# Emergence and game design

Aalto University’s Game Analysis course

Prof. Perttu Hämäläinen 2024

# Contents

- Intro to emergence in games
- Mechanics, Dynamics & Aesthetics (MDA)
- Some additional resources

# Contents

- **Intro to emergence in games**
- Mechanics, Dynamics & Aesthetics (MDA)
- Some additional resources

# What is emergence?

- Complex behavior arising from relatively simple rules or interactions
- Typically results from the interaction of multiple elements

# Interacting physics objects

# Interacting NPCs


## 12 Emergent Narrative

### Past, Present and Future of An Interactive Storytelling Approach

*Sandy Louchart, John Truesdale, Neil Suttie and Ruth Aylett*

### 1. RESEARCH BACKGROUND

The Emergent Narrative (EN) concept is a developing Interactive Storytelling (IS) approach, which aims to investigate the implications of an interactive user within the context of a narrative environment. The EN approach has thus far followed both a theoretical and practical evolution; and in this chapter we aim to discuss the context in which it initially took shape, share our vision for what it could mean for the community at large and discuss the current ongoing developments: specifically Distributed Drama Management (DDM), Relevant Context Modelling (RCM) and Intelligent Narrative Feedback (INF) for authoring EN.

The EN approach was first conceptualised at the turn of the century in the wake of booming interest in Interactive Storytelling (IS) from the Artificial Intelligence (AI) research community (Aylett, 1999). In particular, Janet Murray had previously just published her seminal *Hamlet on the Holodeck* work (Murray, 1997) and Michael Mateas and Andrew Stern were in the process of developing the ground-breaking work *Façade* (Mateas and Stern, 2003) on the back of earlier research into Interactive Drama (ID) in the Oz project (Mateas, 1997). With IS research still in its infancy, the notion of the Narrative Paradox (Louchart and Aylett, 2003) was rightly identified as a core though complex priority issue. How could IS authors and a system's artificial intelligence (AI) reconcile the demands of a carefully structured story experience with the necessary freedoms (movement, decisions) one would expect to grant an interactive user?

Most approaches, particularity at the time, tended to favour some level of top-down (story to interaction) interventions or mechanisms through which a story's plot could be modified or articulated and react accordingly to a user's inputs. Whilst there existed clear benefits in such approaches (Riedl, 2004; Szilas, 2003; Cavazza et al., 2002), Sandy Louchart and Ruth Aylett put forward the position that a top-down approach would always lack flexibility and prove difficult to scale, particularly when ensuring the consistency of interactive non-player characters (NPCs) within a narrative environment (Aylett and Louchart, 2003).

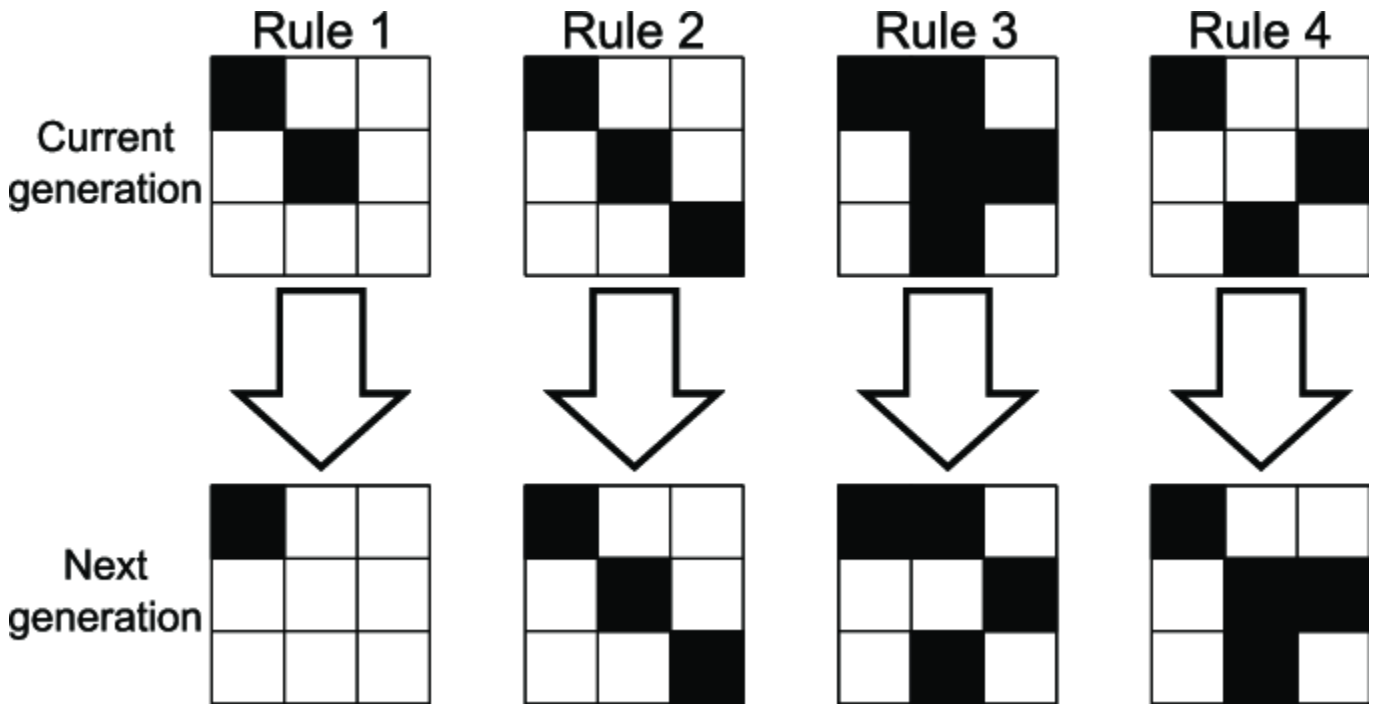
https://pure.hva.nl/ws/files/24131624/IDN_BookafterfinalrevisionsKostkova.pdf#page=202

# Emergence and games

- Pros: Replayability, (almost) endless novelty and variety
- Cons: Hard to test extensively and predict the game experience, prone to glitches and and exploits

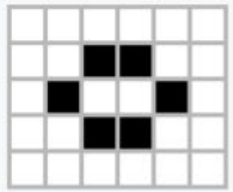
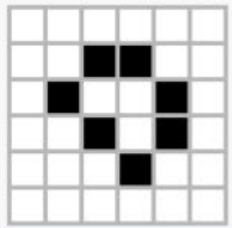
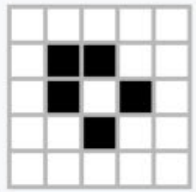
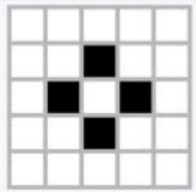
# Conway's Game of Life: 4 rules

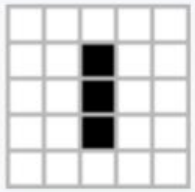
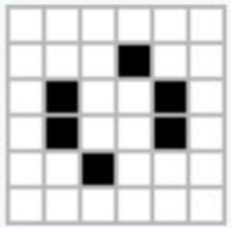
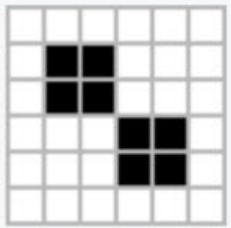
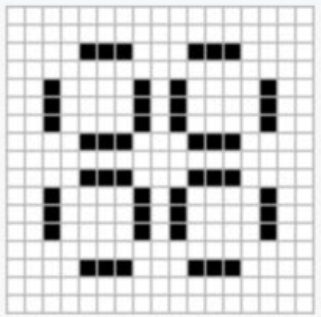
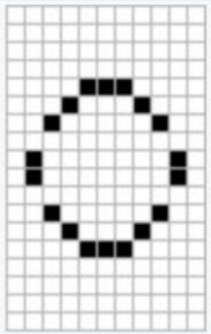
1. A cell with < 2 neighbours dies, as if by underpopulation.
2. A cell with 2 or 3 neighbours lives
3. A cell with > 3 neighbours dies, as if by overpopulation.
4. A dead cell with exactly 3 neighbours becomes alive, as if by reproduction.

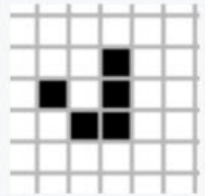
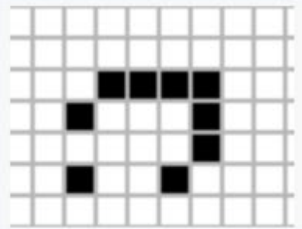
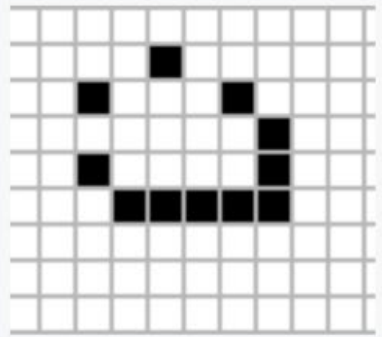
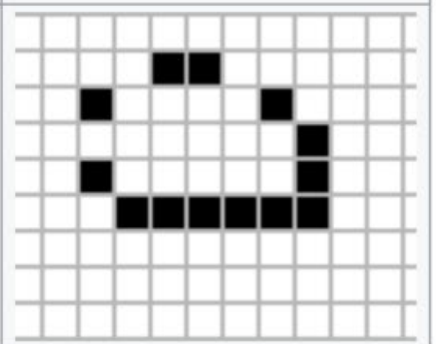



Effect of the rules on a cell (middle) based on the neighbors

https://playgameoflife.com/

| Still lifes | |
|---|---|
| Block |  |
| Bee-hive |  |
| Loaf |  |
| Boat |  |
| Tub |  |

| Oscillators | |
|---|---|
| Blinker (period 2) |  |
| Toad (period 2) |  |
| Beacon (period 2) |  |
| Pulsar (period 3) |  |
| Penta-decathlon (period 15) |  |

| Spaceships | |
|---|---|
| Glider |  |
| Light-weight spaceship (LWSS) |  |
| Middle-weight spaceship (MWSS) |  |
| Heavy-weight spaceship (HWSS) |  |

https://github.com/Ben-Avrahami/Conway-s-Game-of-Life


https://www.youtube.com/watch?v=ZBLoffoZLH8

# Systemic, strategic, and narrative emergence


https://www.youtube.com/watch?v=OrmyLaLCaIo

# Contents

- Intro to emergence in games
- **Mechanics, Dynamics & Aesthetics (MDA)**
- Some additional resources

# Mechanics, Aesthetics, Dynamics (MDA)

- Mechanics (rules, verbs – what is designed and implemented)
- Dynamics (behavior patterns, strategies emerging from mechanics)
- Aesthetics (emotions & Player experience emerging from dynamics)

## MDA: A Formal Approach to Game Design and Game Research

**Robin Hunicke, Marc LeBlanc, Robert Zubek**

hunicke@cs.northwestern.edu, marc_leblanc@alum.mit.edu, rob@cs.northwestern.edu

### Abstract

In this paper we present the MDA framework (standing for Mechanics, Dynamics, and Aesthetics), developed and taught as part of the Game Design and Tuning Workshop at the Game Developers Conference, San Jose 2001-2004.

MDA is a formal approach to understanding games – one which attempts to bridge the gap between game design and development, game criticism, and technical game research. We believe this methodology will clarify and strengthen the iterative processes of developers, scholars and researchers alike, making it easier for all parties to decompose, study and design a broad class of game designs and game artifacts.

### Introduction

All artifacts are created within some design methodology. Whether building a physical prototype, architecting a software interface, constructing an argument or implementing a series of controlled experiments – design methodologies guide the creative thought process and help ensure quality work.

Specifically, iterative, qualitative and quantitative analyses support the designer in two important ways. They help her analyze the *end result* to refine implementation, and analyze the *implementation* to refine the result. By approaching the task from both perspectives, she can consider a wide range of possibilities and interdependencies.

This is especially important when working with computer and video games, where the interaction between coded subsystems creates complex, dynamic (and often unpredictable) behavior. Designers and researchers must consider interdependencies carefully before implementing changes, and scholars must recognize them before drawing conclusions about the nature of the experience generated.

In this paper we present the MDA framework (standing for Mechanics, Dynamics, and Aesthetics), developed and taught as part of the Game Design and Tuning Workshop at the Game Developers Conference, San Jose 2001-2004 [LeBlanc, 2004a]. MDA is a formal approach to understanding games – one which attempts to bridge the gap between game design and development, game criticism, and technical game research. We believe this methodology will clarify and strengthen the iterative processes of developers, scholars and researchers alike, making it easier for all parties to decompose, study and design a broad class of game designs and game artifacts.

### Towards a Comprehensive Framework

Game design and authorship happen at many levels, and the fields of games research and development involve people from diverse creative and scholarly backgrounds. While it's often necessary to focus on one area, everyone, regardless of discipline, will at some point need to consider issues outside that area: base mechanisms of game systems, the overarching design goals, or the desired experiential results of gameplay.

AI coders and researchers are no exception. Seemingly inconsequential decisions about data, representation, algorithms, tools, vocabulary and methodology will trickle upward, shaping the final gameplay. Similarly, all desired user experience must bottom out, somewhere, in code. As games continue to generate increasingly complex agent, object and system behavior, AI and game design merge.

Systematic coherence comes when conflicting constraints are satisfied, and each of the game's parts can relate to each other as a whole. Decomposing, understanding and creating this coherence requires travel between all levels of abstraction – fluent motion from systems and code, to content and play experience, and back.

We propose the MDA framework as a tool to help designers, researchers and scholars perform this translation.

### MDA

Games are created by designers/teams of developers, and consumed by players. They are purchased, used and eventually cast away like most other consumable goods.


The production and consumption of game artifacts.

Designers can only directly control the mechanics

# Predicting player behavior becomes even more difficult with multiple players

# MDA Aesthetics (≈ player experience)

- Sensation (Game as sense-pleasure)
- Fantasy (Game as make-believe)
- Narrative (Game as drama)
- Challenge (Game as obstacle course)
- Fellowship (Game as social framework)
- Discovery (Game as uncharted territory)
- Expression (Game as self-discovery)
- Submission (Game as pastime)

# MDA Aesthetics

- Sensation (Game as sense-pleasure)
- Fantasy (Game as make-believe)
- Narrative (Game as drama)
- Challenge (Game as obstacle course)
- Fellowship (Game as social framework)
- **Discovery (Game as uncharted territory)**
- Expression (Game as self-discovery)
- Submission (Game as pastime)

# Basic human psychological needs

- Need for competence
- **Need for autonomy**
- Need for social relatedness
- **Need for novelty (of sensations and experiences)**

# Other game pleasures

Anticipation

Delight

Gift giving

Humor

Purification

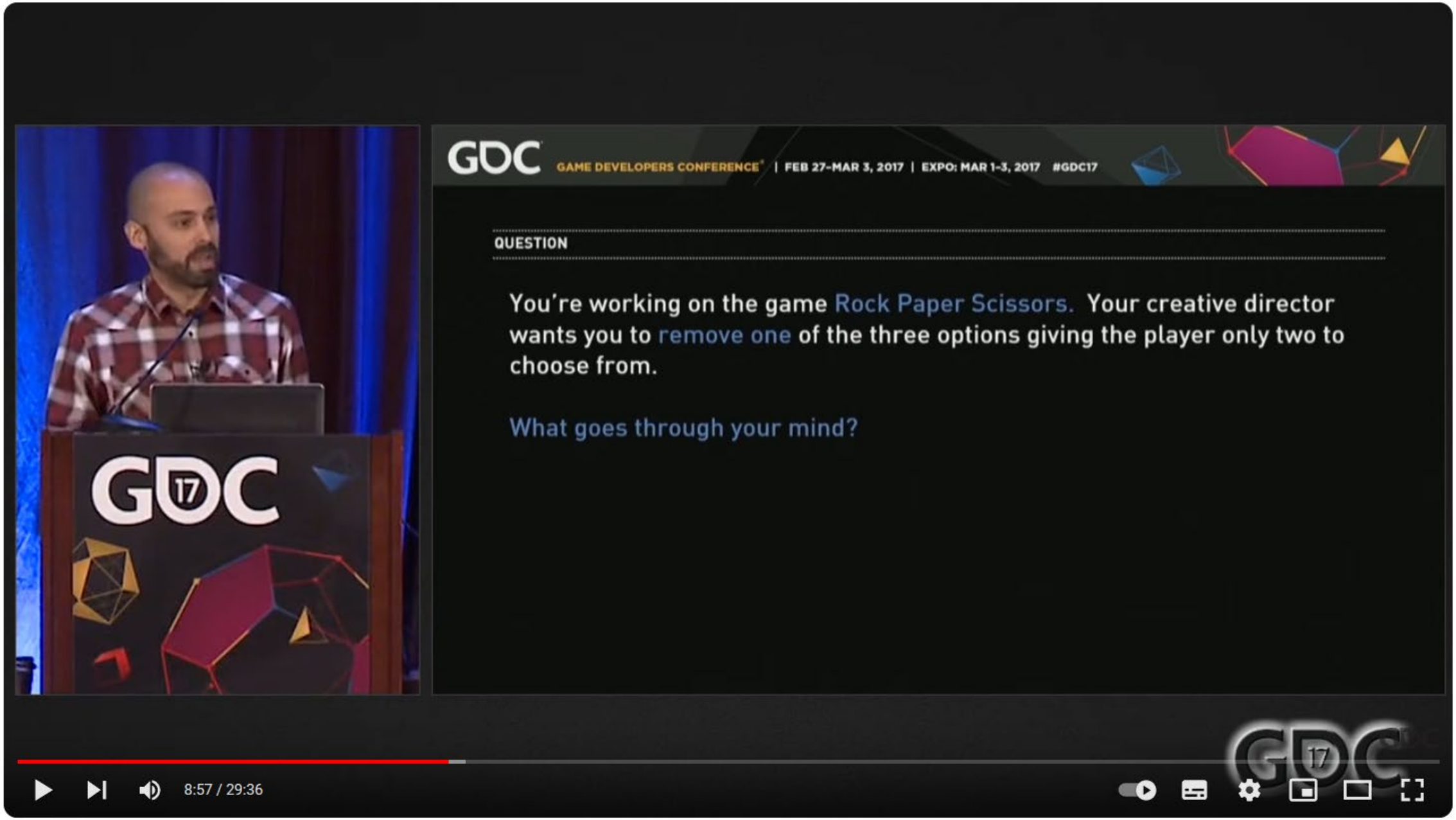
Thrill

Wonder

…

# MDA Example: Rock, paper, scissors

- Mechanics: Rock beats scissors, scissors beat paper, paper beats rock
- Dynamics: Random and balanced play (every choice is equally probable to win). Tool for social decision making.
- Aesthetics: Simple social fun.




Interviewing For Game Design

https://www.youtube.com/watch?v=uUQKbowVsIE

GDC
492K subscribers
Subscribe

3.2K Share Clip Save


GDC GAME DEVELOPERS CONFERENCE | FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 #GDC17

Q You're working on the game **Rock Paper Scissors**. Your creative director wants you to remove one of the three options giving the player only two to choose from. **What goes through your mind?**

## Bad Answer

- "I would ask why he wants to change it..."
- "I don't like removing options..."
- "I would try to convince him not to do that..."
- × "I'd get the team together and we'd figure it out..."
- × "I would remove Paper..."

UBM


GDC GAME DEVELOPERS CONFERENCE | FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 #GDC17

Q You're working on the game **Rock Paper Scissors**. Your creative director wants you to remove one of the three options giving the player only two to choose from. **What goes through your mind?**

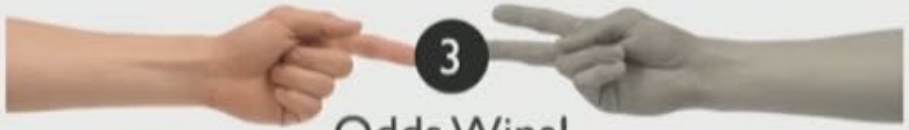
## Good Answers

- √ "IMPOSSIBLE!!!"
- √ "That would destroy the balance of the game."
- √ "If you remove Paper, then Rock would always win..."
- √ "You can't do RPS with two options...one always wins or it's always stalemate.

So What's Your Response to the Creative Director?


GDC GAME DEVELOPERS CONFERENCE | FEB 27–MAR 3, 2017 | EXPO: MAR 1–3, 2017 #GDC17

## Great Answer




Odds Wins!

Odds

Evens

Extra Step – Select Teams

# MDA Example: Monopoly

- Mechanics: movement and transaction rules
- Dynamics: the rich get richer and the course of the game is set very early
- Aesthetics/experience: power fantasy, challenge, fellowship

# MDA Example: Monopoly

- Mechanic: Auction (if player does not want to buy a property they land on, it is auctioned)
- Dynamic: All properties are bought early in the game
- Aesthetics/experience: Might feel ”unfair”, but fixes a problem of the game being too long without it

# MDA Example: Knizia scoring

- Mechanic: Player counts for scoring only the resources of which he has the least
- Dynamic: Players collect all kinds of resources.
- Aesthetics/experience: Increased variety of gameplay (no motivation to keep hoarding just one resource that one happens to gain early in the game)

# MDA Example: Drop item

- Mechanic: Player is able to get rid of excess inventory by dropping items on the ground
- Dynamic: Trading, bartering
- Aesthetics/experience: Increased social connection with other players

# MDA Example: Chess

- Mechanics: rules such as ”A player cannot move their king into check”
- Dynamics: sacrificing lesser pieces to prevent checkmate or capture the opponent’s queen
- Aesthetics/experience: challenge, discovery, fellowship

# MDA Example: Physics/dynamics simulation

- Mechanics: Newton's laws of motion
- Dynamics: Domino effects / chain reactions, shooting a target by bouncing a projectile, stability/instability of buildings...
- Aesthetics: Surprise, curiosity, slapstick comedy

# Designing for Emergence: PlusMinus

Aalto student game, started on the Game Project class
CHI PLAY game design competition winner, also showcased at GDC Experimental Gameplay Workshop 2019


**The 2019 Experimental Gameplay Workshop**

https://youtu.be/RNA-Eak_aso?si=fCnEHyDaggSQln65&t=3186

GDC 2025
525 t. tilaajaa

Tilattu

369

Jaa

Klippi

Tallenna

Aalto University

School of Science

Master's Programme in Computer, Communication and Information Sciences

Antti Sandberg

**Plusminus: Level Design for Emergent Gameplay**

https://aaltodoc.aalto.fi/items/205b2f1e-23b6-43ab-8633-b43e2c381709

Master's Thesis

Espoo, June 20, 2019

Supervisor and Advisor: Perttu Hämäläinen, Professor

Aalto University
School of Science
Degree Programme of Computer Science and Engineering

Juuso Toikka

# Developing the Mechanics of Plusminus:

## Designing for Emergence and Control in a Physics-Based Game

https://aaltodoc.aalto.fi/items/4497354e-21e3-4abb-b5ce-c5cdc29f9ec8

Master's Thesis
Espoo, July 21, 2019

Supervisor: Perttu Hämäläinen
Instructor: Perttu Hämäläinen

# Predictive physics simulation

- Sometimes too emergent, chaotic, uncontrollable
- In this paper, we explored game mechanics based on simulating the physics forward in time in each frame



## Predictive Physics Simulation in Game Mechanics

**Perttu Hämäläinen, Xiaoxiao Ma, Jari Takatalo**
Aalto University
firstname.lastname@aalto.fi

**Julian Togelius**
NYU Tandon School of Engineering
julian@togelius.com

**ABSTRACT**

**INTRODUCTION**

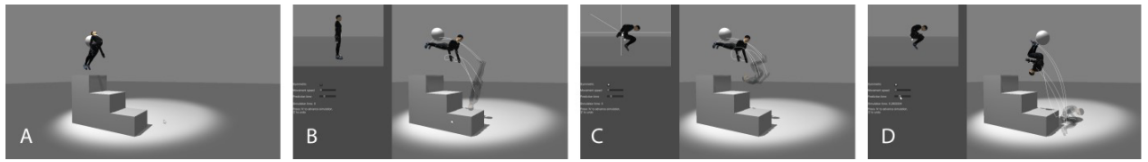



**Figure 1. Our "animation as a game" prototype, where the player's goal is to save the falling character. A) Initial situation where the character is hit by a projectile. B) Game interface appears on the left, and the main view shows predictive movement trajectories and future pose; the character will hit its head unless the player acts. C) Using the posing controls (top-left), the player adjusts the future pose. Tucking increases rotation speed, which is predicted and visualized in real-time. D) Once satisfied with the prediction, the player advances the simulation. See the supplemental video at 03:00 for live gameplay.**

Computers can now simulate simple game physics systems hundreds of times faster than real-time, which enables real-time prediction and visualization of the effects of player actions. Predictive simulation is traditionally used as part of planning and game AI algorithms; we argue that it presents untapped potential for game mechanics and interfaces. We explore this notion through 1) deriving a four-quadrant design space model based on game design and human motor control literature, and 2) developing and evaluating six novel prototypes that demonstrate the potential and challenges of each quadrant. Our work highlights opportunities in enabling direct control of complex simulated characters, and in transforming real-time action into turn-based puzzles. Based on our results, adding predictive simulation to existing game mechanics is less promising, as it may feel alienating or make a game too easy. However, the approach may still be useful for game designers, for example, as it allows one to test designs beyond one's playing skill.

**Author Keywords**
Game design; visualization; physics simulation;

**ACM Classification Keywords**
H.5.2. User interfaces: Interaction styles



Simulation of dynamic physical systems utilizes a set of rules that are unambiguous and mathematically well-defined [5], but can at the same time lead to emergent, rich behavior. This has inspired many types of physically based games and game mechanics, with popular examples such as The Incredible Machine [4], Angry Birds [25], Cut the Rope [36], Where's My Water [3], QWOP [2], and Portal [34]. Gradually, game physics has evolved towards more complex systems, i.e., from 2D to 3D, and from rigid bodies (e.g., stacked objects) to soft bodies (e.g., ropes and cloth) and fluids. During recent years, no fundamentally new physics simulation types have appeared, but on the other hand, personal computing devices can now simulate simple systems hundreds or even thousands times faster than real-time. This paper focuses on capitalizing this excess computing power for novel game mechanics and interfaces.

In this paper, we argue that *real-time predictive physics simulation* provides an underexplored tool for game design. Most games simulate physical systems at a fixed rate such as 60 simulation steps per second, and only use simulations in a *reactive* manner to compute the interactions of objects based on player input. In contrast, we denote by predictive simulation a process of:

1) Saving the current simulation state
2) Simulating the physics forward for multiple steps, up to a prediction horizon,
3) Using the simulation results to preview and evaluate the results of player decisions
4) Restoring simulation state back to the saved state, so that the game may continue without discontinuities.

# Predictive Physics Simulation in Game Mechanics

Perttu Hämäläinen, Ma Xiao Xiao, Jari Takatalo, Julian Togelius

CHI PLAY 2017

# Novel technology that enables gameplay


The Incredible Machine (Dynamix, 1993)


ANGRY BIRDS
HD
LOADING...

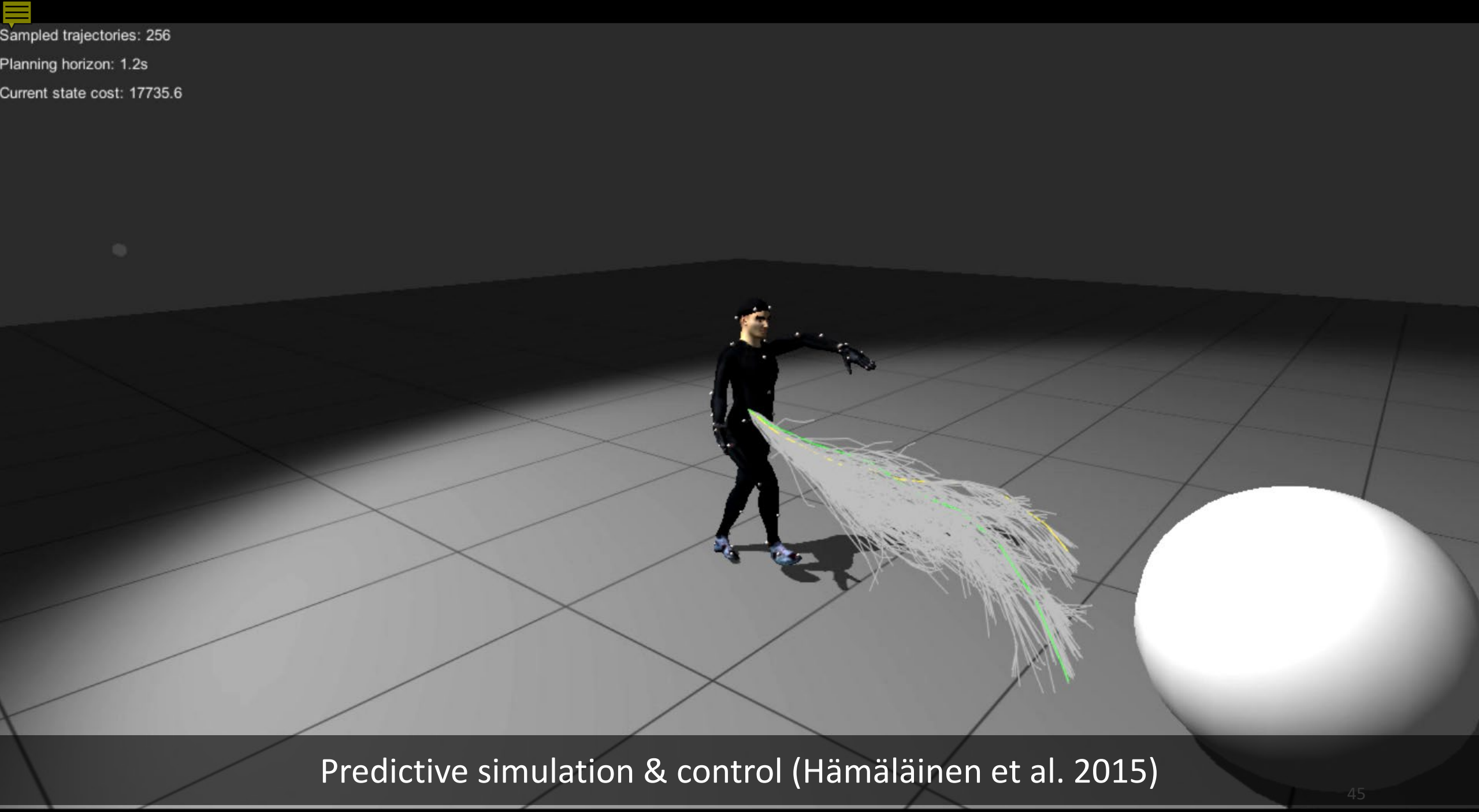

Cut the Rope (ZeptoLab, 2010)


Where’s my Water (Creature Feep, 2011)

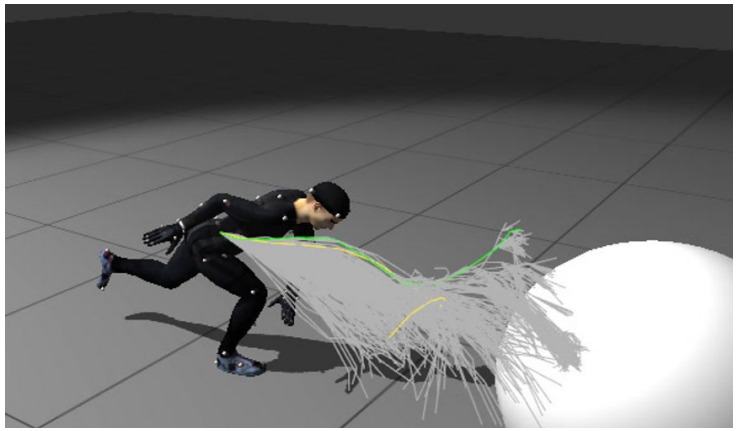

Cities Skylines (Colossal Order, 2015)

1000x faster than real-time physics simulation?

Reactive simulation (Just Cause 3)

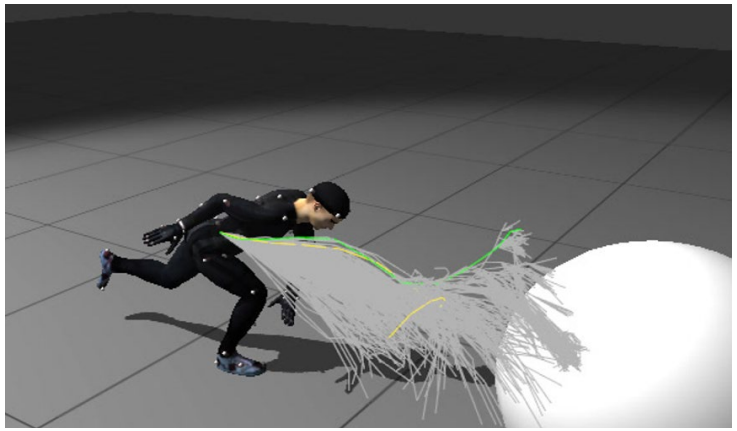



Predictive simulation & control (Hämäläinen et al. 2015)

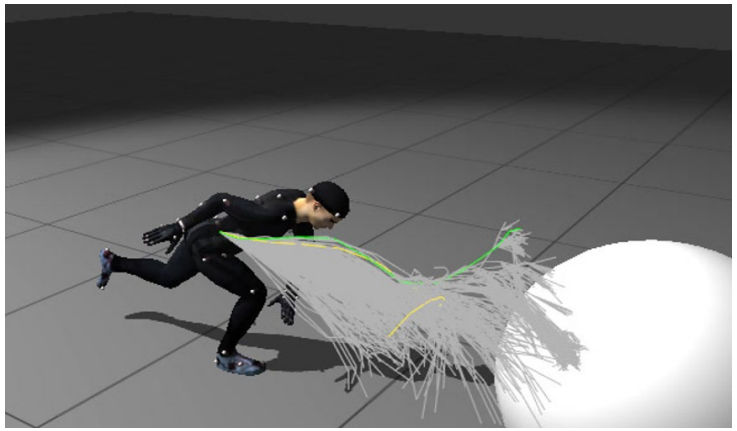


REACTIVE → PREDICTIVE

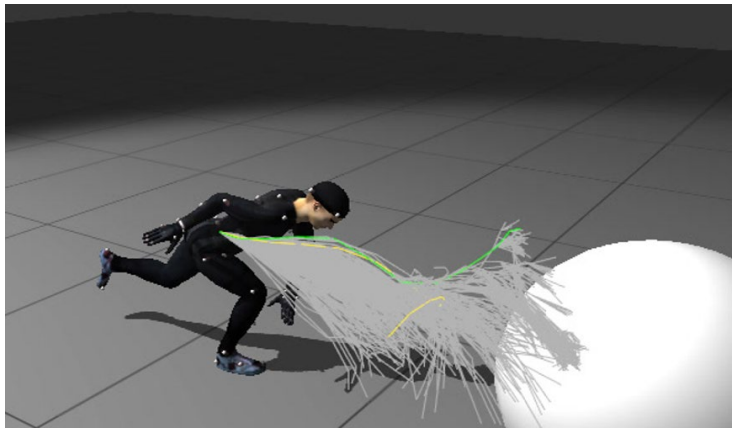

QWOP (Bennet Foddy, 2008)



REACTIVE → PREDICTIVE

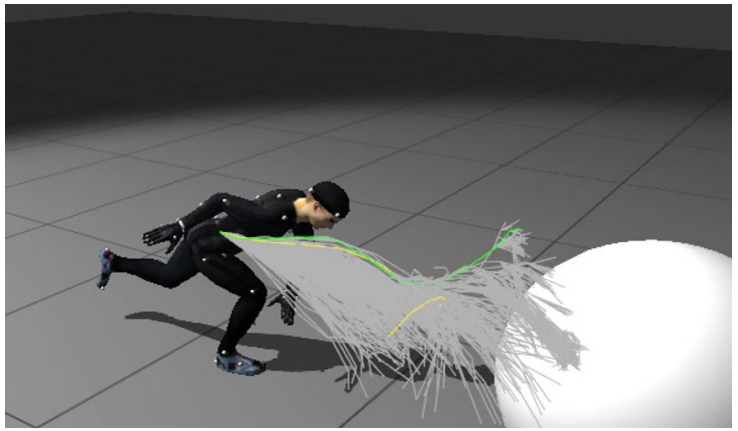


REACTIVE → PREDICTIVE



REACTIVE

PREDICTIVE

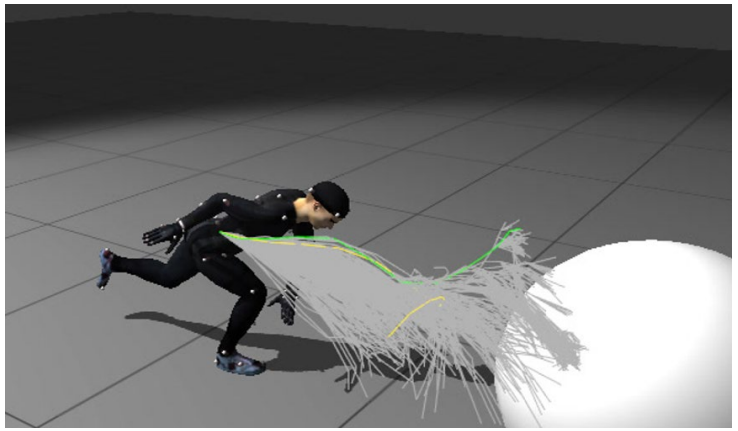


REACTIVE

PREDICTIVE

LOW LEVEL

HIGH LEVEL



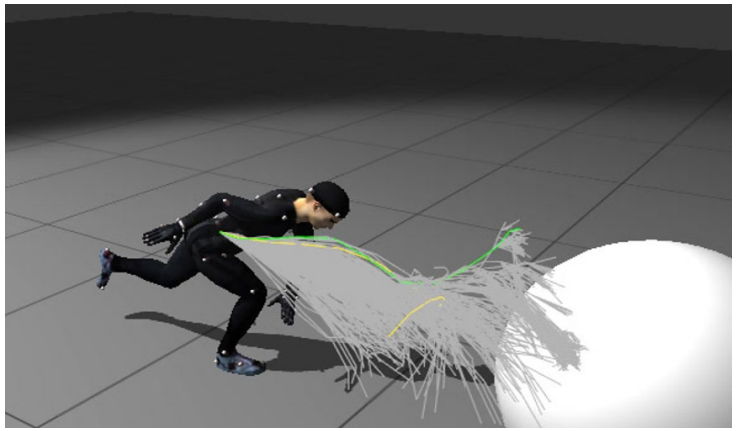
REACTIVE

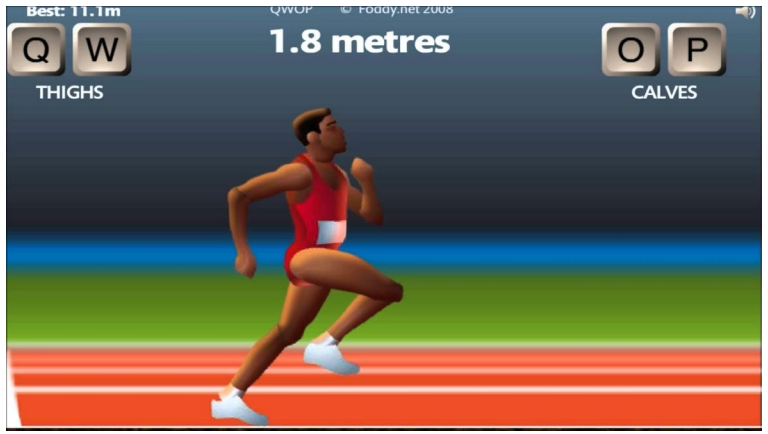
PREDICTIVE

LOW LEVEL

HIGH LEVEL

REACTIVE

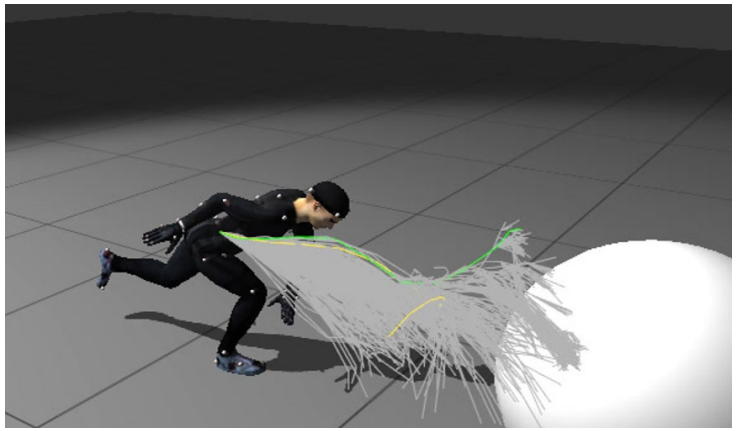





PREDICTIVE

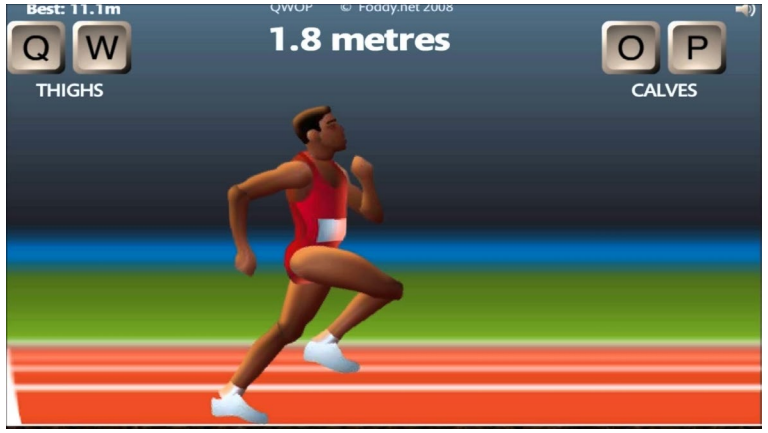
LOW LEVEL

HIGH LEVEL



REACTIVE

PREDICTIVE



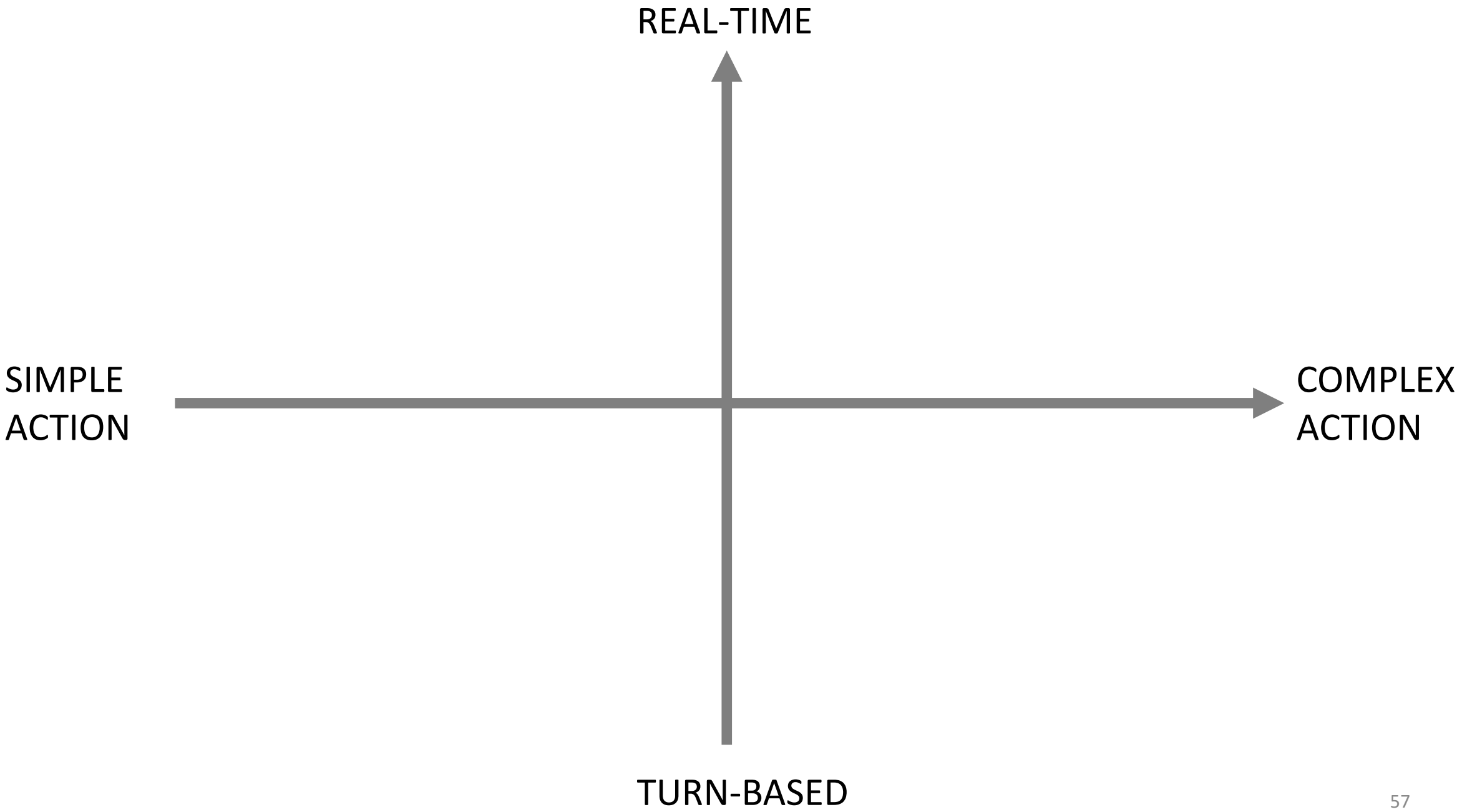

?

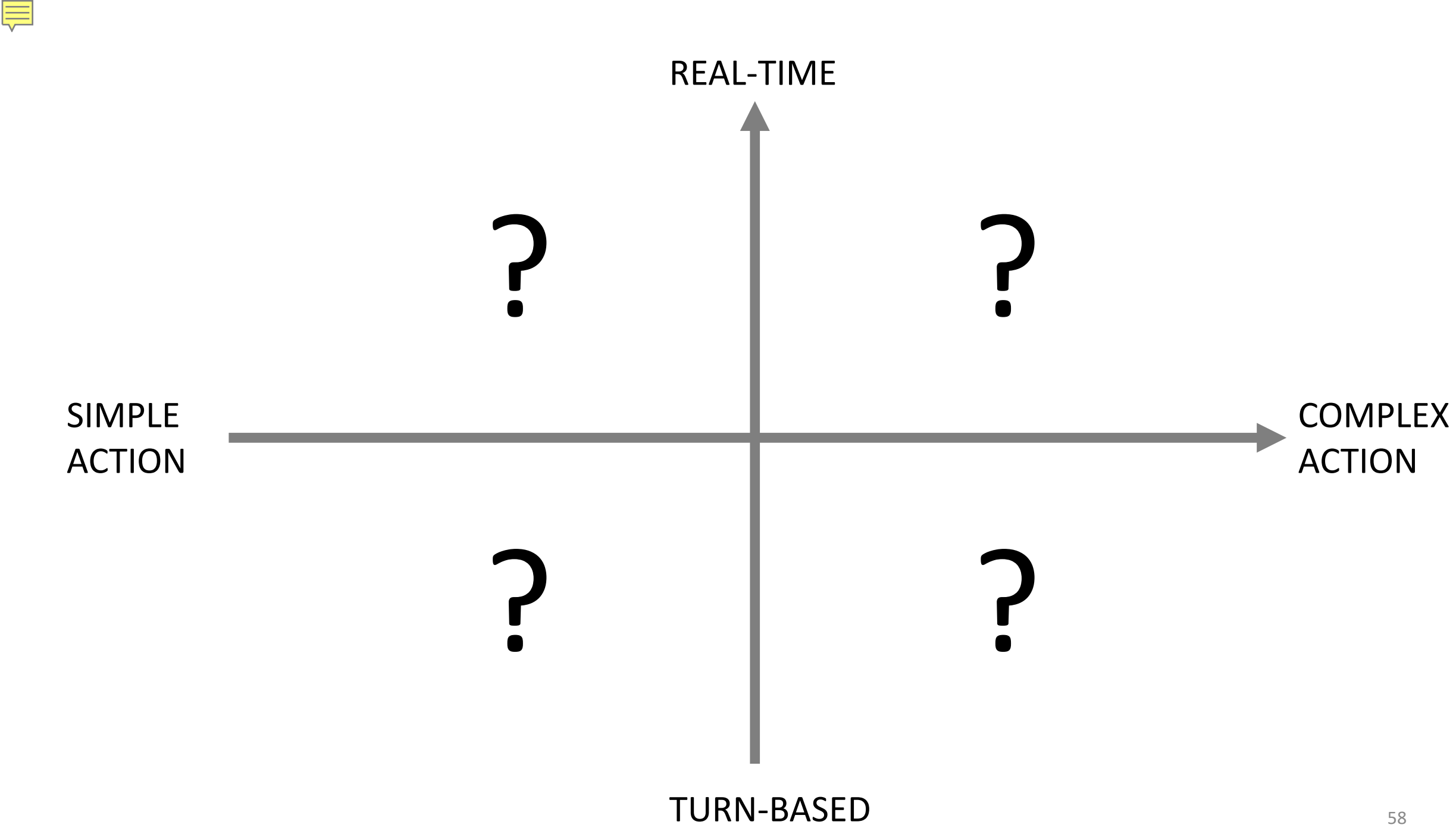
LOW LEVEL

# Prediction enables (re)planning of actions

SIMPLE ACTION

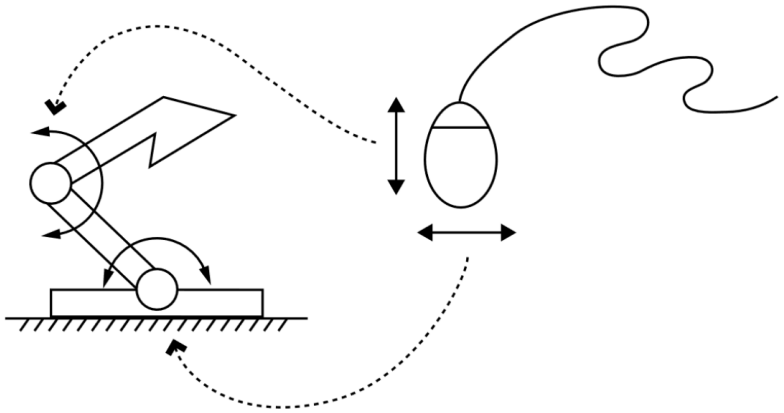
COMPLEX ACTION


REAL-TIME
SIMPLE ACTION
COMPLEX ACTION
TURN-BASED

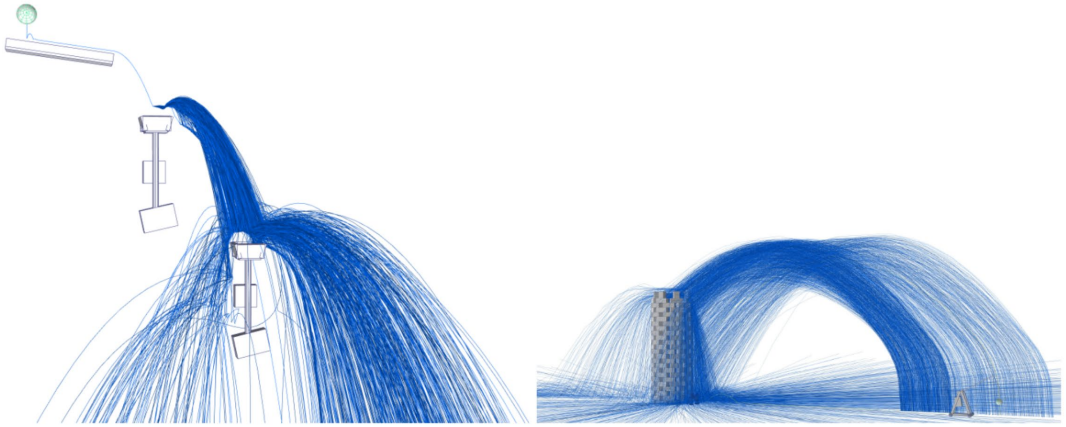

REAL-TIME
?
?
SIMPLE ACTION
COMPLEX ACTION
?
?
TURN-BASED

# Related work

Lazlo et al. 2000 (SIGGRAPH)

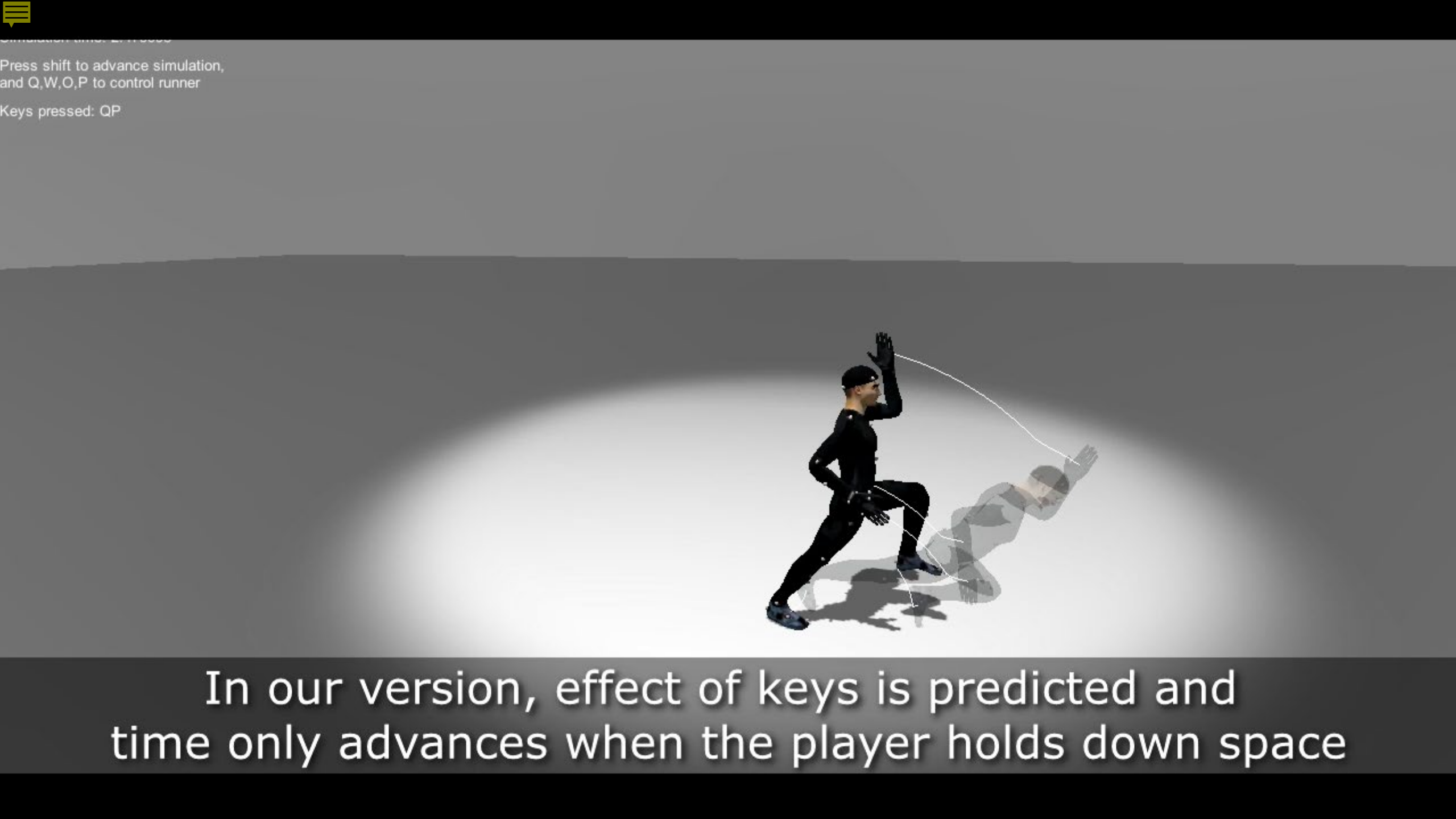


Twigg & James 2007 (SIGGRAPH)

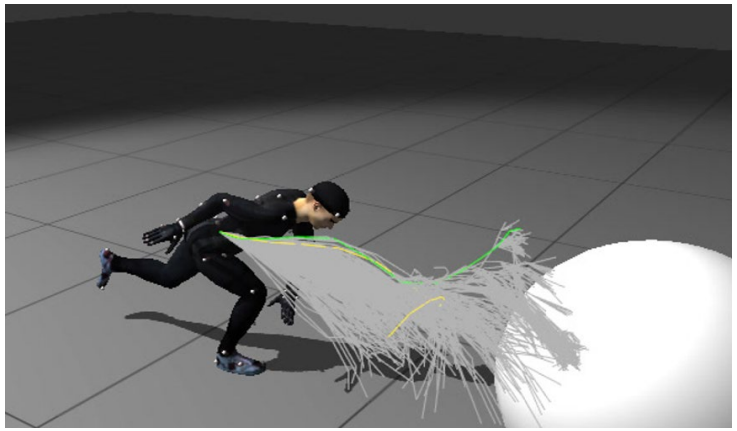


Toribash (Nabi Studios 2006)


QWOP (Bennet Foddy 2008)

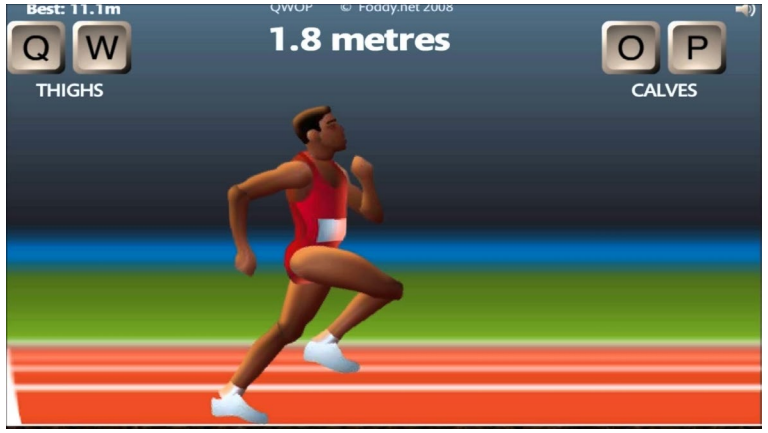

Press shift to advance simulation,
and Q,W,O,P to control runner
Keys pressed: QP
In our version, effect of keys is predicted and
time only advances when the player holds down space

HIGH LEVEL



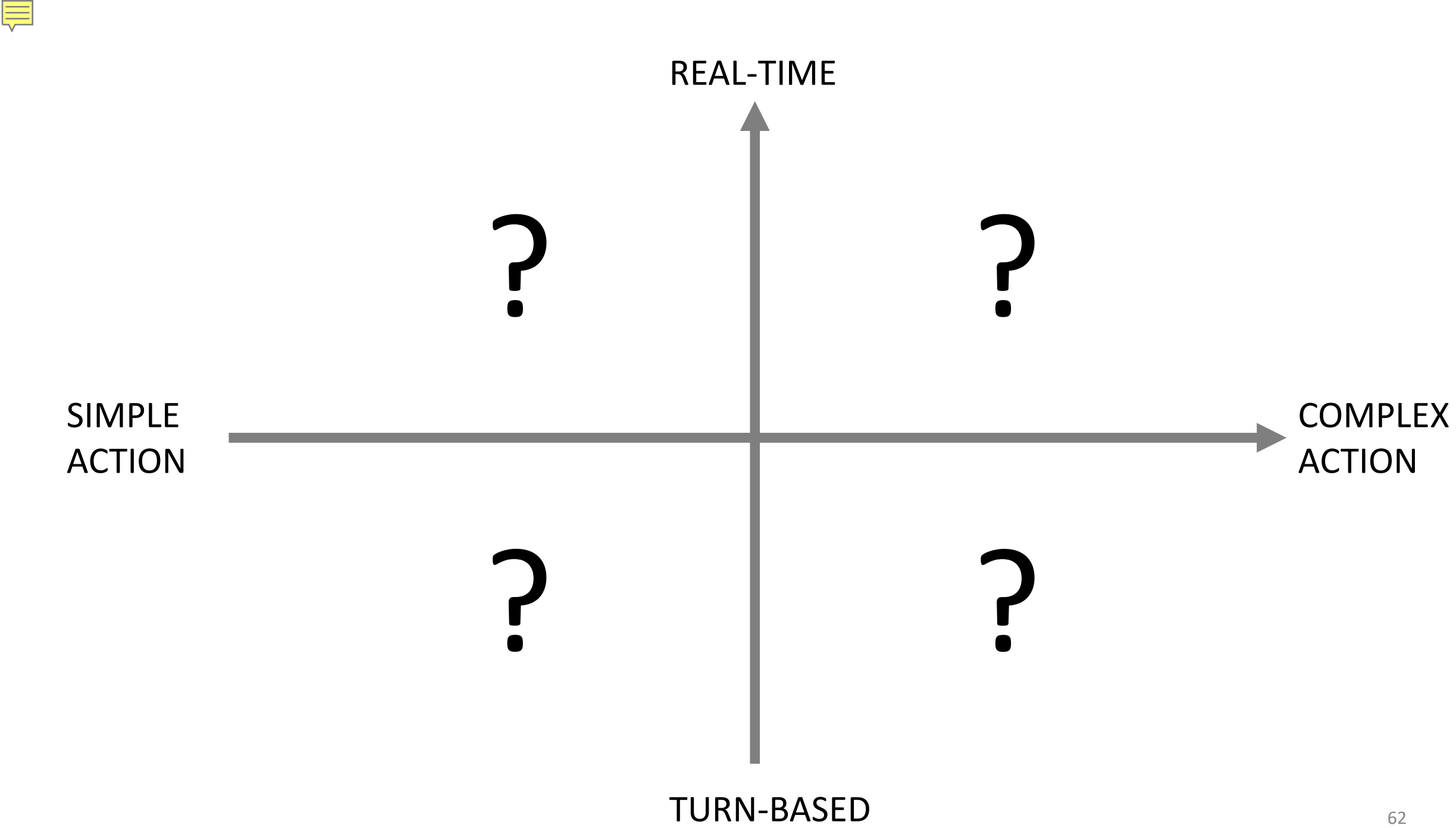
REACTIVE

PREDICTIVE

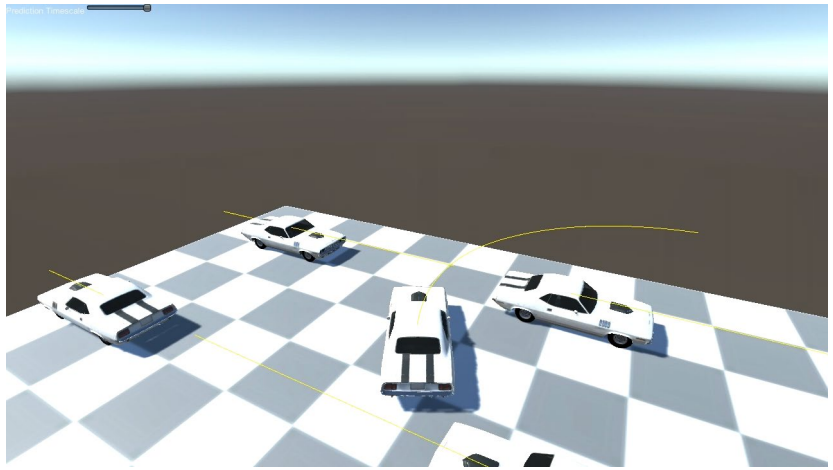



?

LOW LEVEL


REAL-TIME
?
?
SIMPLE ACTION
COMPLEX ACTION
?
?
TURN-BASED

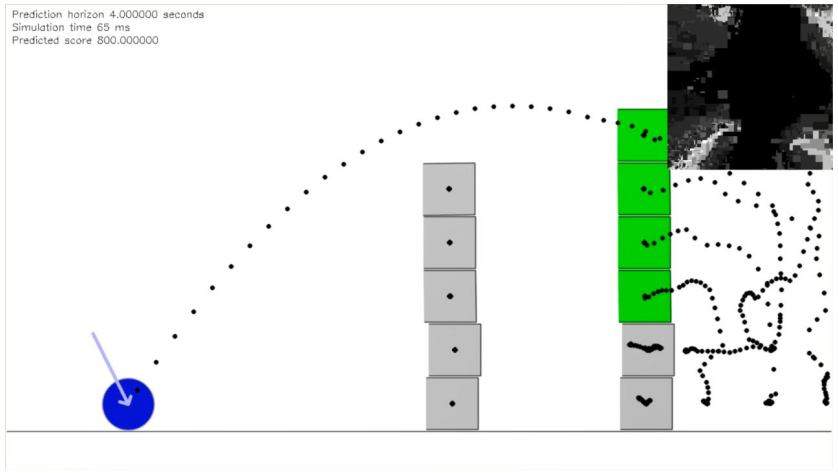
REAL-TIME



?

SIMPLE ACTION

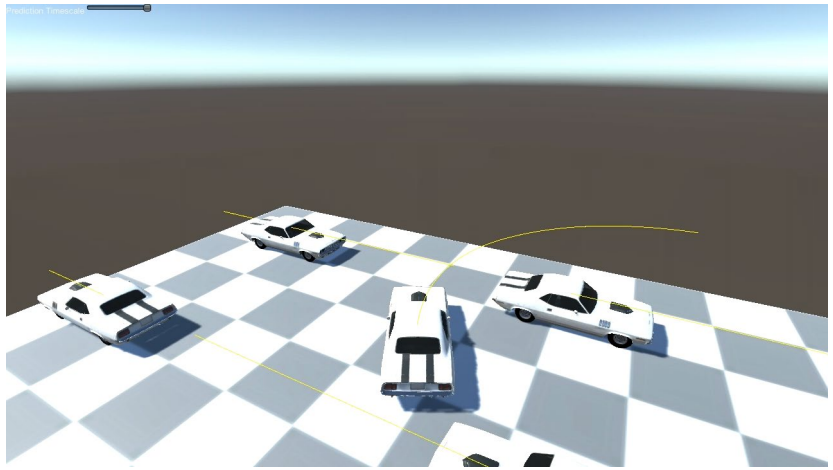
COMPLEX ACTION

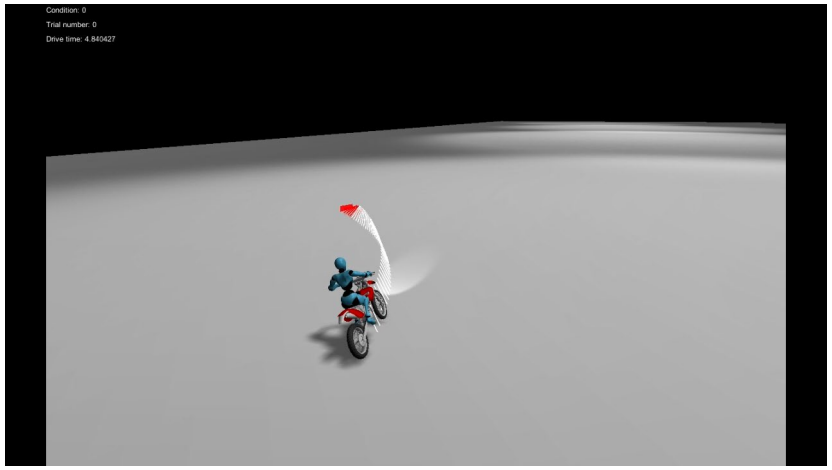



?

TURN-BASED

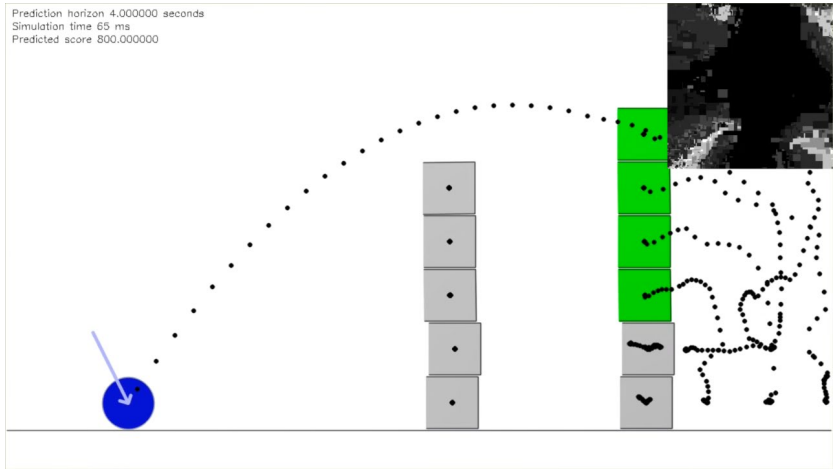
REAL-TIME






SIMPLE ACTION

COMPLEX ACTION



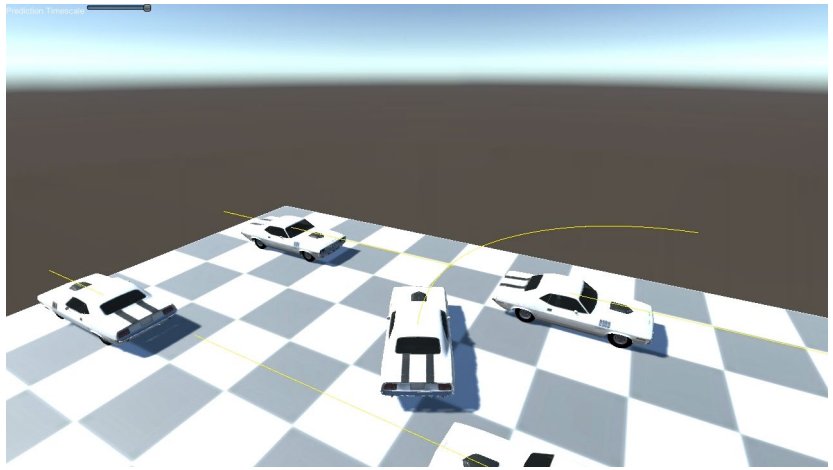

?

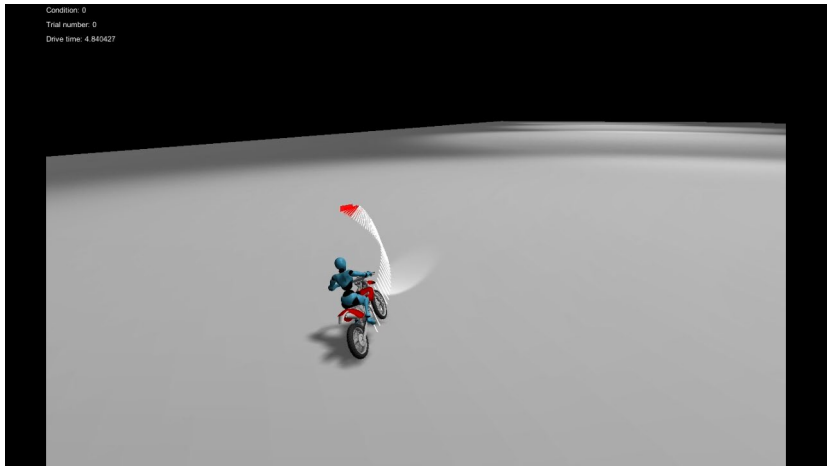
TURN-BASED

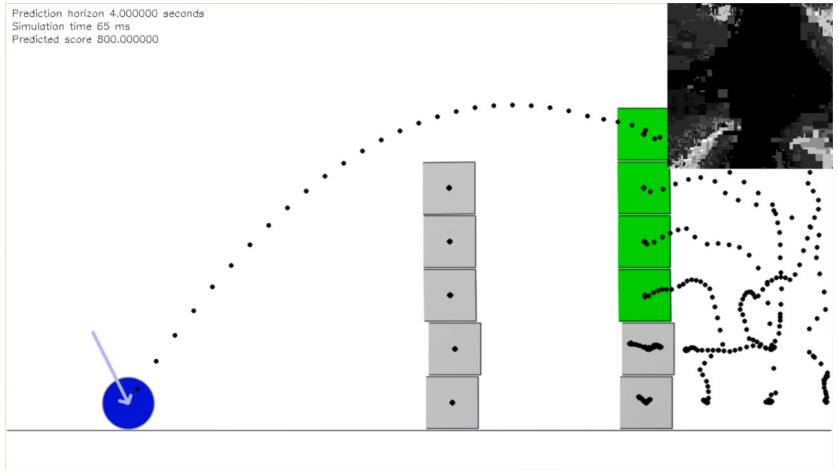
REAL-TIME

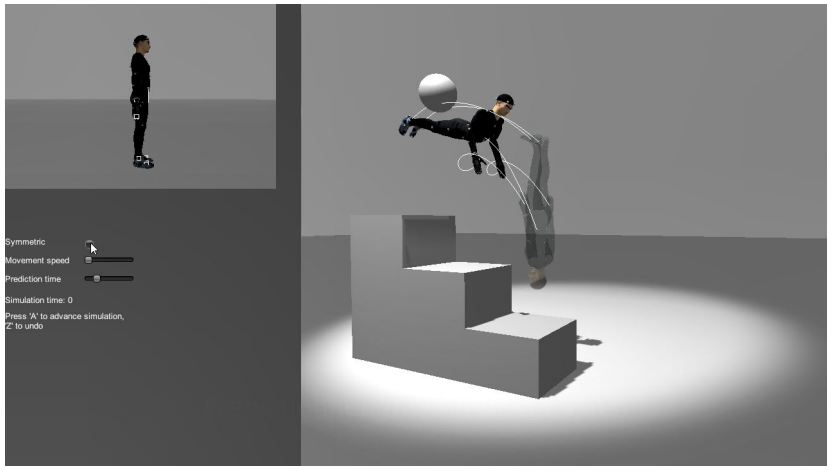
SIMPLE ACTION

COMPLEX ACTION





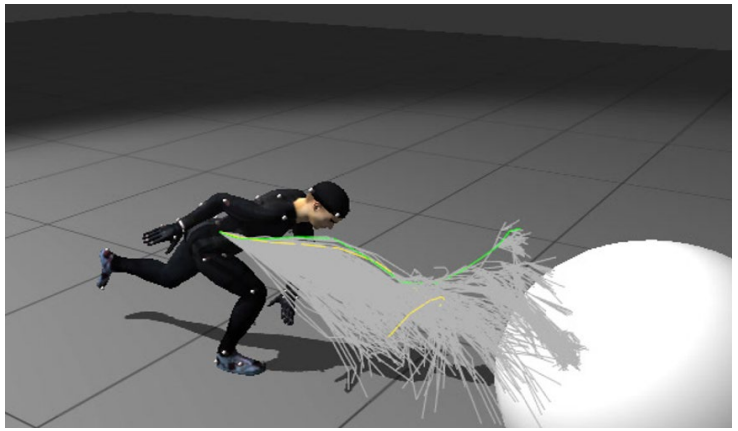



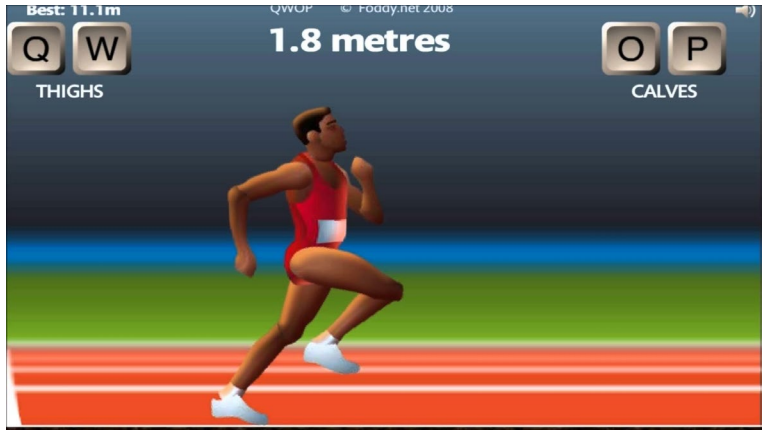

TURN-BASED


https://www.funnypica.com/top-30-funny-bike-accidents/

HIGH LEVEL

REACTIVE






?

PREDICTIVE

LOW LEVEL

# MDA Example: What mechanics make this dynamic emerge?

- Dynamic: Turtling, camping
- Mechanics: predetermined spawn locations (camping where a valuable item is due to spawn)
- More generally: lower cost or higher rewards in passive than active play

# MDA Example: What mechanics make this dynamic emerge?

- Dynamic: Button mashing
- Mechanics: Actions with no cost and no cooldown

# Additional resources

## Simulating Mechanics to Study Emergence in Games

**Joris Dormans**
Amsterdam University of Applied Sciences
Duivendrechtsekade 36-38
1096 AH, Amsterdam, the Netherlands

**Abstract**

This paper presents the latest version of the Machinations framework. This framework uses diagrams to represent the flow of tangible and abstract resources through a game. This flow represents the mechanics that make up a game's interbal economy and has a large impact on the emergent gameplay of most simulation games, strategy games and board games. This paper shows how Machinations diagrams can be used simulate and balance games before they are built.

Games that display dynamic, emergent behavior are hard to design. One of the biggest challenges is that the mechanics of these games require a delicate balance. Developers must rely on frequent game testing to test for this balance. This requires much time, but also a functional prototype of the game. This paper looks into simulating games in early stages of development, before prototypes are built, in order to design dynamic but balanced game mechanics effectively and efficiently. To this end it utilizes the most recent version of the Machinations framework for representing game mechanics. The Machinations tool has been extended to simulate dynamic games and to collect data from a multitude of simulated play session. In the second section of the paper, the tool is put to the test by simulating and balancing the game *SimWar*, which has been described by Will Wright, but never was built.

### The Machinations Framework

The *Machinations framework* formalizes a particular view on games as rule-based, dynamic systems. It focuses on game mechanics and the relation of these mechanics and the dynamic gameplay that emerges from them. It is based on the theoretical notion that structural features of game mechanics are for a large part responsible for the dynamic gameplay of the game as a whole. Game mechanics and their structural features are not immediately visible in most games. Some mechanics might be close to a game's surface, but many are obscured by the game's system. Only a detailed study of a game's mechanics can shed a light on the game's structure. Unfortunately, the models that are used to represent game mechanics, such as representations in code, finite state diagrams or Petri nets, are complex and not really accessible for non-programmers. What is more, these are ill-suited to represent games at a sufficient level of abstraction, on which structural features, such as feedback loops, become immediately apparent. To this end, the Machinations framework includes a diagrammatic language: *Machinations diagrams* which are designed to represent game mechanics in a way that is accessible, yet retains the same structural features and dynamic behavior of the game it represents. Earlier versions of the framework were presented elsewhere (Dormans 2008; 2009). The version presented here uses the same core elements as the previous version, although there have been minor changes and improvements. The main difference is a change in the resource connections which now can have only one label, where it could have two before (see below). In addition, support for artificial players, charts representing game state over time and collecting data from many simulated sessions are new additions.



The theoretical vision that drives the Machinations framework is that gameplay is ultimately determined by the flow of tangible and abstract resources through the game system. Machinations diagrams represent these flows and foreground the feedback structures that might exist within the game system. It is these feedback structures that for a large part determine the dynamic behavior of games (Hunicke, LeBlanc, and Zubek 2004; Salen and Zimmerman 2004). This is consistent with findings in the science of complexity that studies dynamic and emergent behavior in a wide variety of complex systems (Wolfram 2002; Fromm 2005).

Machinations diagrams have an exact and consistent syntax. This means that the diagrams can be interpreted by a computer, in fact, the Machinations software tool implements diagrams. In other words, they are interactive and dynamic, just like the games they are modeling, and can be executed in a similar way. It allows dynamic models of games to be designed and tested quickly and efficiently. The tool also allows the designer to quickly gather quantitative data from simulated play sessions. Unfortunately this property of the software tool does not translate to paper; the interactive tool, and the examples discussed in this paper, can be found on the Machinations web page: `http://www.jorisdormans.nl/machinations`.

BBC WORLD SERVICE

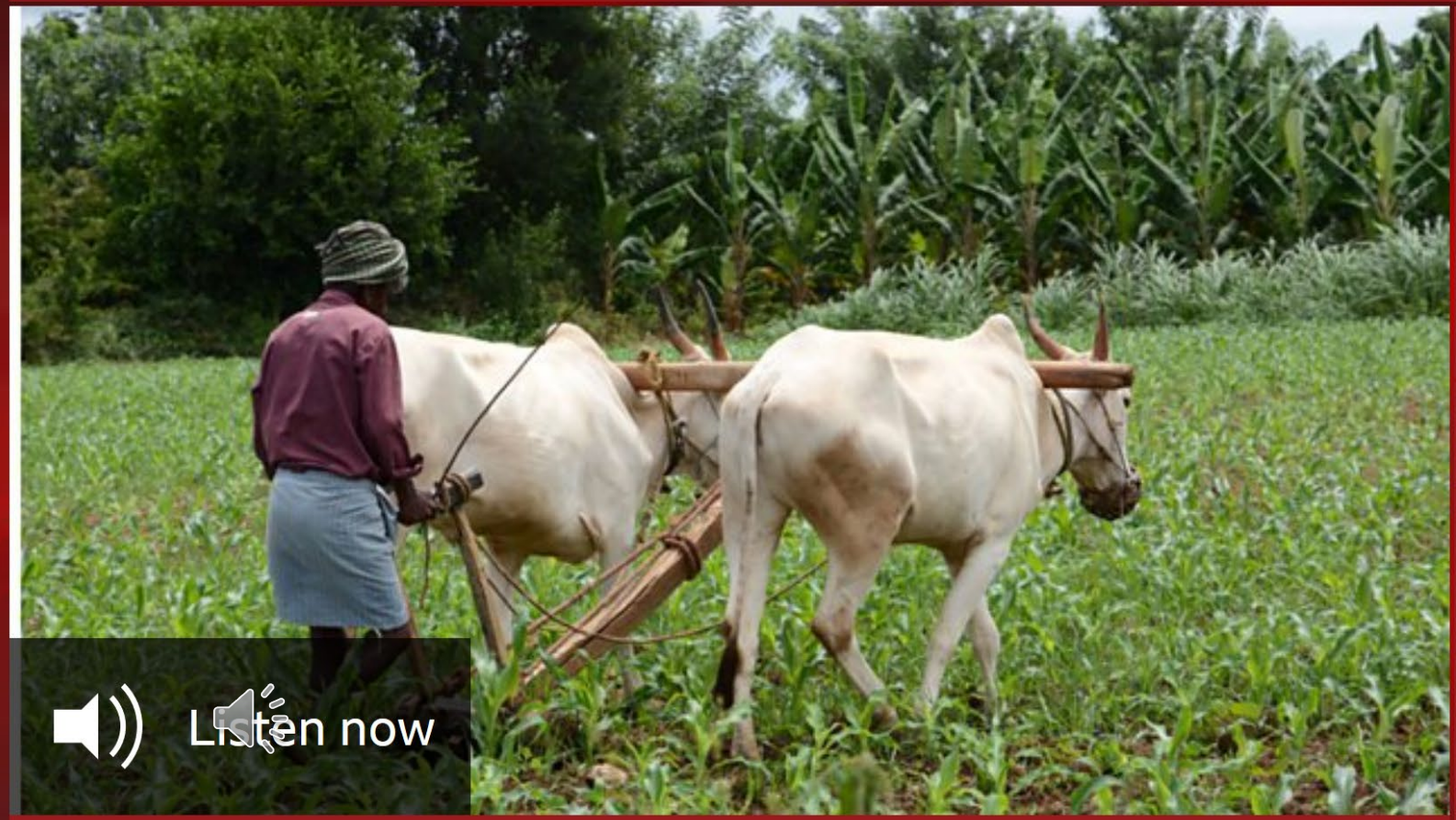
# 50 Things That Made the Modern Economy

LISTEN

On Now : **Sportsworld**

The Sportsworld Report

Home Episodes Clips Podcast The 51st Thing




## The Plough

https://www.bbc.co.uk/programmes/w3csvt0k

## Last on

BBC WORLD SERVICE

Mon 23 Oct 2017

06:50

Local time

**BBC WORLD SERVICE** ONLINE & UK DAB/FREEVIEW ONLY

## More episodes

**PREVIOUS**

**Cold Chain**

**NEXT**

**Number 51**

# MDA in the real world

- Mechanics: The plough
- Dynamics: Specialization, trade, rulers and ruled, social inequality, misogyny, tyranny, malnutrition, dense population, disease, war
- Aesthetics: Cultivation, Ownership, Safety, Fear

http://sciencenordic.com/how-heavy-plough-changed-world

# Machinations

https://machinations.io/

A visual tool for building and simulating, e.g., game economics models

## Simulating Mechanics to Study Emergence in Games

**Joris Dormans**
Amsterdam University of Applied Sciences
Duivendrechtsekade 36-38
1096 AH, Amsterdam, the Netherlands

**Abstract**

This paper presents the latest version of the Machinations framework. This framework uses diagrams to represent the flow of tangible and abstract resources through a game. This flow represents the mechanics that make up a game's interbal economy and has a large impact on the emergent gameplay of most simulation games, strategy games and board games. This paper shows how Machinations diagrams can be used simulate and balance games before they are built.

Games that display dynamic, emergent behavior are hard to design. One of the biggest challenges is that the mechanics of these games require a delicate balance. Developers must rely on frequent game testing to test for this balance. This requires much time, but also a functional prototype of the game. This paper looks into simulating games in early stages of development, before prototypes are built, in order to design dynamic but balanced game mechanics effectively and efficiently. To this end it utilizes the most recent version of the Machinations framework for representing game mechanics. The Machinations tool has been extended to simulate dynamic games and to collect data from a multitude of simulated play session. In the second section of the paper, the tool is put to the test by simulating and balancing the game *SimWar*, which has been described by Will Wright, but never was built.

### The Machinations Framework

The *Machinations framework* formalizes a particular view on games as rule-based, dynamic systems. It focuses on game mechanics and the relation of these mechanics and the dynamic gameplay that emerges from them. It is based on the theoretical notion that structural features of game mechanics are for a large part responsible for the dynamic gameplay of the game as a whole. Game mechanics and their structural features are not immediately visible in most games. Some mechanics might be close to a game's surface, but many are obscured by the game's system. Only a detailed study of a game's mechanics can shed a light on the game's structure. Unfortunately, the models that are used to represent game mechanics, such as representations in code, finite state diagrams or Petri nets, are complex and not really accessible for non-programmers. What is more, these are ill-suited to represent games at a sufficient level of abstraction, on which structural features, such as feedback loops, become immediately apparent. To this end, the Machinations framework includes a diagrammatic language: *Machinations diagrams* which are designed to represent game mechanics in a way that is accessible, yet retains the same structural features and dynamic behavior of the game it represents. Earlier versions of the framework were presented elsewhere (Dormans 2008; 2009). The version presented here uses the same core elements as the previous version, although there have been minor changes and improvements. The main difference is a change in the resource connections which now can have only one label, where it could have two before (see below). In addition, support for artificial players, charts representing game state over time and collecting data from many simulated sessions are new additions.



The theoretical vision that drives the Machinations framework is that gameplay is ultimately determined by the flow of tangible and abstract resources through the game system. Machinations diagrams represent these flows and foreground the feedback structures that might exist within the game system. It is these feedback structures that for a large part determine the dynamic behavior of games (Hunicke, LeBlanc, and Zubek 2004; Salen and Zimmerman 2004). This is consistent with findings in the science of complexity that studies dynamic and emergent behavior in a wide variety of complex systems (Wolfram 2002; Fromm 2005).

Machinations diagrams have an exact and consistent syntax. This means that the diagrams can be interpreted by a computer, in fact, the Machinations software tool implements diagrams. In other words, they are interactive and dynamic, just like the games they are modeling, and can be executed in a similar way. It allows dynamic models of games to be designed and tested quickly and efficiently. The tool also allows the designer to quickly gather quantitative data from simulated play sessions. Unfortunately this property of the software tool does not translate to paper; the interactive tool, and the examples discussed in this paper, can be found on the Machinations web page: `http://www.jorisdormans.nl/machinations`.

# Explorable explanations

- Play these ”explorable explanations”, alone or together with a pair:
  - Parable of Polygons (https://ncase.me/polygons/ )
  - Evolution of Trust (https://ncase.me/trust/ )
- Discuss and add thoughts, insights & lessons learned to Presemo. Consider, e.g.:
  - What are the mechanics and dynamics?
  - Systemic, strategic, narrative emergence?
  - What do you find most interesting or surprising about the mechanics, dynamics & aesthetics, and why?
- See also: https://explorabl.es/, http://www.complexity-explorables.org/

# MDA & explorable explanations debriefing

# Preparation for the rest of the course

For week 2:

- Play Clash Royale (reach 1-2 new arenas after the training ground)
- Watch this GDC talk on balancing Clash Royale: https://www.youtube.com/watch?v=bHLQQh8Ctu4

For week 3:

- Play Walking Dead No Man's Land  (Complete 1-2 chapters)