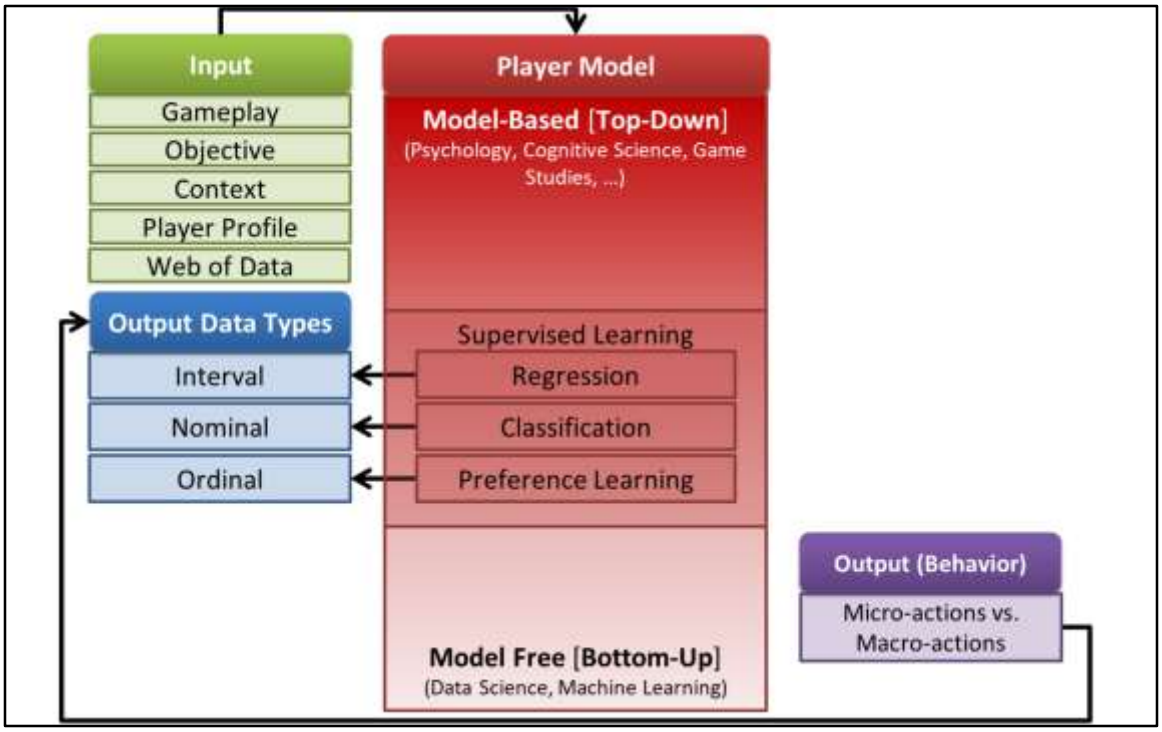
Somewhere between the highly dynamic in-game behavior and the static profile information about the player we may also consider linked data retrieved from web services that are not associated with gameplay per se. This data, for instance, may include our social media posts, emoticons, emojis [199], tags used, places visited, game reviews written, or any relevant semantic information extracted from diverse Web content.




[see Section 5.5 for more details]

Having outlined all the possible options for the model of the input we now move on to the model's **output.** The model's output, i.e., that which we wish to model, is usually a representation of the player's state. In the following slides we explore three options for the output of the model that serve different purposes in player modeling. If we wish to model the experience

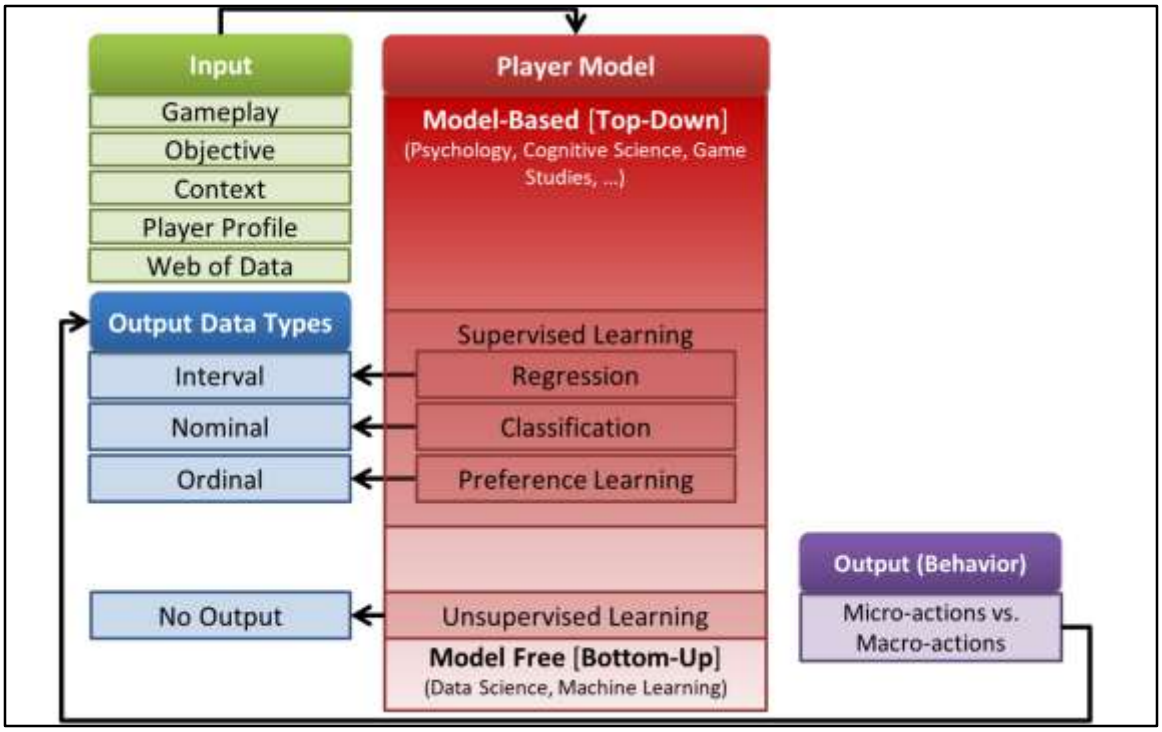
of the player the output is provided predominately through manual annotation. If instead we wish to model aspects of player behavior the output is predominately based on in-game actions. Finally, it may very well be that the model has no output.




We start by looking at behavioural data that have associated labels [see Section 5.6.1 for more details]

Player modeling consists of finding a function that maps a set of measurable attributes of the player to a particular player state. Following the supervised learning approach this is achieved by machine learning, or automatically adjusting, the parameters of a model to fit a dataset that contains a set of input samples, each one paired with target outputs. The input samples correspond to the list of measurable attributes (or features) while the target outputs correspond to the annotations of the player's states for each of the input samples that we are interested to learn to predict. As mentioned already, the annotations may vary from behavioral characteristics, such as completion times of a level or player archetypes, to estimates of player experience, such as player frustration.

Popular supervised learning techniques, including artificial neural networks (shallow or deep architectures), decision trees, and support vector machines, can be used in games for the analysis, the imitation and the prediction of player behavior, and the modeling of playing experience. The data type of the annotation determines the output of the model and, in turn, the type of the machine learning approach that can be applied. The three supervised learning alternatives for learning from numerical (or interval), nominal and ordinal annotations are, respectively, regression, classification and preference learning.




We will also look at examples of datasets that do not have any target outputs associated with it. [see Section 5.3.3 and 5.6.3 for more details]

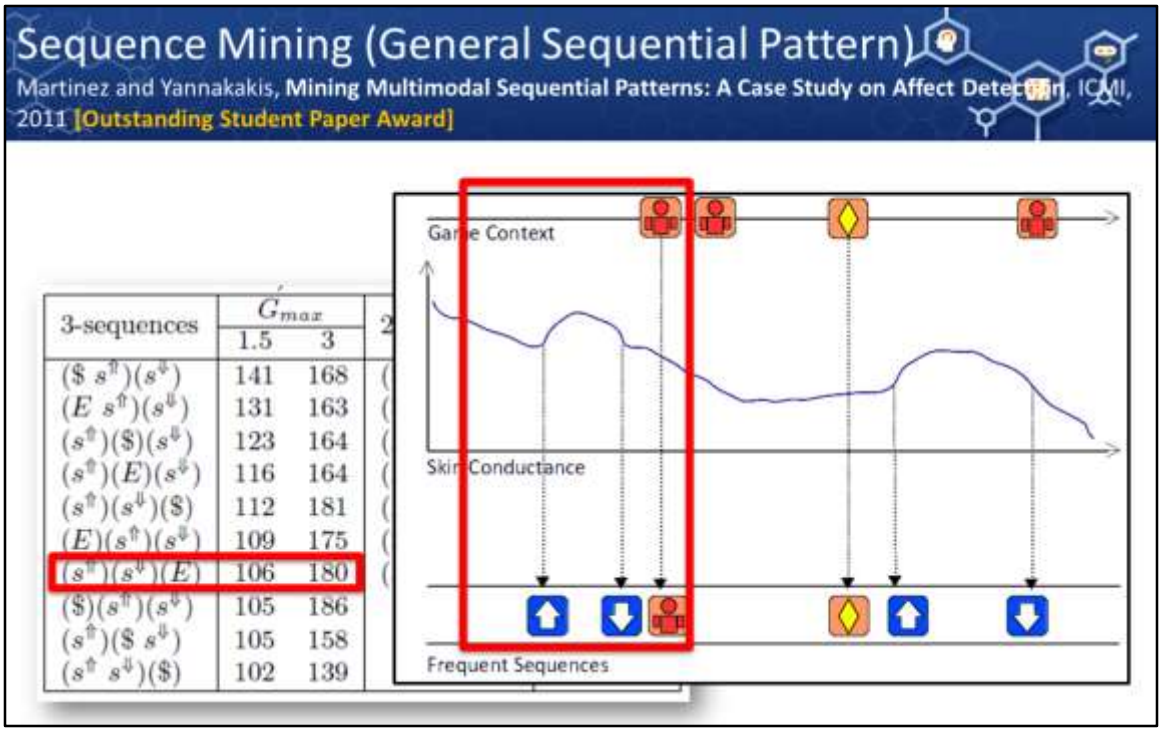
Very often we are faced with datasets where target outputs about player behavioural or experience states are not available. In such instances modeling of players must rely on unsupervised learning. Unsupervised learning, focuses on fitting a model to observations by discovering associations of the input and without having access to a target output. The input is generally treated as a set of random variables and a model is built through the observations of associations among the input vectors.

Unsupervised learning as applied to modeling players involves tasks such as clustering and association mining.


[see Section 5.7.2.2 for more details]

We use the Mazeball (3D prey-predator game) dataset as a paradigm for processing player data without labels

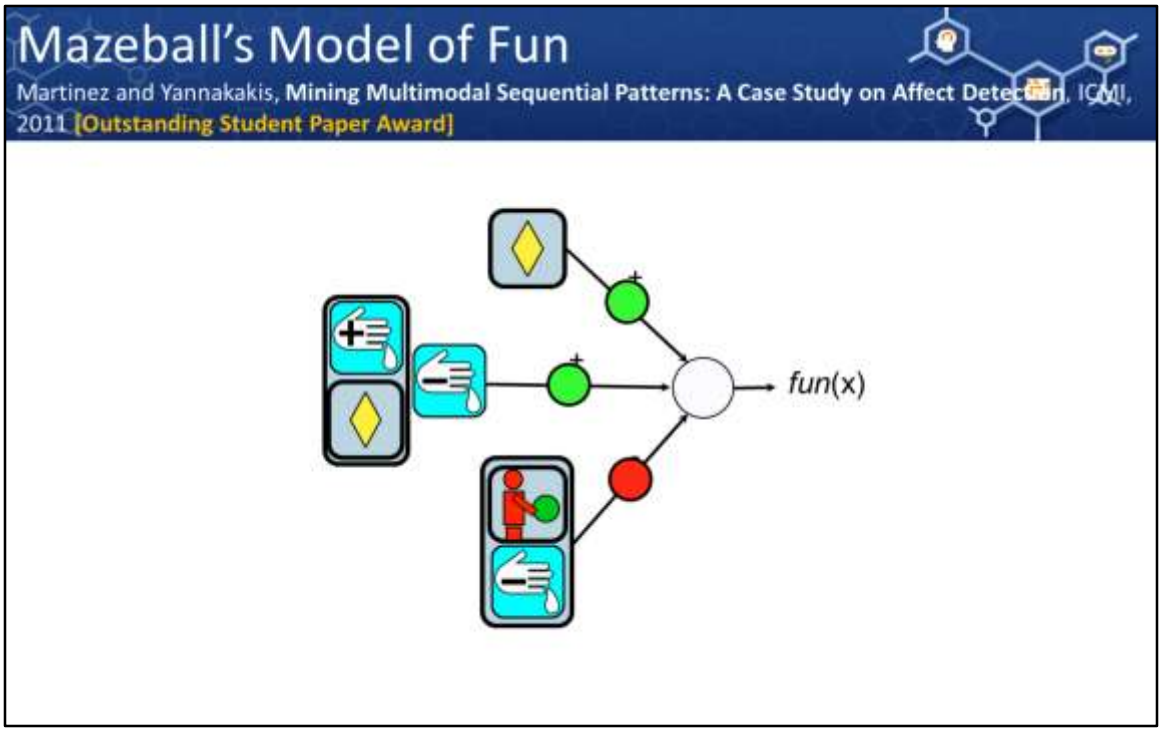
More details can be found at http://hectormartinez.com/




Association mining for feature extraction on the multimodal input space via **Generalized Sequential Pattern** Algorithm **(GSP)**

- GSP is a very simple method that reveals frequent sequences of data
- The algorithm couples signals at the feature level
- Evidence suggests that it is a very useful technique for feature extraction of multimodal player data (galvanic skin response and gameplay in this example)

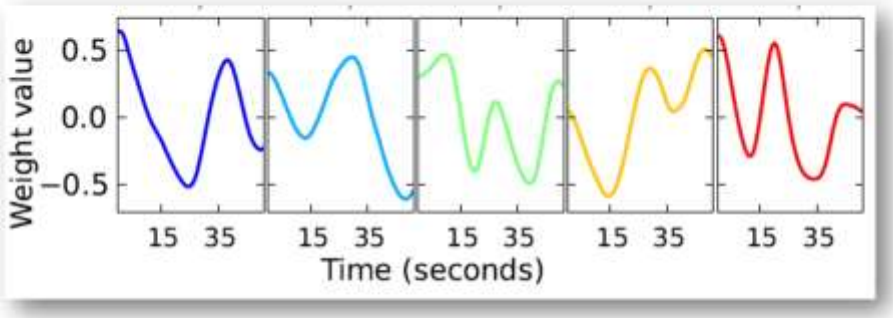
For more details please refer to the cited paper




The image depicts the derived model of fun (as a simple perceptron of three inputs) based on the most frequent sequences of multimodal data. The three inputs considered were from top to bottom: 1) pellets gathered 2) number of times the skin conductance of the player increased and then decreased after a pellet was gathered and 3) number of times the skin conductance decreased in conjunction with loosing by an opponent.

The model is trained via neuroevolutionary preference learning on pairwise comparisons of games with regards to "fun"

For more details please refer to the cited paper



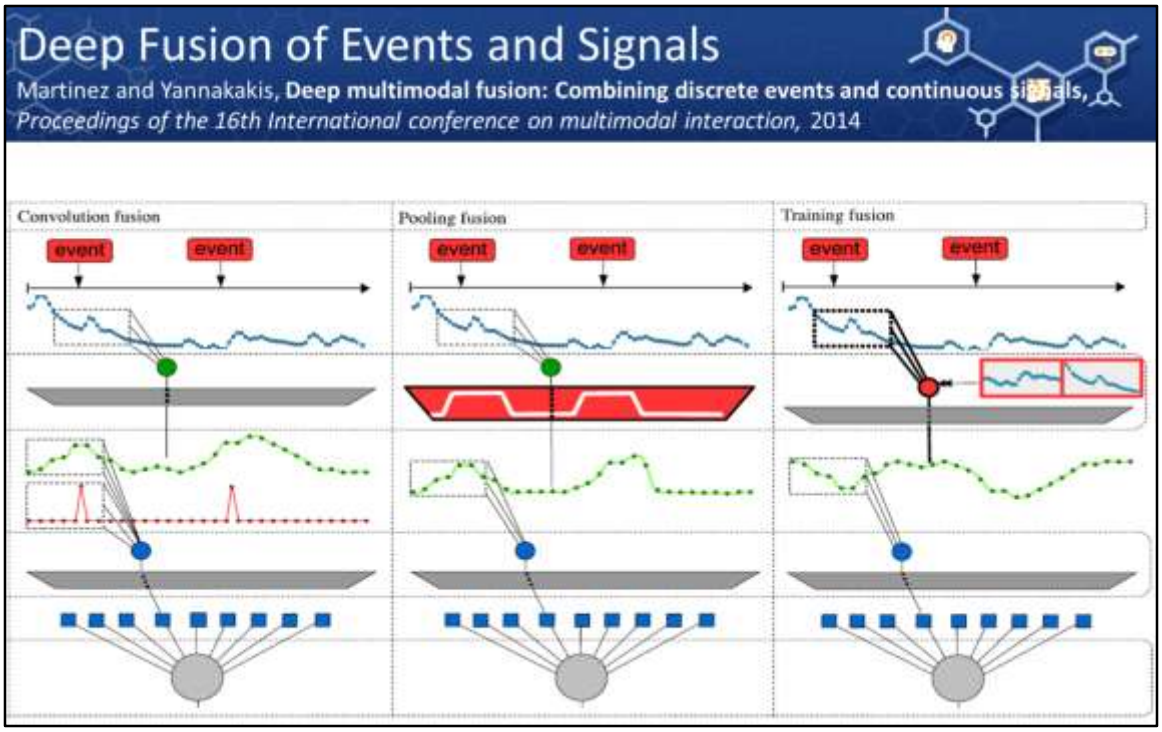

## Convolution for Affect Detection

Martinez, Bengio and Yannakakis, **Learning Deep Physiological Models of Affect**, *IEEE Computational Intelligence Magazine*, 2013

An example of the application of convolutional neural networks (CNNs) on skin conductance signals for affect detection

- Deep learning : supreme method for image and speech recognition. It automatically designs feature sets – most distinct patterns (main factors of variation in data). CNNs are applied directly to the signal and any signal type (discrete/continuous)
- Key finding: deep learning captures efficient temporal features of skin conductance (see the 5 images above) that no ad-hoc feature can capture

For more details please refer to the cited paper




In (Martinez and Yannakakis 2014) we explored the various ways we can fuse discrete events and continuous signals retrieved from games via CNNs.

The first convolutional layer receives as input a continuous signal at a high time resolution, which is further reduced by a pooling layer. The resulting signal (feature map) presents a lower time resolution. The second convolutional layer can combine this feature map with additional modalities at the same low resolution.

- Convolution fusion network (left figure): the two events are introduced at this level as a pulse signal [equals **one** only when the event was recorded and **zero** otherwise].
- **Pooling fusion** network (middle figure): events are introduced as part of the first pooling layer, resulting in a filtered feature map [simple filter that equals 1 within the time window selected and decays linearly to zero].
- Training fusion network (right figure): events affect the training process of the first convolutional layer [resampling.

## Example (Player Behavior Modeling)
## Tomb Raider: Underworld

Drachen, Canossa & Yannakakis, **Player modelling using self-organisation in *Tomb Raider: Underworld***, IEEE CIG 2009

[see 5.7.1.1 for more details]

We will now look at another unsupervised learning example. This time we will model player behavior (not experience) in the popular *Tomb Raider: Underworld* (TRU) game

For further information about this study please refer to the cited paper

# Motivation

- How do people play TRU?
- User testing: Do people play TRU as intended?
- Player modeling using *metrics* via machine learning → alternative quantitative approach to traditional qualitative approaches of user and playability testing

Our key motivations for this study

# Clustering in TRU

- Commercial major game test-bed: Tomb Raider Underworld
- Large-scale data collection (1365 players)
- Data is *clean* and *live:* gathered in a natural setup via an industrial logging system and a commercial web service
- First application of SOMs on high-level behaviors of completed games
- Directly addressing game-industry requirements
  - limitations of scalability and commercial-game practicality are minimized
  - bridging the AI gap

# Player Data Collection

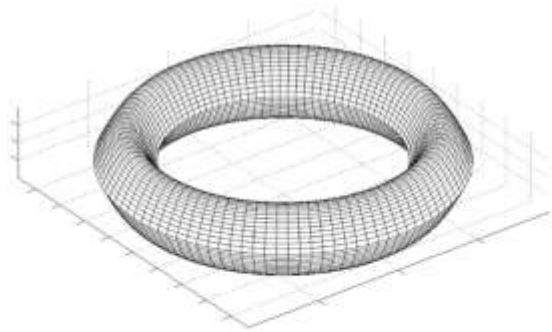
- EIDOS Metrics Suite software
  - Record gameplay data (*metrics*) of EIDOS games
  - Data are stored in an SQL-server via ELT process
  - Several Features are extracted (work done by *Crystal Dynamics*)
    - 3D coordinates, completion time, # deaths etc.
- **Live Data** (xBox Live!)
  - Published version of TRU played in gamers' natural habitats
  - Data free from laboratory bias and experiment expectancy effects
- Data collection during November 2008
  - 25240 Players
  - 1365 of those completed the game
  - Data are stored for all 7 levels (100 map units)

# Extracted Features

- **Causes of Death (% over total number of deaths)**
  - **Opponent** – 28.9% of all deaths – Min: 6%, Max: 60%
  - **Environment** – 13.7% of all deaths – Min: 2%, Max: 45%
  - **Falling** – 57.2% of all deaths – Min: 27%, Max: 83%
- **Total Number of Deaths** – 140 on average – Min: 16, Max: 458
- **Completion Time** – 550 min on average – Min: 3h, Max: 29h
- **Help-on-Demand (# hint + answer requested)** – 29 on average – Min: 0, Max: 148

# Emergent Self-Organizing Maps (ESOMs)

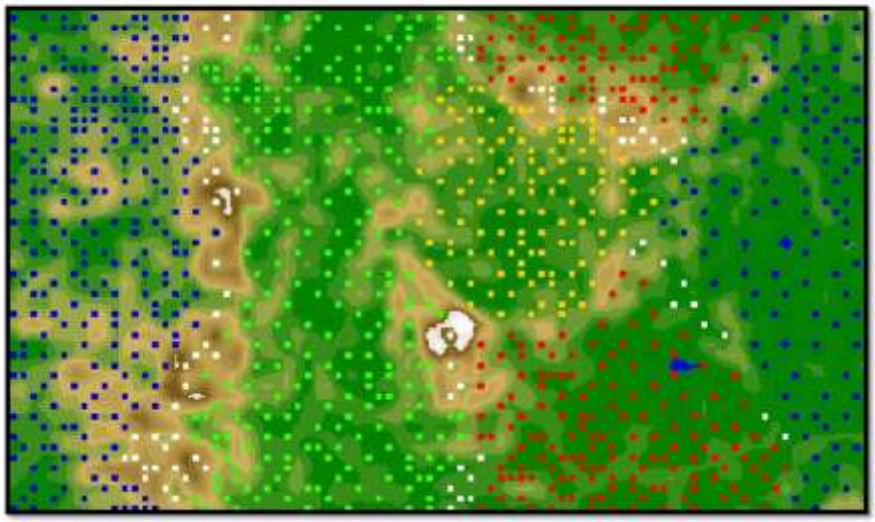
- Unsupervised learning through self-organization of a neuron map
- Dimensionality reduction to 2D
- Training
  - Batch
  - Toroid topology (50X100 neurons)
  - Rectangular grid
- Performance Measures
  - Topographic error
  - Quantization error



ESOM: Large-scale SOMs (The power of self-organization — which generates emergence of structure in the data — is disused when small SOMs are utilized.)

The outcome of SOM training is that **neighboring neurons have similar weight vectors** which can be used for projecting data to 2D

# Clustering in TRU (ESOMs)



Drachen, Canossa & Yannakakis, **Player modelling using self-organisation in *Tomb Raider: Underworld***, IEEE CIG 2009

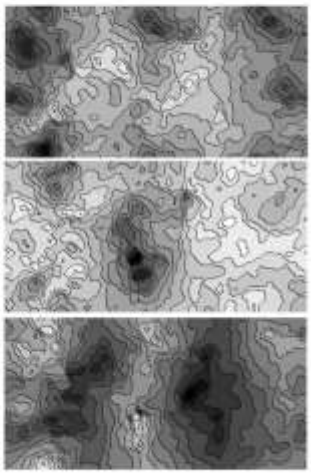
E-SOMs (neural nets)

Visualization:

–U-matrix: local distance structure in the data (distance of neighboring weight vectors)

–U-Matrix: Valleys: clusters; Mountains: cluster borders

# Feature Planes

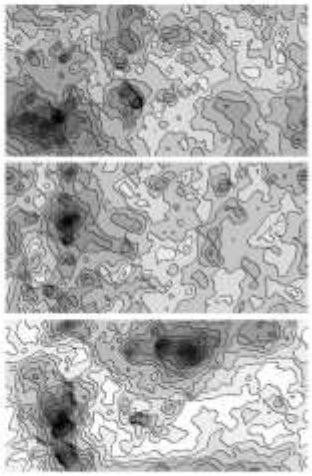


Cause of Death: Opponent

Cause of Death: Environment

Cause of Death: Falling

# Feature Planes



Number of Deaths

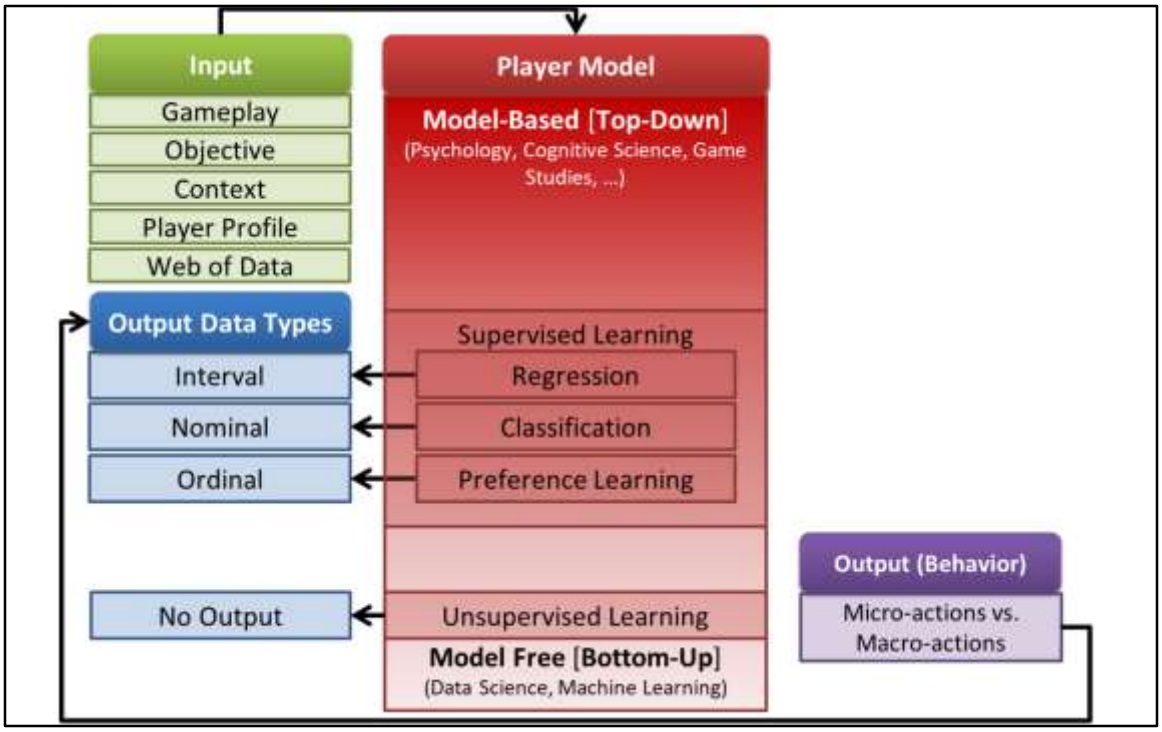
Completion Time

Help on Demand (HoD)


A video about the resulting clusters : https://www.youtube.com/watch?v=HJS-SxgXAI4

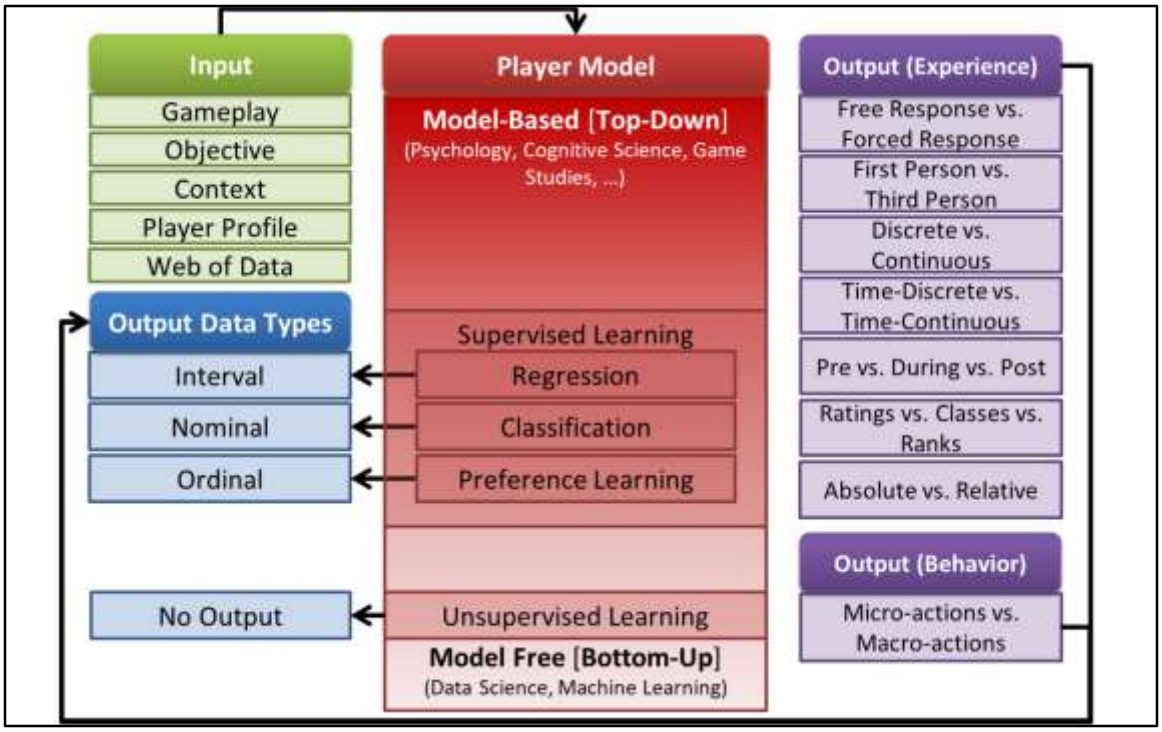
A long video about the Tomb Raider: Underworld Work: https://www.youtube.com/watch?v=A89ZDjF51Nk


Analysing Player Behaviour in Tomb Raider: Underworld Video – By “AI and Games” Youtube Channel




[see Section 5.5.1 for more details]

The task of modeling player behavior refers to the prediction or imitation of a particular behavioral state or a set of states. Note that if no target outputs are available then we are faced with either an unsupervised learning problem or a reinforcement learning problem. The output we must learn to predict (or imitate) in a supervised learning manner can be of two major types of gameplay data: either micro-actions or macro-actions (see Figure). The first machine learning problem considers the moment-to-moment game state and player action space that are available at a frequency of frame rates. For example, we can learn to imitate the moves of a player on a frame-to-frame basis by comparing the play traces of an AI agent and a human as e.g., done for Super Mario Bros (Nintendo,1985). When macro-actions are considered instead, the target output is normally an aggregated feature of player behavior over time, or a behavioral pattern. Examples of such outputs include game completion times, win rates, churn, trajectories, and game balance.




The model's output, i.e., that which we wish to model, is usually a representation of the player's state. In the following slides we explore the ways for modeling the experience of the player which implies that the output is provided predominately through manual annotation.


[see Section 5.5.2 for more details]

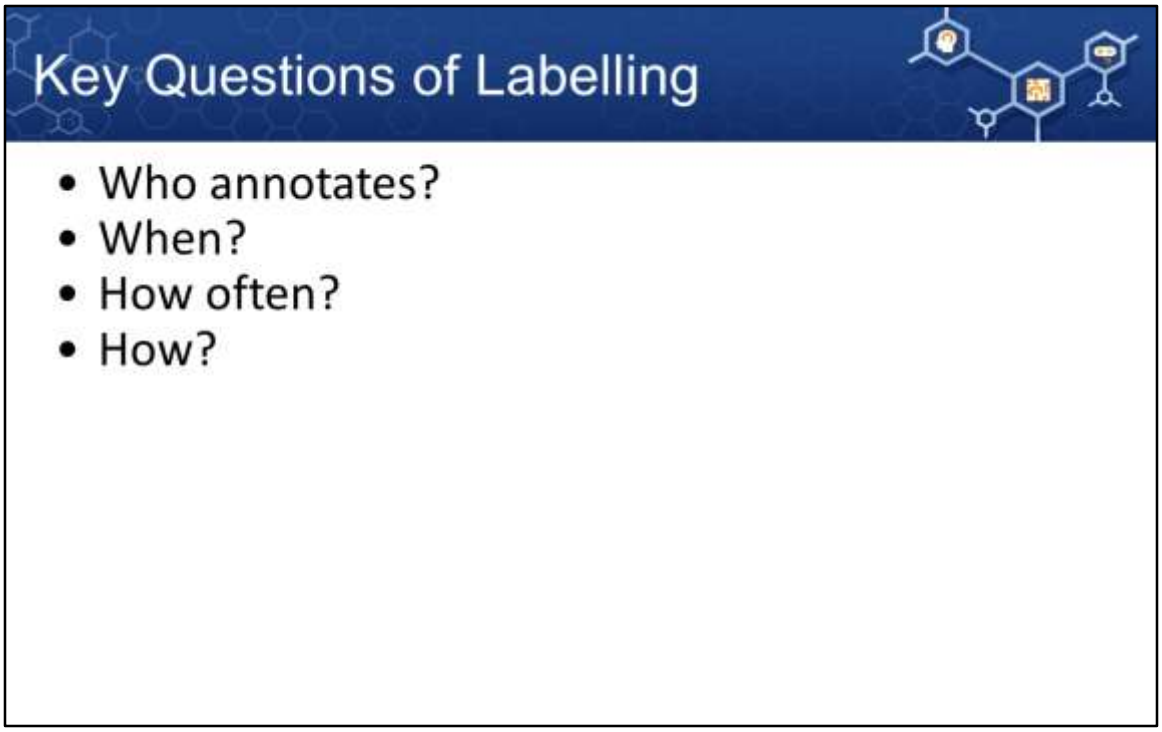
To model the experience of the player one needs to have access to labels of that experience. Those labels ideally need to be as close to the ground truth of experience as possible. The ground truth (or gold standard) in affective sciences refers to a hypothesized and unknown label, value, or function, that best characterizes and represents an affective construct or an experience. Labels are normally provided through manual annotation which is a rather laborious process. Manual annotation is however necessary given that we require some estimate of the ground truth for subjective notions such as the emotional states of the player. The accuracy of that estimation is regularly questioned as there are numerous factors contributing to a deviation between a label and the actual underlying player experience.

## Why Output (Annotation) is Key?

- Annotation is the labelling of experience
- This is ultimately the *ground truth* (golden standard) of experience
- This is the **training signal** for your computational models

The importance of labels is crucial!

- Data is available in massive amounts.
- Deep (machine) learning is doing well on its part
- The quality of labels are key for success – (e.g. see adversarial examples)
- So how to we label experience?




[see Section 5.5.2 for more details]

Manually annotating players and their gameplay is a challenge in its own right with respect to both the human annotators involved and the annotation protocol chosen. On one hand, the annotators need to be skilled enough to be able to approximate the actual experience well. On the other hand, there are still many open questions left for us to address when it comes to the annotation tools and protocols used. Such questions include:

- Who will do the labeling: the person experiencing the gameplay or others?
- Will the labeling of player experience involve states (discrete representation) or instead involve the use of intensity or experience dimensions (continuous representation)?
- When it comes to time, should it be done in real-time or offline, in discrete time periods or continuously?
- Should the annotators be asked to rate the affect in an absolute fashion or, instead, rank it in a relative fashion?

Answers to the above questions yield different data annotation protocols and, inevitably, varying degrees of data quality, validity and reliability. In the following sections we attempt to address a number of such critical questions that are usually raised in subjective annotations of player states.

# Who Annotates?

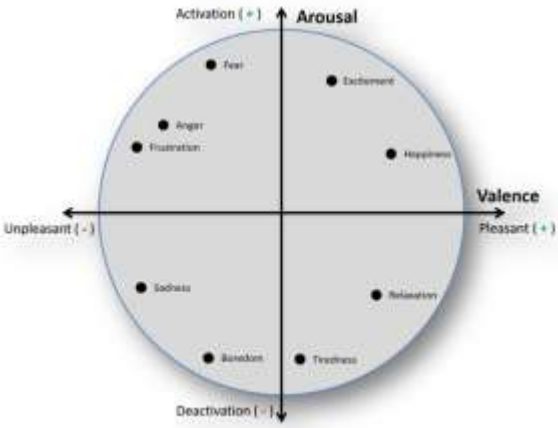
- Third Person
  - Usually a domain expert (game designer) or a psychologist
- First Person
  - The person actually experiencing the emotion/affect

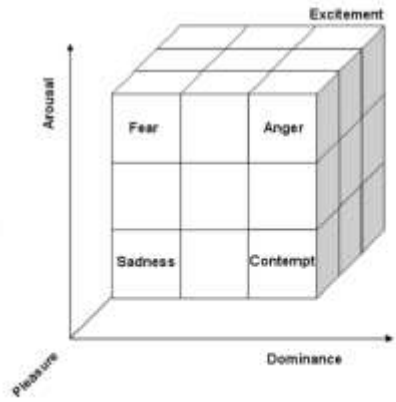
| | Third person | First person |
|---|---|---|
| + | • Expert knowledge | • Reported true experience |
| - | • Assumptions about the true emotion<br>• Reporting effects | • Self-deception<br>• Reporting effects<br>• No expert knowledge |

[see Section 5.5.2.2 for more details]

# How is Player Experience Represented?

- Discrete states (e.g. fun, engagement, frustration)
- Continuous dimensions (e.g. arousal and valence)

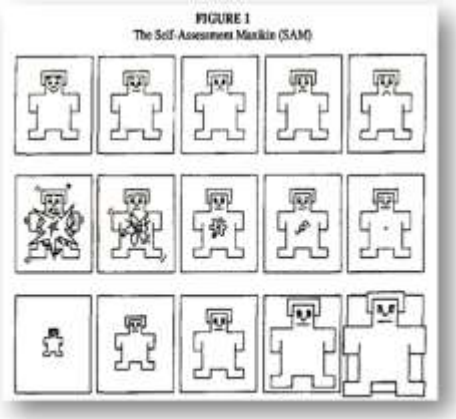



[see Section 5.5.2.3 for more details]

# How Often to Annotate?

- Time-Discrete (e.g. self-assessment manikin)
- Time-Continuous (e.g. FeelTrace, *AffectRank*)




[see Section 5.5.2.4 for more details]

## How Often to Annotate?

- Depends on
  - Application (speed of interaction: e.g. games vs. movies vs. e-learning apps)
  - Signal (e.g. physiology is slower than body movement and speech)
- No gold standard

[see Section 5.5.2.4 for more details]

# When to Annotate?

- Before play (**Pre**-Experience)
- **During** (real-time) experience
- After play (**Post**-experience)

| | Before | During | After |
|---|---|---|---|
| + | • Set the baseline of a player's state prior to playing a game<br>• Information that enriches our models<br>• Detecting the *relative* change from baseline | • Report on the spot!<br>• Real experience (better ground truth?)<br>• Limited memory effects | • Controlled<br>• Non-intrusive |
| - | • No data about experience | • Highly intrusive<br>• Distorts the experience (first person) | • Self-deception<br>• Various reporting effects |

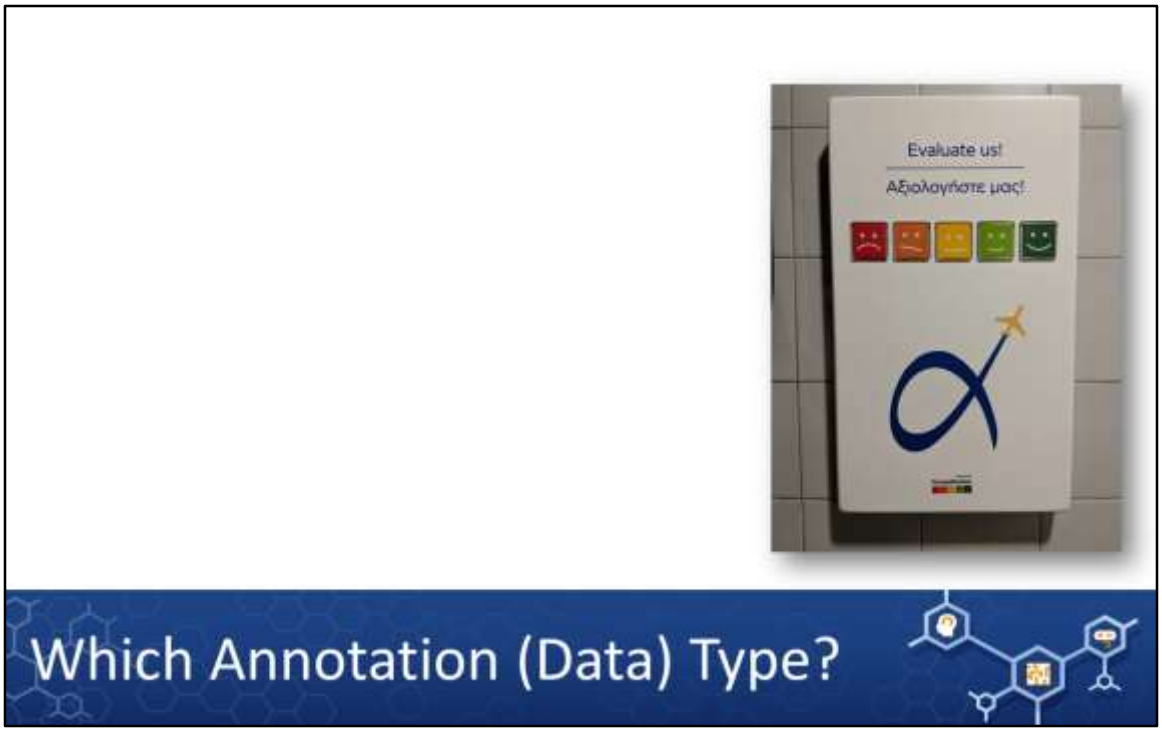
[see Section 5.5.2.5 for more details]

## A note about time and self-report!

- Self-reports are time-dependent
- Real experience vs. Post-experience
  - Few seconds → Real experience
  - Few minutes/hours → *Episodic* memory (context retrieval)
  - More → *Semantic* Memory (beliefs)

NB. The gap between our **memory of experience** and our **experience** is more prominent when we report unpleasant emotions such as anger, sadness and tension. Also: The experience felt near the end of a session (e.g. a game level or a game) affects our report – aka ***peak-end rule***.

[see Section 5.5.2.5 for more details]



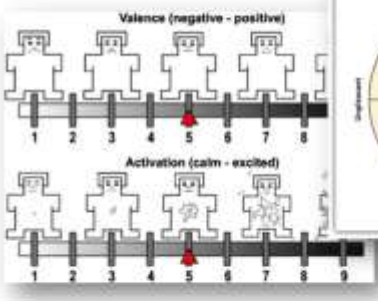

[see Section 5.5.2.6 for more details]

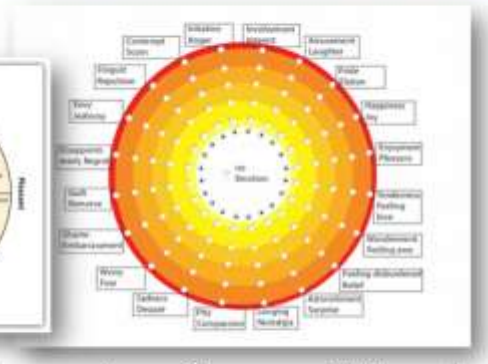
## Which Annotation (Data) Type?

- Scalar (Likert scale, arousal/valence score, SAM) – **Rating**
- Binary value or a class – **Class**
- Preference between two or more options – **Rank**

[see Section 5.5.2.6 for more details]

# Rating



X was challenging

Strongly Disagree — Strongly Agree

0 1 2 3 4 5

Examples: Geneva Wheel, SAM, Likert Scales, PAD values

# Class

Examples:

- This facial expression is **happy**! (Eckman)
- Arousal values higher than 0.6 belong to class **aroused**
- This skin conductance peak denotes **stress**

# Rank

X is **more/less** challenging / frustrating / arousing / boring / fearful / … than Y

- Requires at least two instances!
- N-Alternative Forced Choice (4-AFC is popular)

- [ ] X is **more/less** frustrating than Y
- [ ] Both are **equally** frustrating
- [ ] **Neither** is frustrating

# Which Annotation (Data) Type? Summary

| | Class | Scalar | Rank |
|---|---|---|---|
| + | Easy to analyse and process. Only one instance (to be annotated) is required. Part of subjectivity is eliminated. | Easy to analyse and process. Only one instance (to be annotated) is required. | Controls for reporting memory effects (increasing/decreasing the memory window). Eliminates subjectivity biases. It is natural to compare (key trend in marketing research). Evidence from neuroscience. |
| - | Assumptions made about classification. What is the "gold" threshold value that splits classes? | **Highly** subjective; Use of scale-bias; Culture-bias, Personality (temperament, interests)-bias; Increased between-participants effects; Logical errors: confused by ordinal scales. *Primacy* and *Recency* order effects. Memory effects. Key **fundamental issues** (see next slide). | *Primacy* and *Recency* order effects. Memory effects |


[see Section 5.5.2.7 for more details]

A **thesis**: emotions are intrinsically **ordinal** (relative)

The thesis is supported by **theoretical arguments** and **empirical evidence** across disciplines

It **reframes** the way emotions are viewed, represented and analysed computationally

## One of the first Challenges in Psychology

Mapping the intensities of responses to particular stimuli

**That is basic to AC/HCI/UX/GUR…:** we call it **labelling**

Two approaches have a long history

- The older (Fechner) was based on comparing stimuli, and finding '**just noticeable differences**'
- Much later, Stevens introduced '**magnitude estimation**' – asking people to give a number. Twenty years ago, psychologists tried a magnitude estimation approach to labelling. The data are in, and we know it doesn't work as straightforwardly as they hoped.

Gustav Fechner 1860 (top image): the father of pairwise comparisons in value (images of brightness) – the whole of psychophysics from 1860 to 1930s was based on pairwise comparisons

Stevens (bottom image): went back to assign numbers. He called that new approach: magnitude estimation

Magnitude estimation works but it is not as **robust** as the paired comparison! Part of the reason is that it is phenomenally context-dependent!

Psychologists took the convenient way; but that had its limitation

## The core finding is simple…

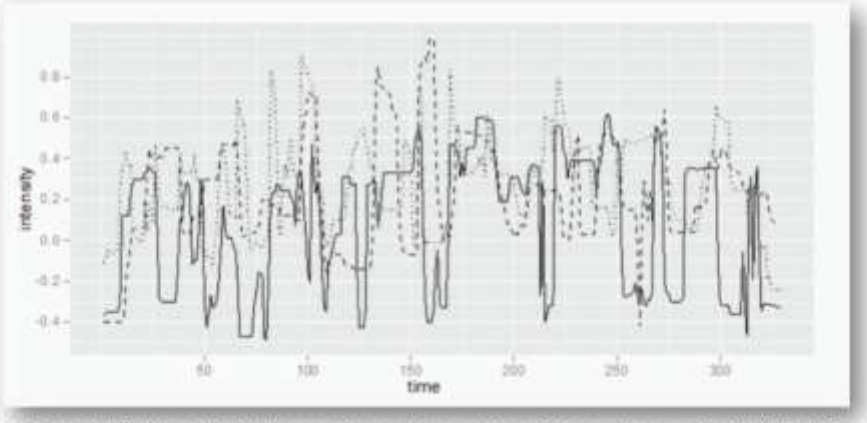
When people are presented with a *piece of data* and asked to assign a *magnitude* describing an emotional response, they tend to **disagree** quite substantially.

Douglas-Cowie et al. "**Multimodal databases of everyday emotion: Facing up to complexity**," *Ninth European Conference on Speech Communication and Technology*. 2005.

For more details please refer to the cited paper

## Or else…

….there is **seriously** no point in this….




Douglas-Cowie et al. "**Multimodal databases of everyday emotion: Facing up to complexity**," *Ninth European Conference on Speech Communication and Technology*. 2005.

For more details please refer to the cited paper


**Reason 1** Data are typically multivalued

•A scene will contain *multiple elements*, which have *different valences*, and there is no self-evident way to reduce them to a *single measure*.

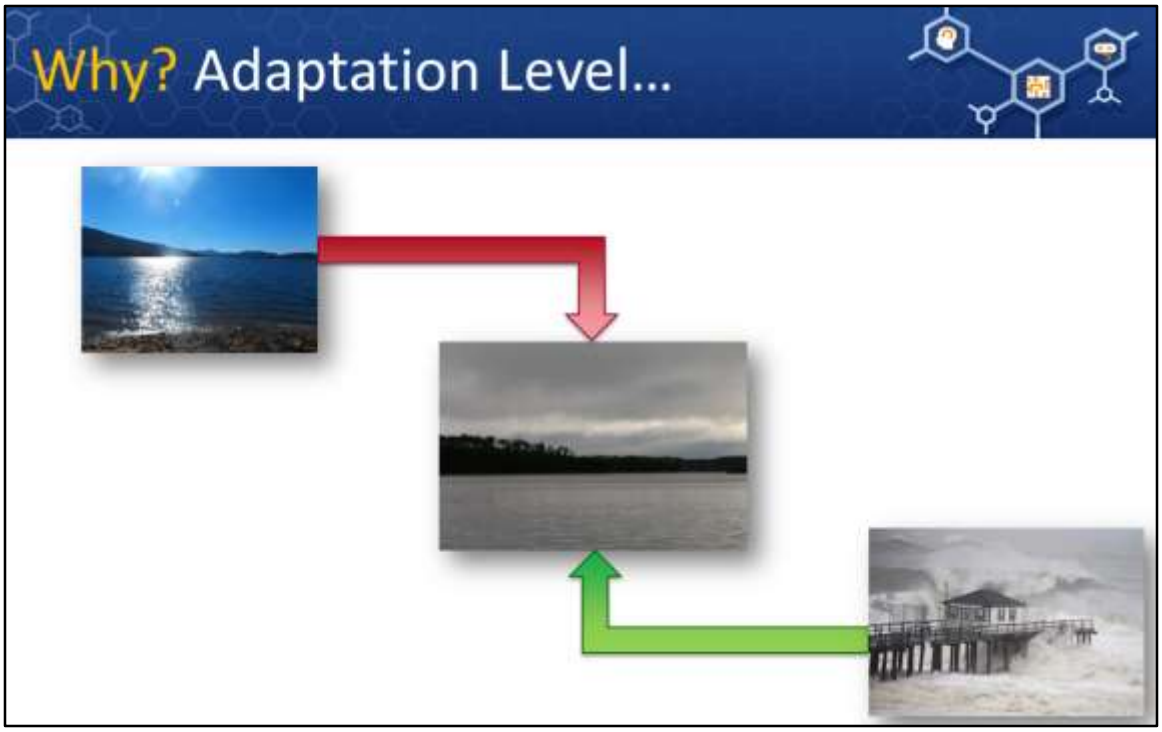
Little Red Riding hood example (you know what the outcome will be like) the mindscale has multiple valences in it and not simple way to add them up

In the background we have

– Valence for the instance that the wolf will eat the girl

+ Valence because you know that will not happen

This is typical of an emotional setting

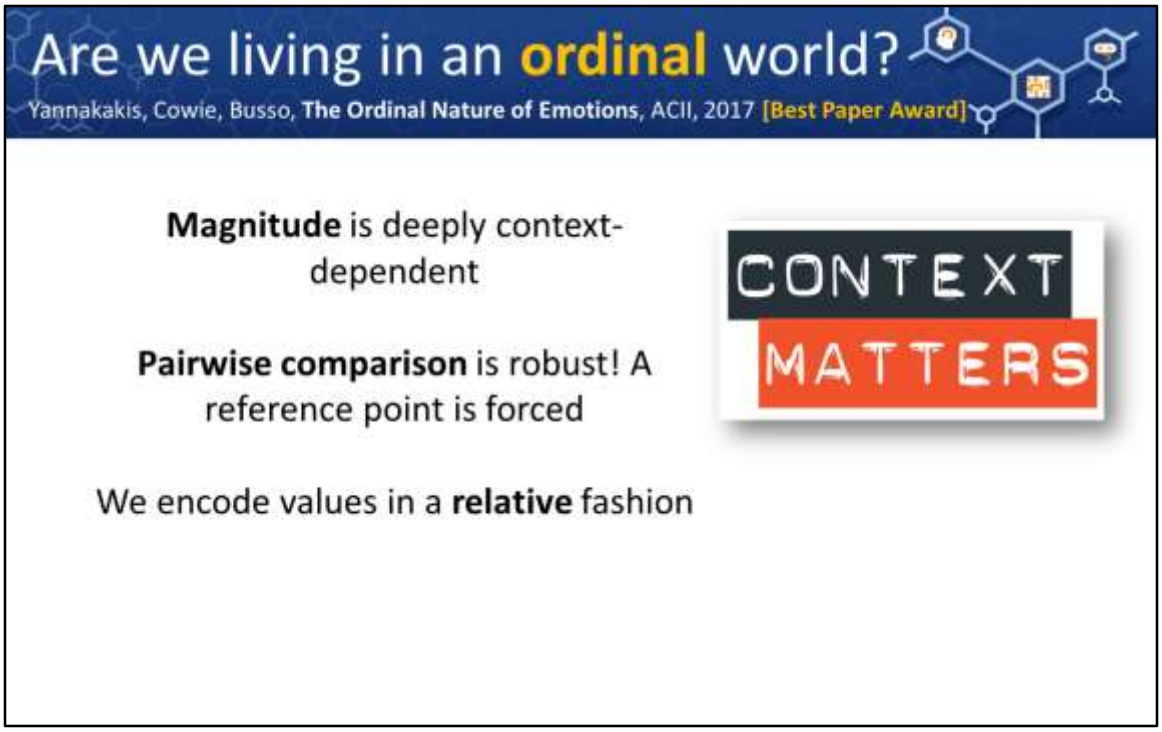



Say today is a **grey day**; what feelings will it evoke?

**–ve:** if it's ending a *sunny spell*

**+ve:** if we are coming out of a *hurricane*

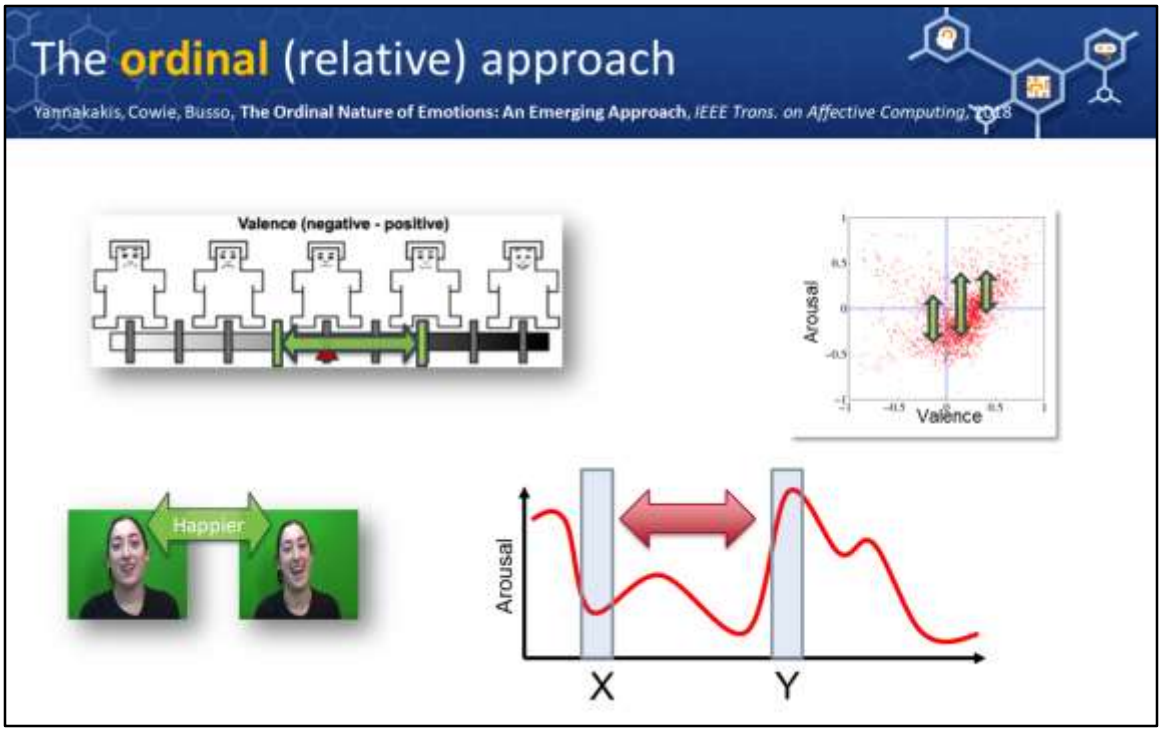
- But labelling is associating a value; So, which should we associate?
- It is terribly clear that emotional response are intensely depended on what has happened before
- The problem is labelling the stimulus
- What is robust is the relationship between




Emotions are intrinsically **ordinal** (relative)…and the benefits of representing them that way are many!

This is supported by **theoretical arguments** across disciplines and **empirical evidence** in Affective Computing

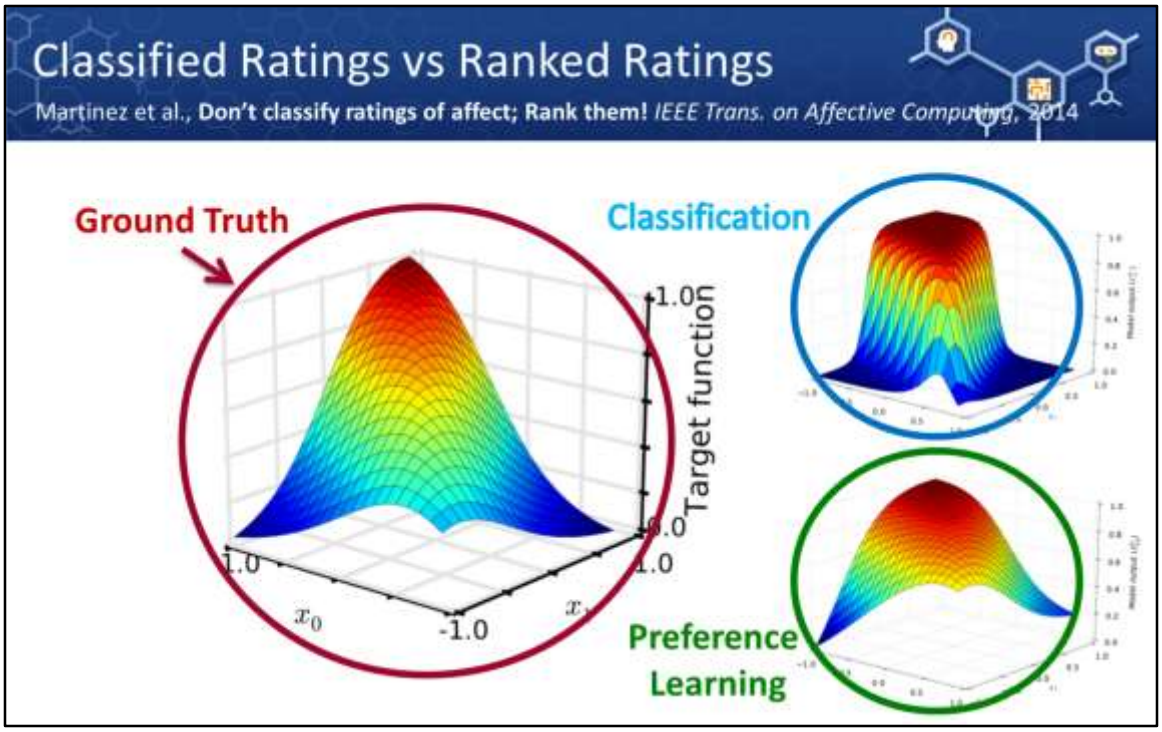
It **reframes** the way emotions are viewed, represented and analysed computationally




- What if these were options to compare? This is angrier than that…
- Their value is unknown (noisy/biased) – their relationship is robust!
- The anchor does not have to be retrieved unconsciously or intuitively; it is forced.
- The **anchor** is a **real option** one uses as a reference during the annotation.

Please refer to the cited paper for more details

# Some Evidence for the Ordinal Nature of Experience

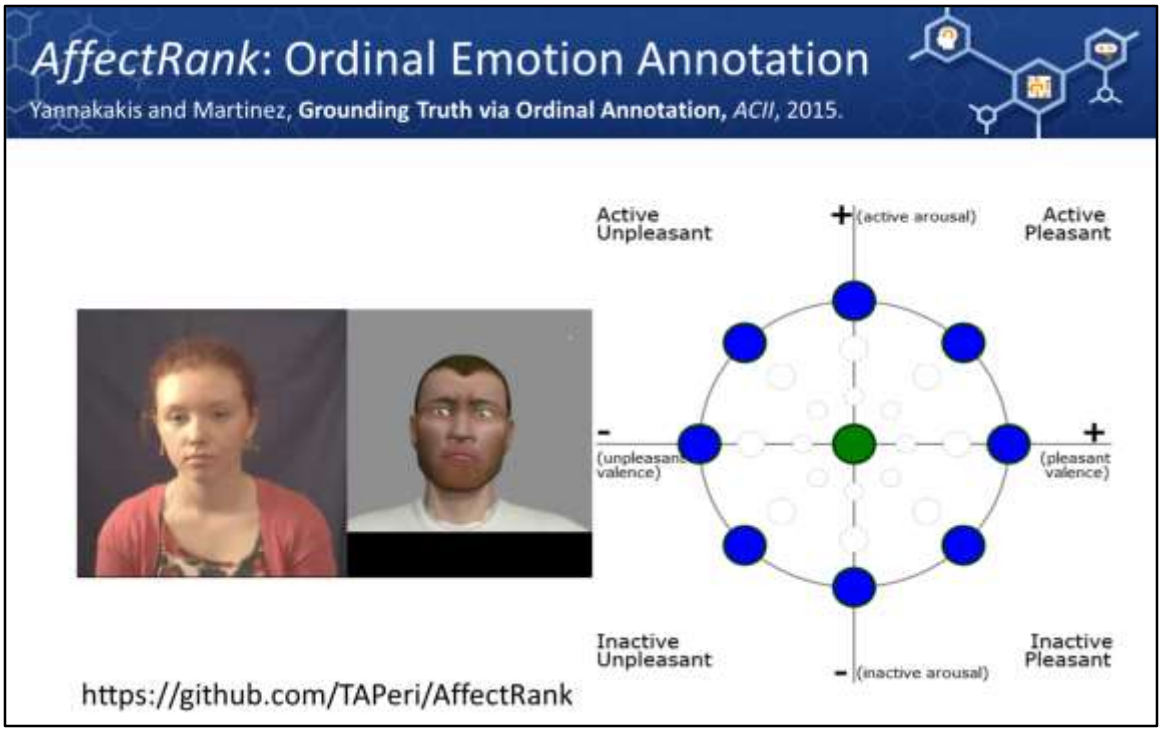



[see Section 5.6.1.4 for more details]

In the cited paper the authors compared models that are the result of converting ratings to classes (classification) versus ordinal models that are trained directly via preference learning.

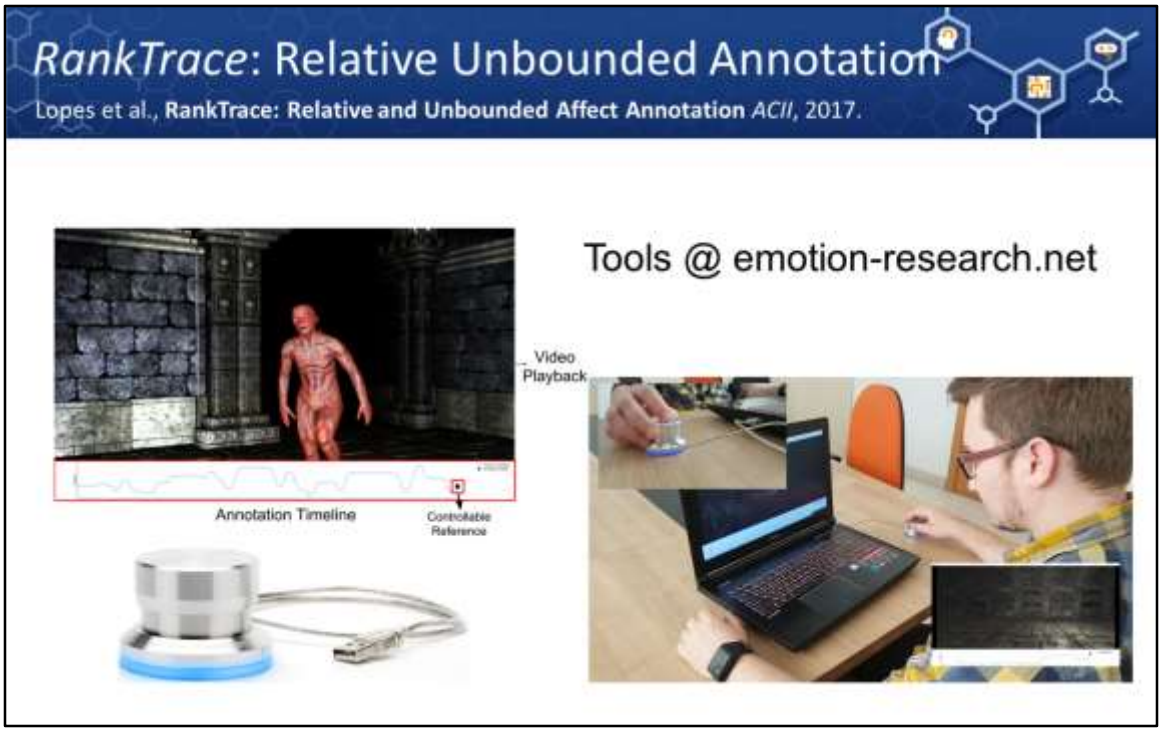
Models trained via preference learning outperform the classification models of affect in terms of cross-validation

Please refer to the cited paper for more details




- We put our hypothesis on further testing and we compared FeelTrace against its rank-based version, we named AffectRank
- The core finding is that rank-based annotation yields higher inter-rater agreement!

Please refer to the cited paper for more details



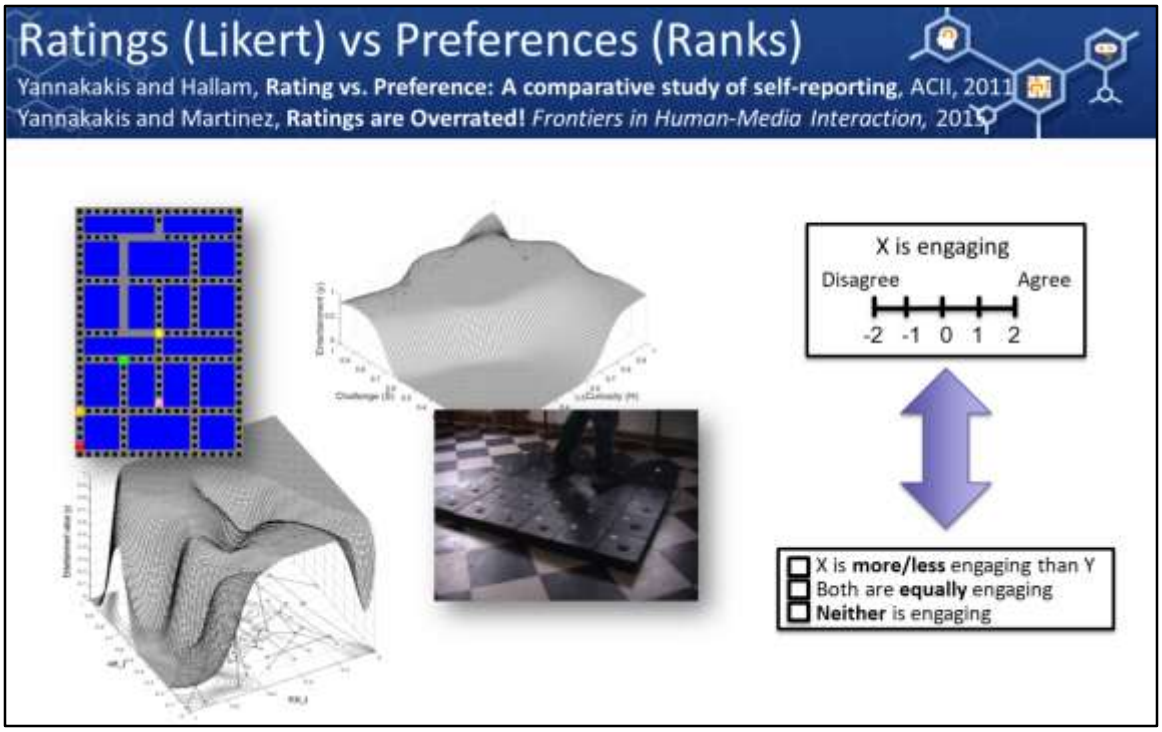

RankTrace:

- Wheel-like natural interfacing
- Unbounded annotation (natural)
- Relative processing of data

We correlated relative (average gradient) and absolute (mean) traces of tension annotations with SC phasic activation

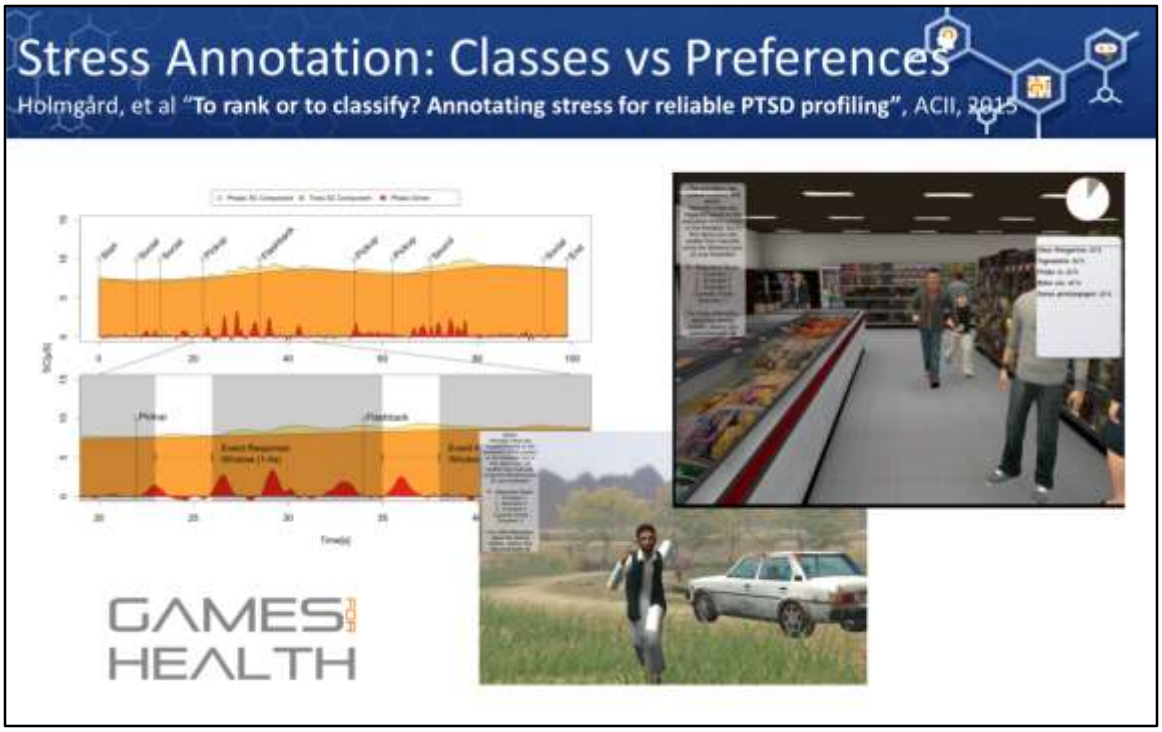
Relative annotation traces are better predictors of SC and robust -- > Across 2 windowing methods, 2 degrees of annotator memory

Please refer to the cited paper for more details




The reliability of ratings has been compared to that of ranks across a number of studies within games (prey-predator and physical interactive games). The key findings of the cited papers are as follows:

- Rater consistency (reliability) is higher when ranking protocols are used.
- The order of answering affects ratings more than ranks.

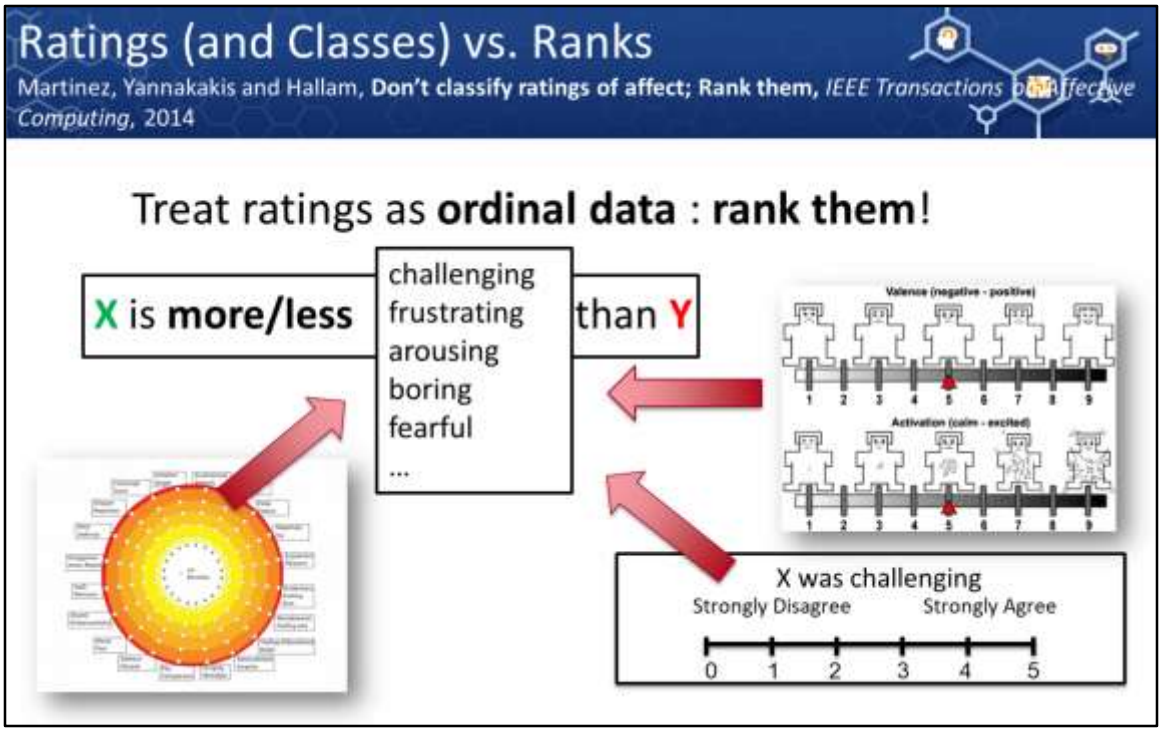



In the cited paper authors asked players of the *StartleMart* game (PTSD exposure therapy game) to annotate stress in two ways

- They reported the most stressful event in a game (class-based annotation)
- They compared stress across game events (rank-based annotation)

Findings: ordinal annotations are more accurate predictors of the phasic driver of skin conductance (i.e. a reliable indicator of underlying stress).

Please refer to the cited work for more details




Even if we had to use ratings (when ranks are not possible) – we can (and we should) convert them to ranks

Why? It generates more accurate / reliable /generalisable models!

- Regression of labels is not a good idea
- Any other conversion to a non-ordinal scale (class) generates biases

# To sum it up: don't do this…

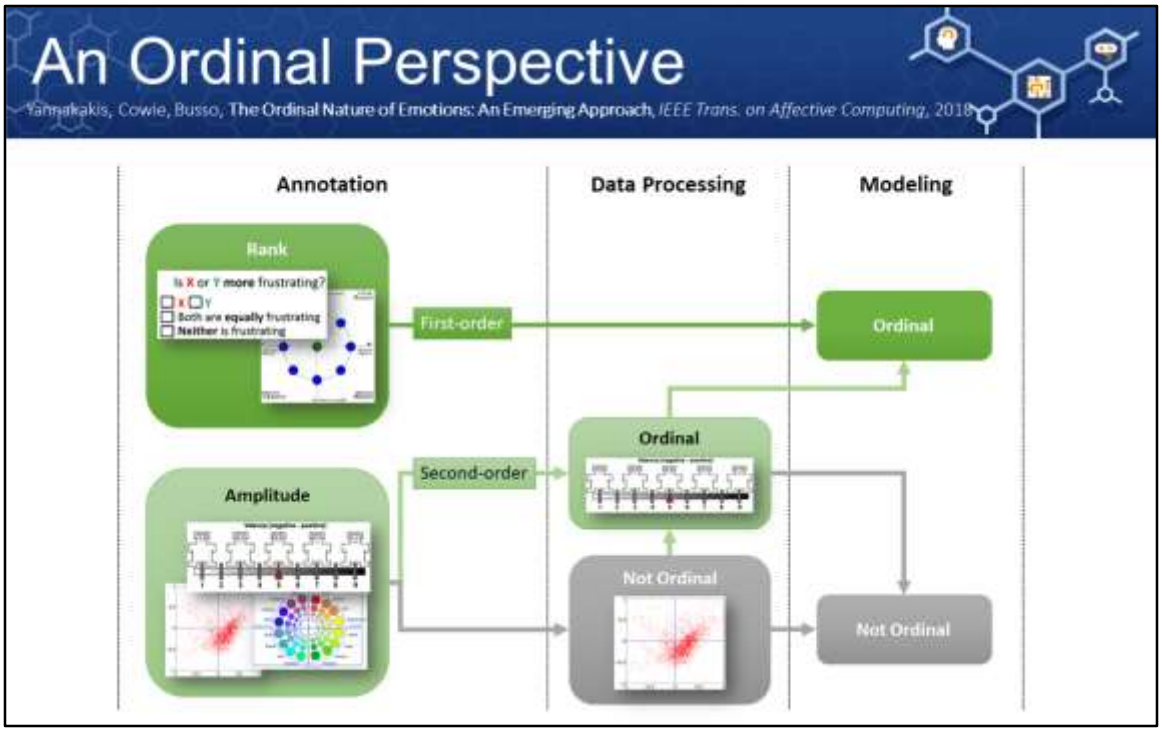
- Wasteful Info due to
  - Scale-bias
  - Personal-bias
  - Labels are **NOT** numbers
  - High inconsistency (randomness)
  - …

# Do this instead…

- ☐ I like **Julian's** class more/less than **Georgios'** class
- ☐ I like them both equally
- ☐ I like neither

You gain on:

- Reliability
- Validity
- Generality




An ordinal-centric taxonomy of approaches through the phases of annotation, data processing and data modeling. With dark green and light green color, respectively, we illustrate the *first-order* and *second-order* approaches for the ordinal analysis of annotations. With gray color we indicate data processing or modeling approaches that do not follow the ordinal perspective.

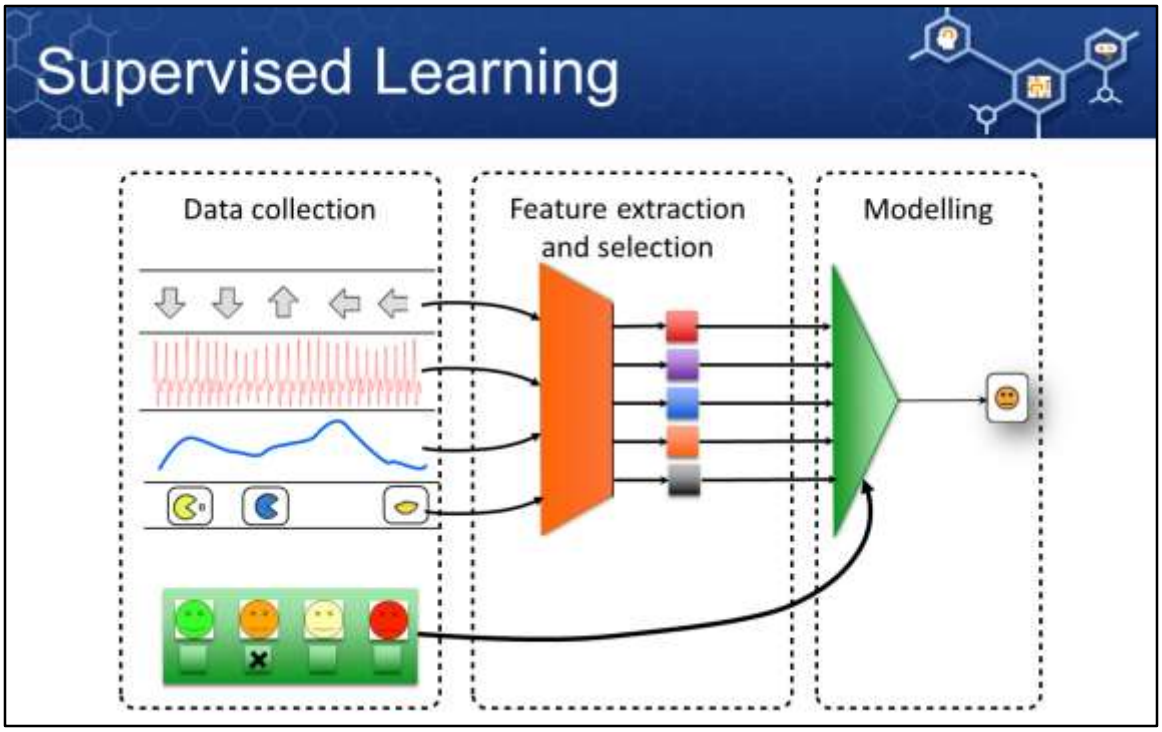
Please refer to the cited paper for more details

# Annotation – Take away messages

- 1st vs. 3rd person: depends on the application
- Try to get reports as close to the *true experience* as possible (time-wise)
- No report is ideal (they suffer from biases)
- Annotate experience as **ranks** whenever possible
- If ratings are available
  - Regression of ratings is **fundamentally wrong**
  - Do not convert them to classes – it will cost you on model performance
  - **Convert them to ranks** (treat them as ordinal scales)!

## How Can we Model Players?

[see Section 5.6 for more details]




[see section 5.6.1 for more details]

Player modeling consists of finding a function that maps a set of measurable attributes of the player to a particular player state. Following the supervised learning approach this is achieved by machine learning, or automatically adjusting, the parameters of a model to fit a dataset that contains a set of input samples, each one paired with target outputs. The input samples correspond to the list of measurable attributes (or features) while the target outputs correspond to the annotations of the player's states for each of the input samples that we are interested to learn to predict.

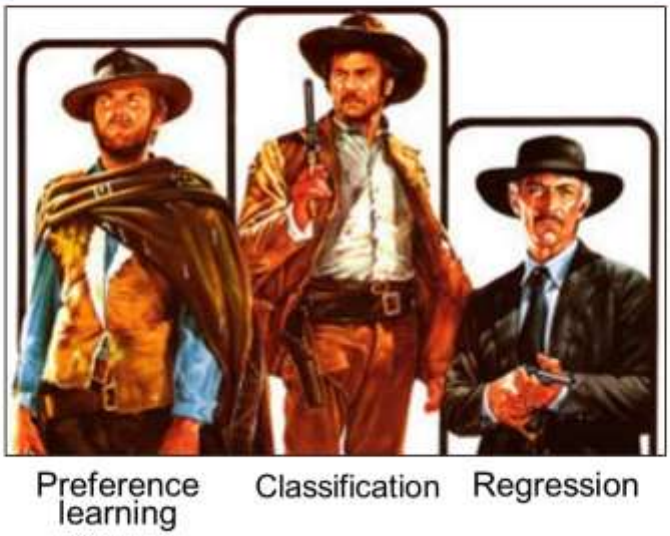
As mentioned already, the annotations may vary from behavioral characteristics, such as completion times of a level or player archetypes, to estimates of player experience, such as player frustration.

## Supervised Learning

- The output of the model is the *estimated experience*
- The **ground truth** is given by annotated experience given as
  - Nominal values (e.g. sample A is frustrated)
  - Numerical values (e.g. sample A is 0.86 frustrated)
  - Ordinal values
    - Ranks (e.g. sample A is more frustrating than sample B)
    - Ratings (e.g. sample A is 'extremely frustrating' and sample B is 'fairly frustrating'

Popular supervised learning techniques, including artificial neural networks (shallow or deep architectures), decision trees, and support vector machines, can be used in games for the analysis, the imitation and the prediction of player behavior, and the modeling of playing experience. The data type of the annotation determines the output of the model and, in turn, the type of the machine learning approach that can be applied. The three supervised learning alternatives for learning from numerical (or interval), nominal and ordinal annotations are respectively, regression, classification and preference learning

# Which Training Method?

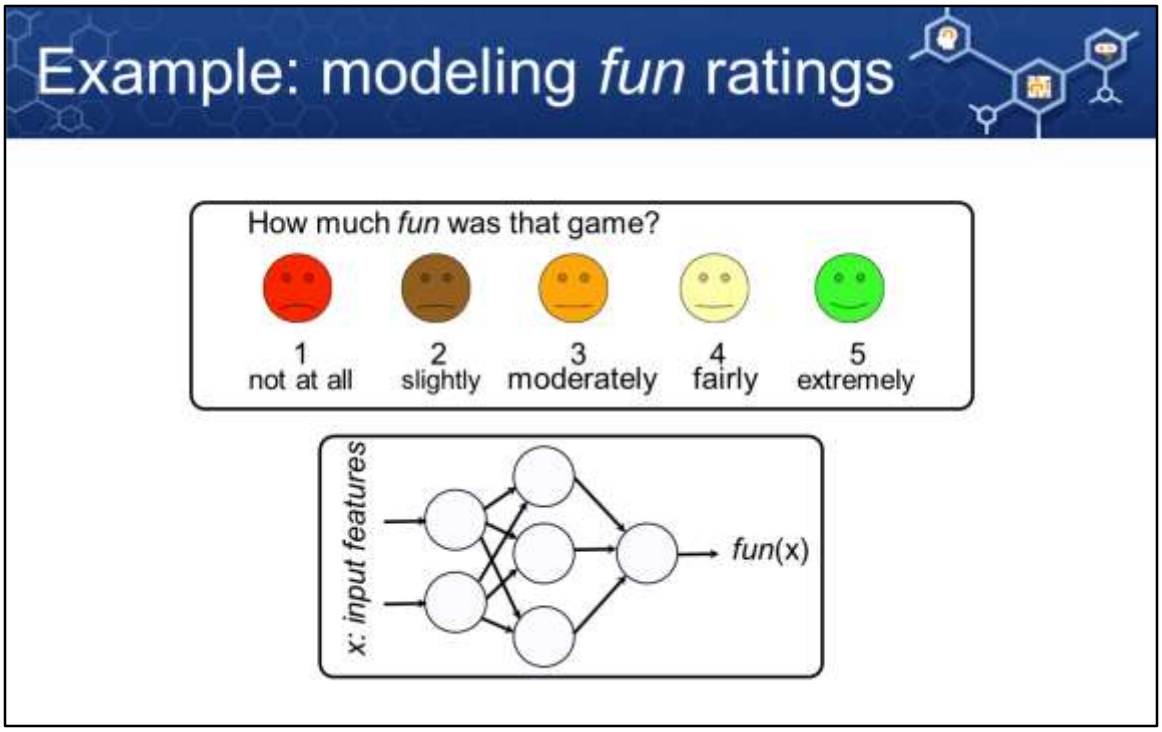



[see Section 5.6.1 for more details]

Is my target output numerical? -> Regression [Section 5.6.1.1]

Is my target output nominal? -> Classification [Section 5.6.1.2]

Is my target output ordinal? -> Preference learning [Section 5.6.1.3]




Here is an example of modelling player experience (given as *fun* ratings) based on a number of player characteristics (e.g. gameplay data, physiology etc..) using an artificial neural network

# Reminder: Backpropagation

(1) Initialise to random weights

(2) For each training pattern p:

(a) Present input pattern $\vec{x}^{(p)}$

(b) Compute output(s) using forward mode

(c) Compute output error $E^{(p)}$

(d) Compute error derivatives $\frac{\partial E}{\partial w_{jk}}$

(e) Update weights by $-\eta \frac{\partial E}{\partial w_{ij}}$

(3) Is error small?

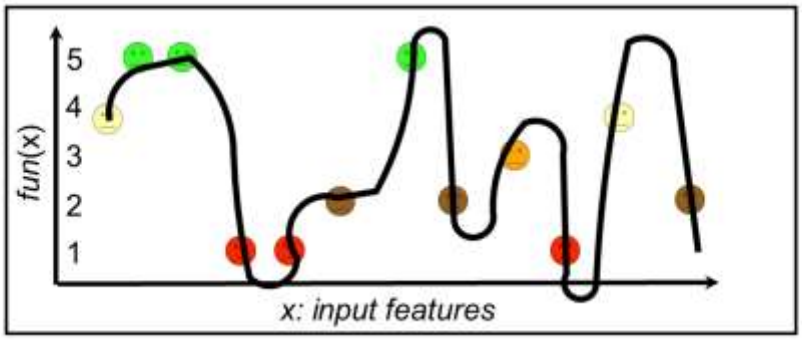
– Yes: then STOP

– No: loop to step (2)

A reminder slide: the basic steps of standard backpropagation – see further details in Chapter 2

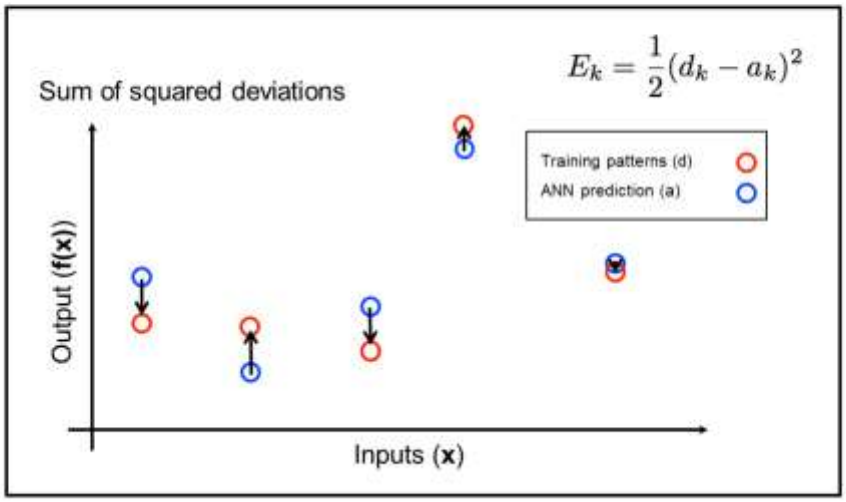
# The bad: Regression

- Remember: ratings are **NOT** numbers!
  - Not everyone uses scales in the same way
  - Items in the scale are not equidistant



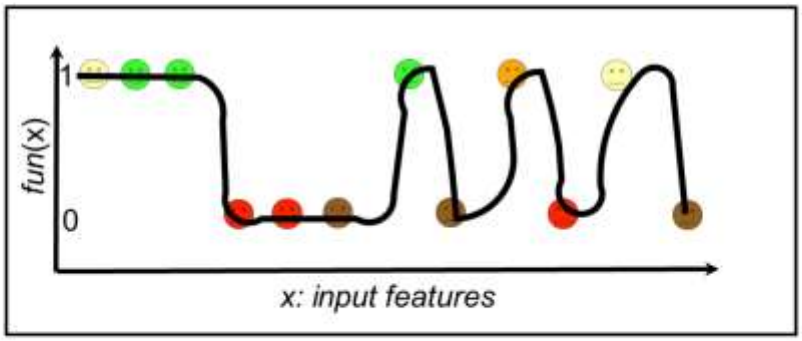

[see Section 5.6.1.1 for more details]

# Regression with Backpropagation



# The ugly: Classification

- Converting ratings into classes eliminates a lot of information and it can introduce biases



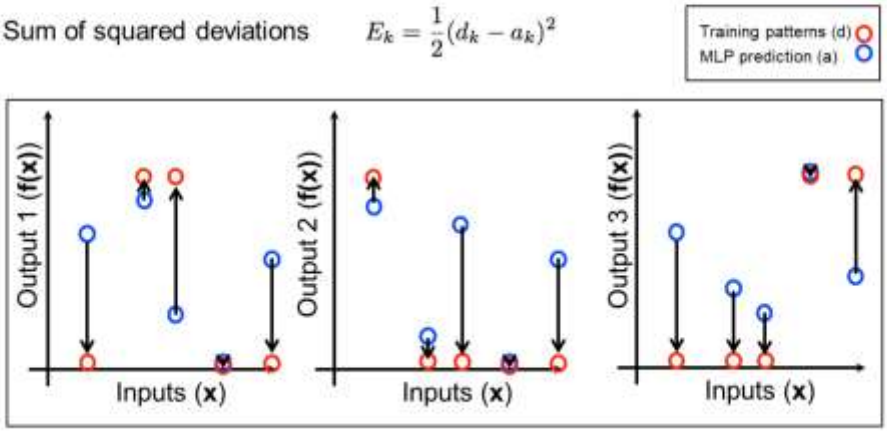

H. P. Martinez, G. N. Yannakakis and J. Hallam, "**Don't Classify Ratings of Affect; Rank them!**," *IEEE Transactions on Affective Computing*, 2014

[see Section 5.6.1.2 for more details]

# Classification with Backpropagation

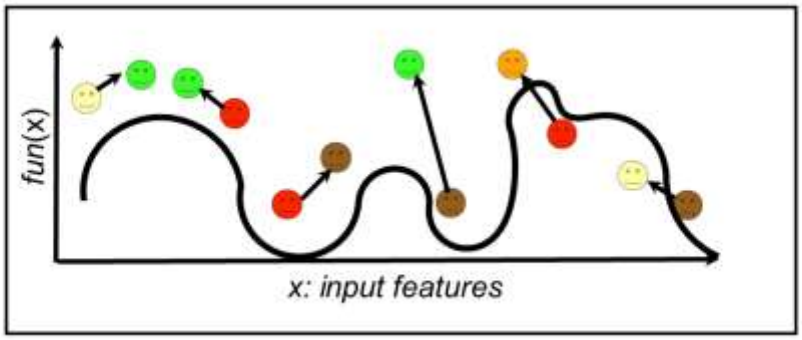
- Same as regression but with one output per class

Sum of squared deviations $E_k = \frac{1}{2}(d_k - a_k)^2$



# The good: Preference Learning

- Learn only the ordinal relations
- Valid whenever the annotator is consistent on her use of the scale

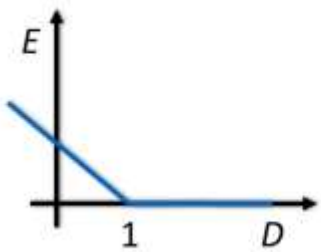


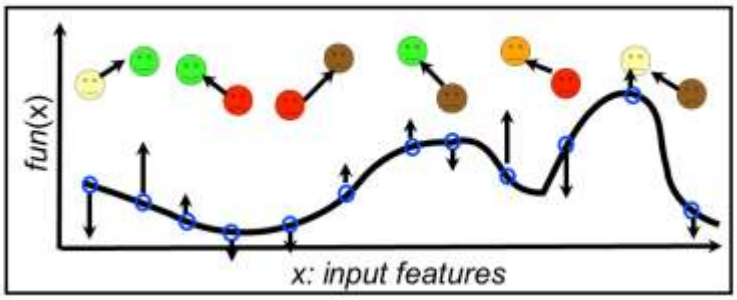

[see Section 5.6.1.3 for more details]

# (Deep) Preference Learning

- Error function maximizes the distance between the output for the preferred sample ($d^A$) and the output for the non preferred sample ($d^B$)

$$E = max(0, 1 - (d^A - d^B)) \qquad \frac{\partial E}{\partial w_{ij}} = \begin{cases} -\frac{\partial d^A}{\partial w_{ij}} + \frac{\partial d^B}{\partial w_{ij}} & \text{, if } d^A - d^B < 1 \\ 0 & \text{, otherwise} \end{cases}$$



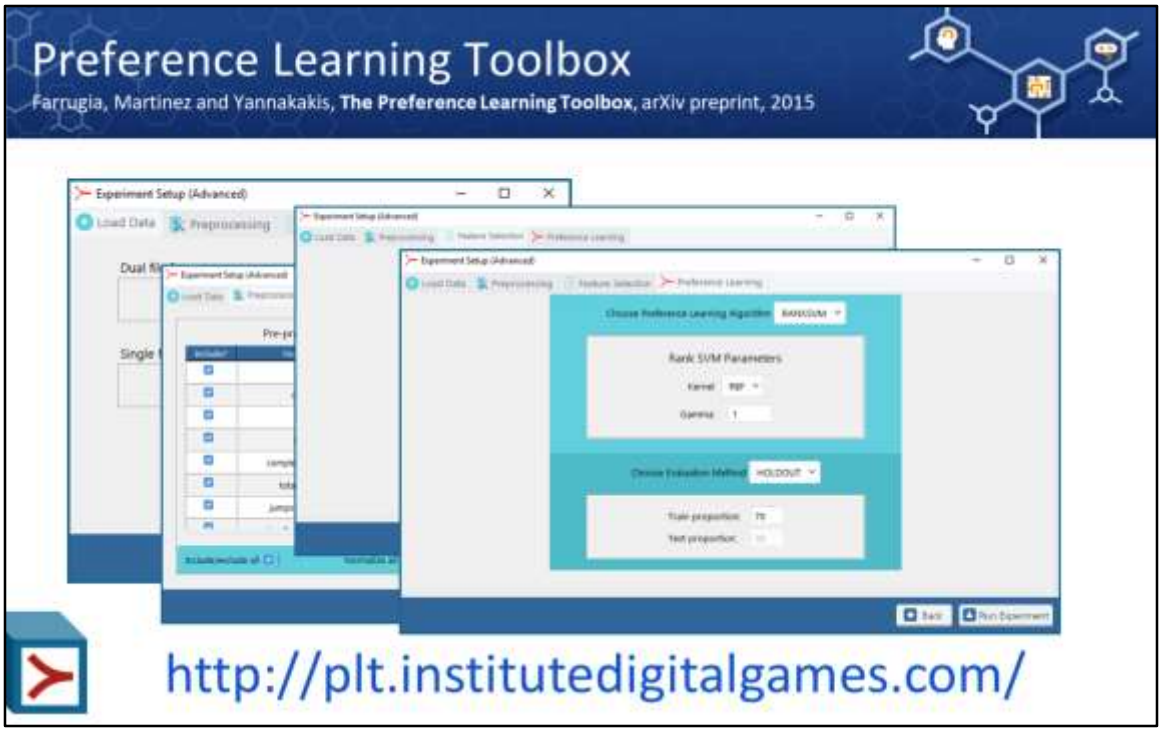

This function decreases linearly as the difference between the predicted value for preferred and non-preferred samples

increases. The function becomes zero if this difference is greater than 1, i.e., there is **enough margin** to separate the

preferred from the non-preferred sample

# (Deep) Preference Learning Beyond BP

- The concept of **learning from pairs of preferences** can be implemented in most supervised learning methods by adapting the error/fitness function
  - Neuroevolution
    - Fitness that rewards match of pairs
  - Rank-based ANN (RankNet)
  - SVMs (RankSVM)
  - Decision Trees
  - ...




To facilitate the use of proper machine learning methods on preference learning problems, a number of such preference learning methods as well as data preprocessing and feature selection algorithms have been made available as part of the Preference Learning Toolbox (PLT). PLT is an open-access, user-friendly and accessible toolkit built and constantly updated for the purpose of easing the processing of (and promoting the use of) ranks.

# Preference Learning Examples

# Player Experience Modeling in Super Mario

- 327 subjects (1308 games)
- Input: Playing Behavior and Content Features
- Output: Engagement, Frustration, Challenge self-reported ranks (pairwise) of short games
- ANN trained via **Neuroevolutionary Preference Learning**
- Player experience model accuracy: **73-92**%

## Super Mario Bros Example:
## The Annotated Experience (ANN output)

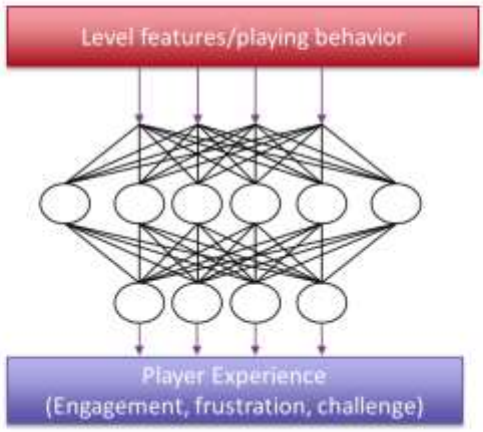
Shaker, Asteriadis, Yannakakis and Karpouzis, **Fusing Visual and Behavioral Cues for Modelling User Experience in Games,** *IEEE Trans. on Systems, Man and Cybernetics* (B), 2013

- Three Player experience states modelled:
  - *Engagement*, *Frustration*, *Challenge*
- Player Experience is self-reported (post-experience) via a 4-alternative forced choice questionnaire:

☐ **Game A** is more/less **engaging** than **Game B**
☐ Both are equally **engaging**
☐ Neither is **engaging**

# Super Mario Bros Example:
## The Modeling Approach

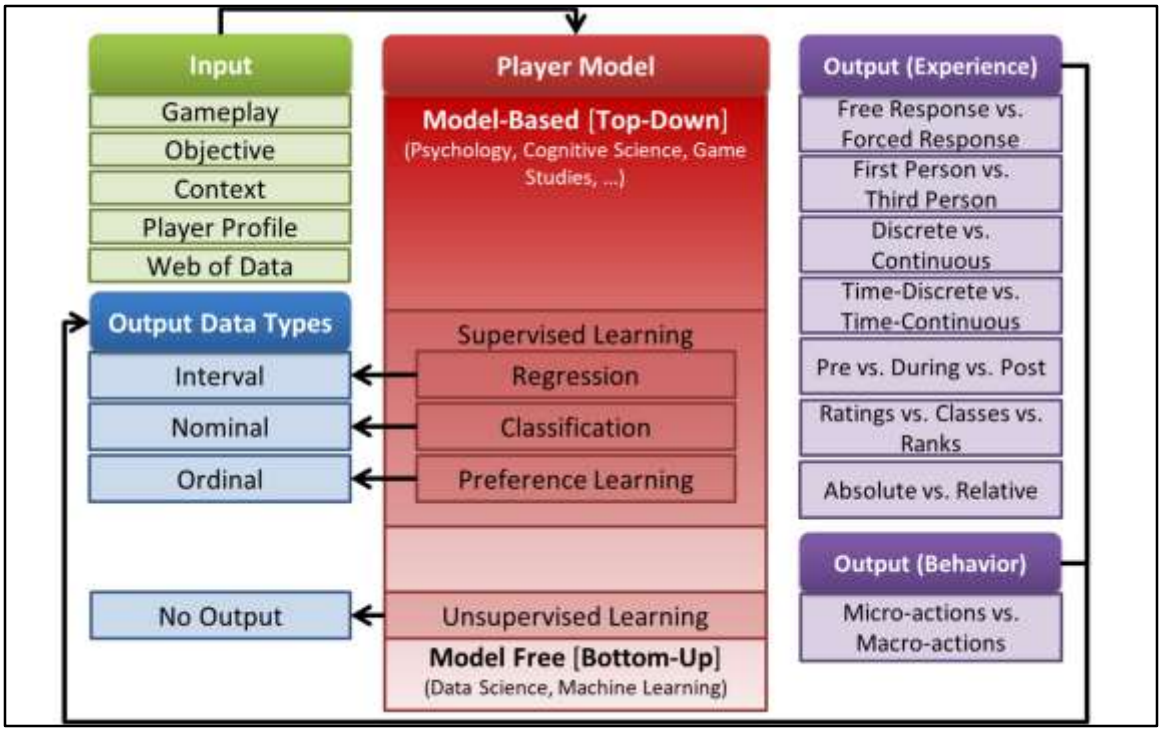
Shaker, Asteriadis, Yannakakis and Karpouzis, **Fusing Visual and Behavioral Cues for Modelling User Experience in Games,** *IEEE Trans. on Systems, Man and Cybernetics* (B), 2013

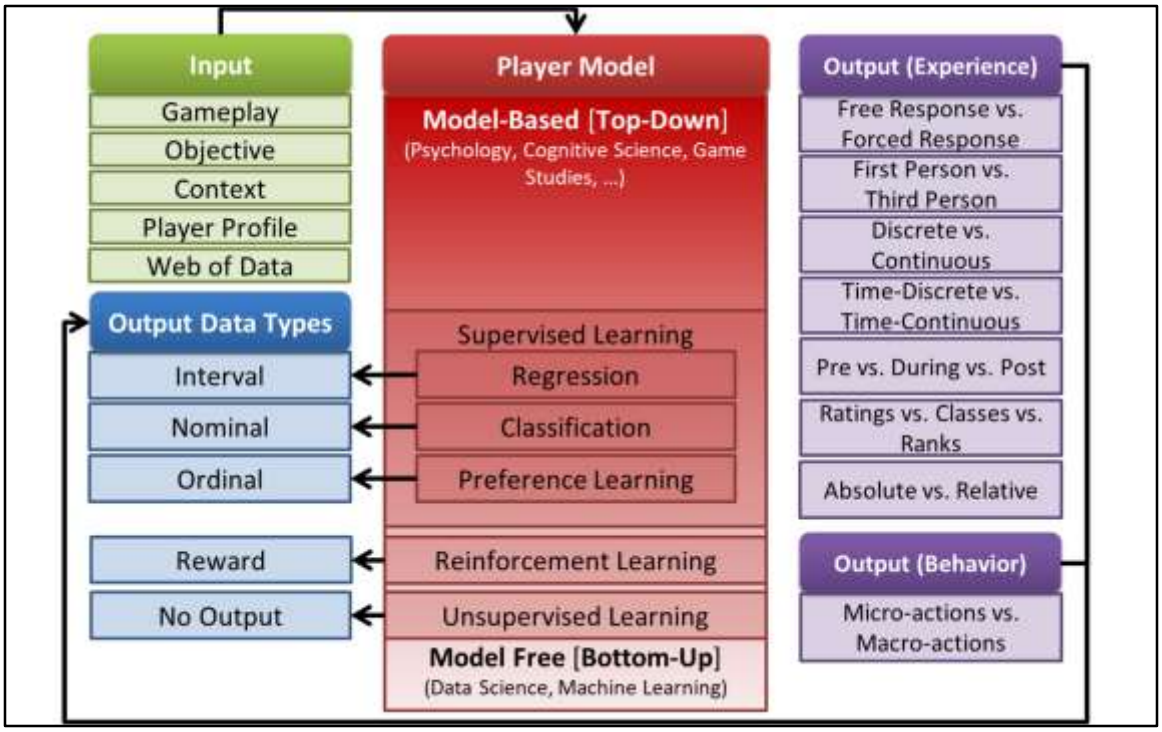


# Platformer Experience Dataset

K. Karpouzis, G. Yannakakis, N Shaker, S. Asteriadis. **The Platformer Experience Dataset**, Sixth Affective Computing and Intelligent Interaction (ACII) Conference, 2015.

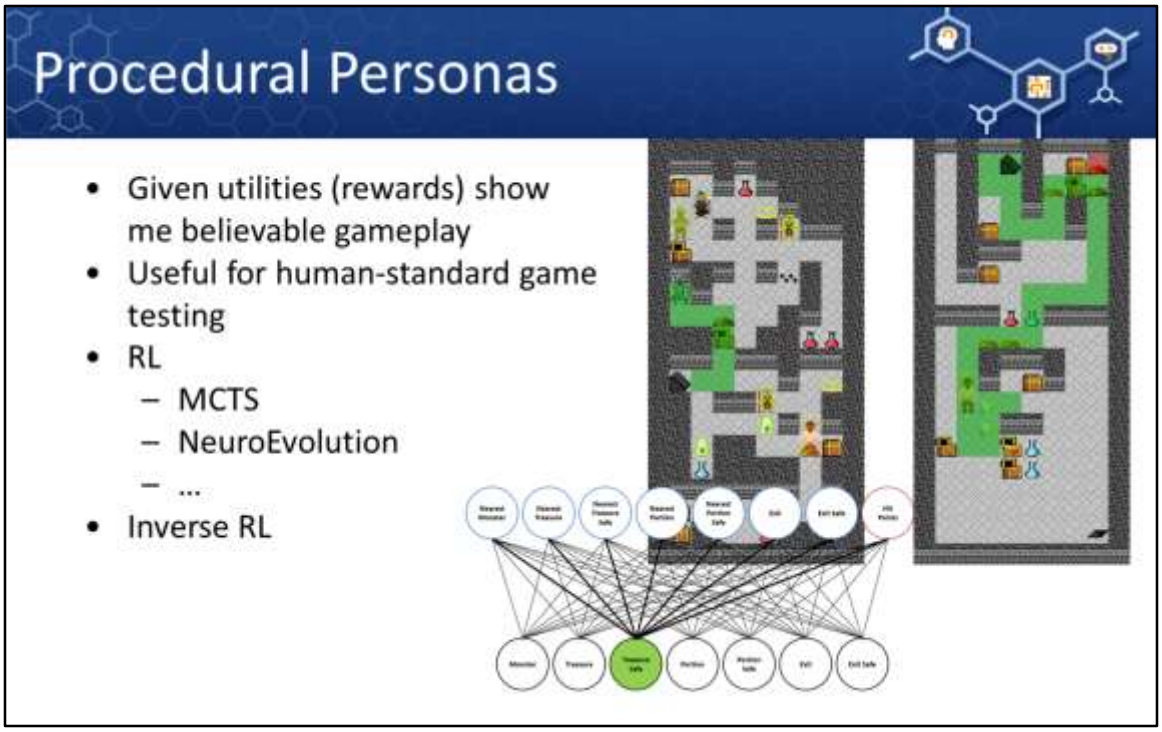
http://ped.institutedigitalgames.com/


Input
Gameplay
Objective
Context
Player Profile
Web of Data
Output Data Types
Interval
Nominal
Ordinal
No Output
Player Model
Model-Based [Top-Down]
(Psychology, Cognitive Science, Game Studies, ...)
Supervised Learning
Regression
Classification
Preference Learning
Unsupervised Learning
Model Free [Bottom-Up]
(Data Science, Machine Learning)
Output (Experience)
Free Response vs. Forced Response
First Person vs. Third Person
Discrete vs. Continuous
Time-Discrete vs. Time-Continuous
Pre vs. During vs. Post
Ratings vs. Classes vs. Ranks
Absolute vs. Relative
Output (Behavior)
Micro-actions vs. Macro-actions



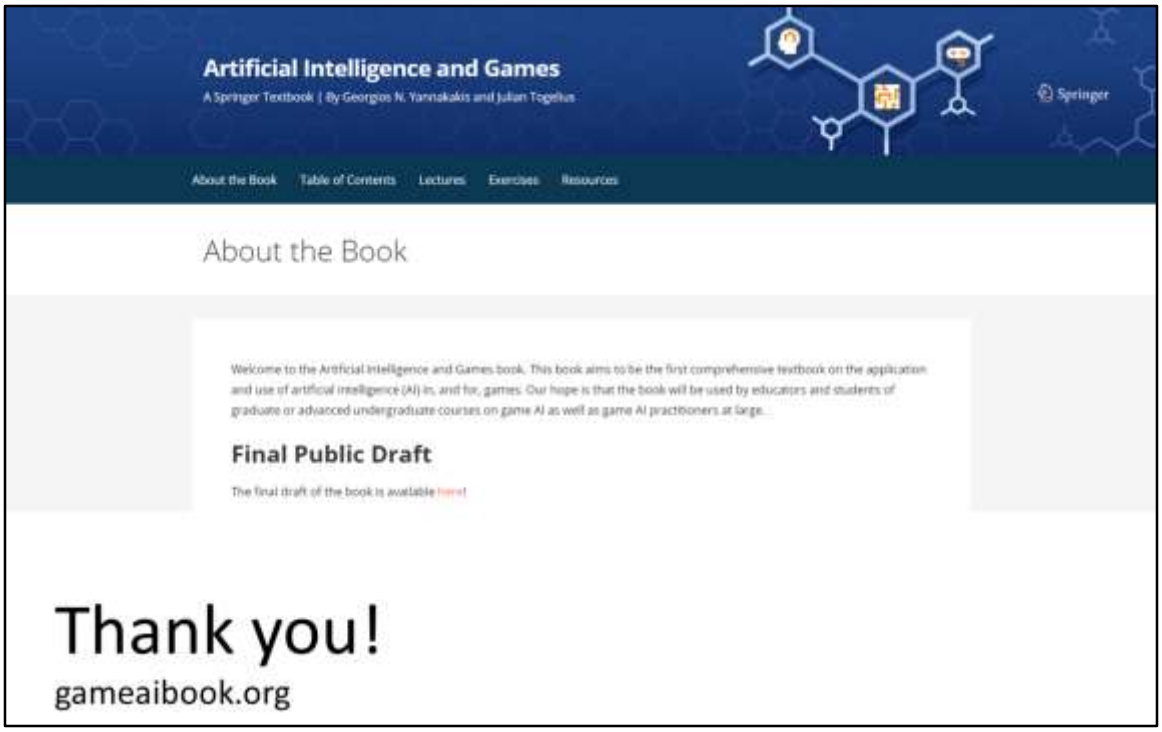

[see Section 5.6.2 for more details]

While it is possible to use reinforcement learning to model aspects of users during their interaction the RL approach for modeling players has been tried mostly in comparatively simplistic and abstract games and has not seen much application in computer games. The key motivation for the use of RL for modeling players is that it can capture the relative valuation of game states as encoded internally by humans during play. At first glance, player modeling with RL may seem to be an application of RL for game playing, and we discuss this in Chapter 3 as part of the play for experience aim. In this section, we instead discuss this approach from the perspective that a policy learned via RL can capture internal player states with no corresponding absolute target values such as decision making, learnability, cognitive skills or emotive patterns. Further, those policies can be trained on player data such as play traces. The derived player model depicts psychometrically-valid, abstract simulations of a human player's internal cognitive or affective processes. The model can be used directly to interpret human play, or indirectly, it can be featured in AI agents which can be used as playtesting bots during the game design process, as baselines for adapting agents to mimic classes of human players, or as believable, human-like opponents.




[see Section 5.7.1.3 for more details]

**Procedural personas** are generative models of player behavior, meaning that they can replicate in-game behavior and be used for playing games in the same role as players; additionally, procedural personas are meant to represent archetypical players rather than individual players. A procedural persona can be defined as the parameters of a utility vector that describe the preferences of a player. For example, a player might allocate different weight to finishing a game fast, exploring dialog options, getting a high score, etc.; these preferences can be numerically encoded in a utility vector where each parameter corresponds to the persona's interest in a particular activity or outcome. Once these utilities are defined, reinforcement learning via TD learning or neuroevolution can be used to find a policy that reflects these utilities, or a tree search algorithm such as MCTS can be used with these utilities as evaluation functions. Approaches similar to the procedural persona concept have also been used for modeling the learning process of the player in educational games via reinforcement learning




# Thank you!

gameaibook.org