

# **Automated Puzzle Difficulty Estimation**

---

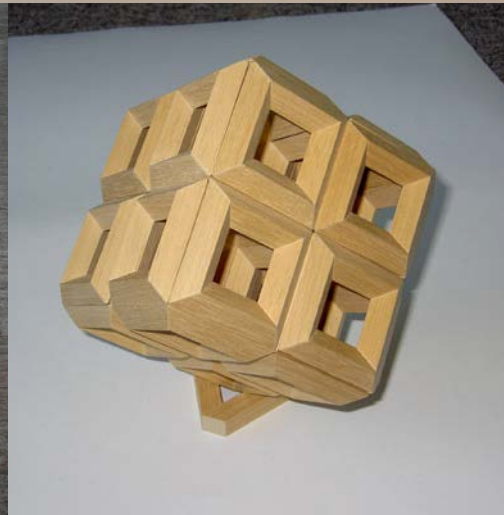
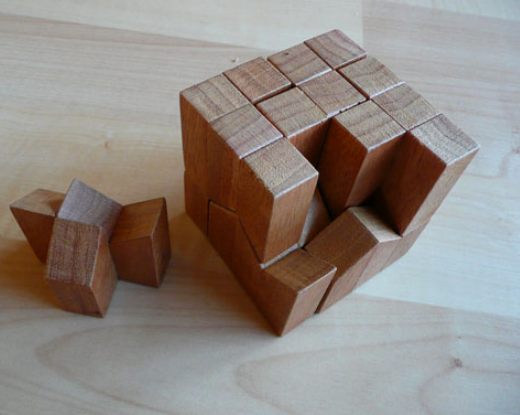
**Marc van Kreveld**

**Maarten Löffler**

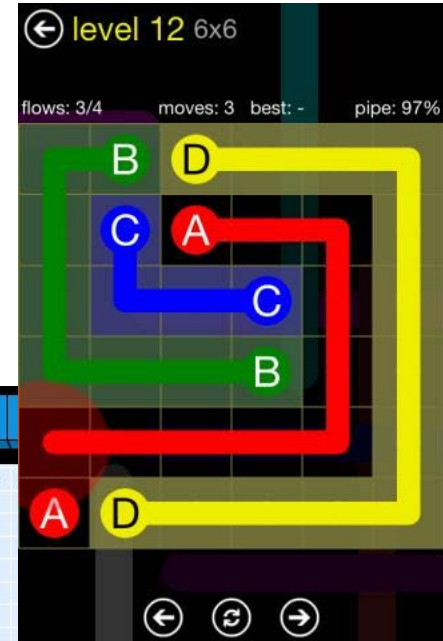
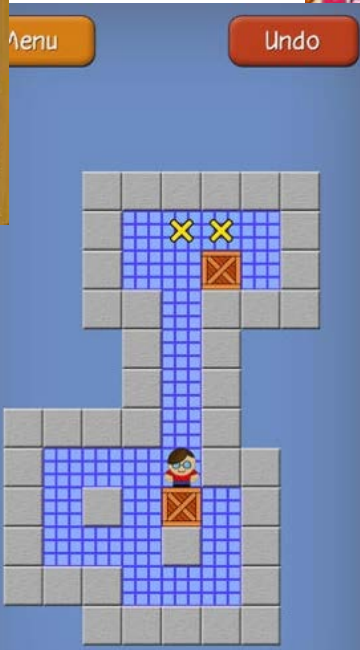
**Paul Mutser**

**Department of Information and Computing Sciences  
Utrecht University**

# Puzzles



# Puzzle Games



# Puzzle Games: Move

**MOVE**  
a brain shifting game

PLAY NOW

SETTINGS

STORE

INVITE FRIENDS

MORE GAMES

## < Levels

Get More Hints

- |                              |       |
|------------------------------|-------|
| 01.First Moves               | 1/100 |
| 3x3 board, 3 pawns, 1 color  |       |
| 02.Moving on                 | 0/100 |
| 3x3 board, 3 pawns, 2 colors |       |
| 03.Bust a Move               | 1/100 |
| 3x3 board, 3 pawns, 3 colors |       |
| 04.Move it or Lose it        | 0/100 |
| 4x4 board, 4 pawns, 1 color  |       |
| 05.Keep on Moving            | 3/100 |
| 4x4 board, 4 pawns, 2 colors |       |
| 06.Get a Move on             | 0/100 |

## < 03.Bust a Move

This pack has 100 levels

1-25



26

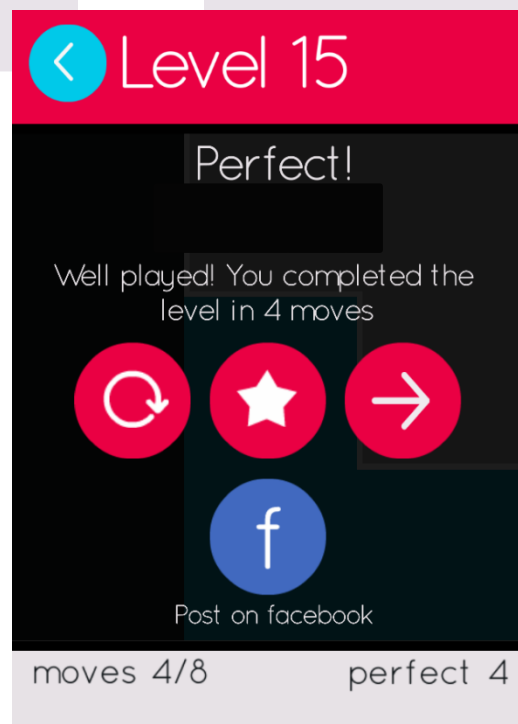
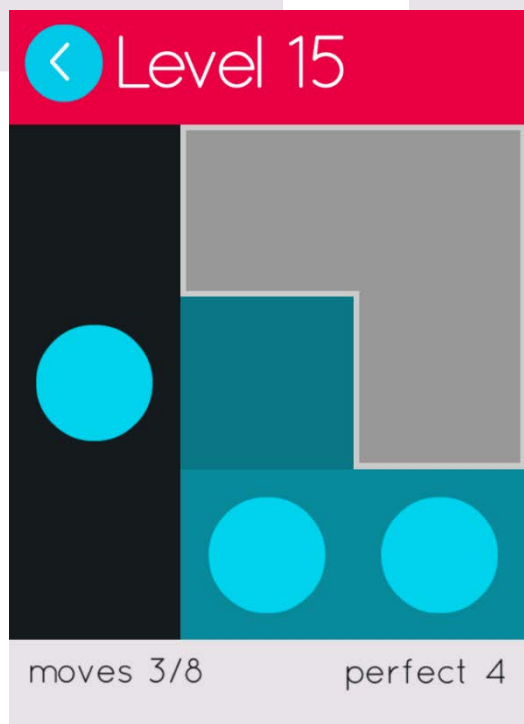
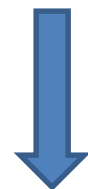
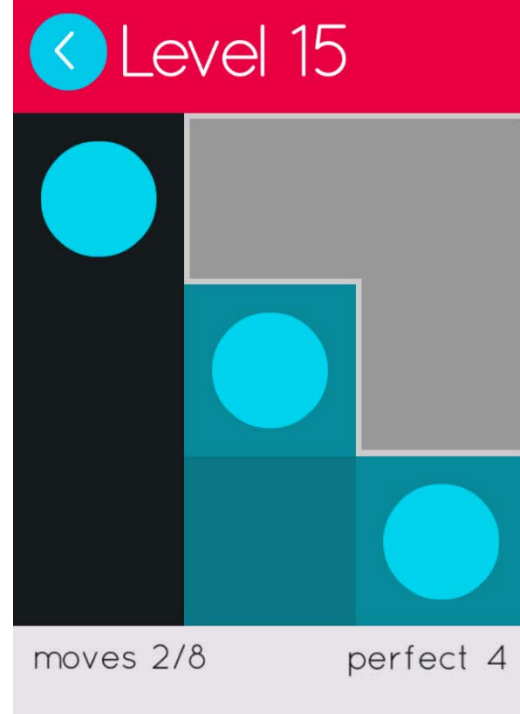
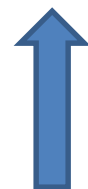
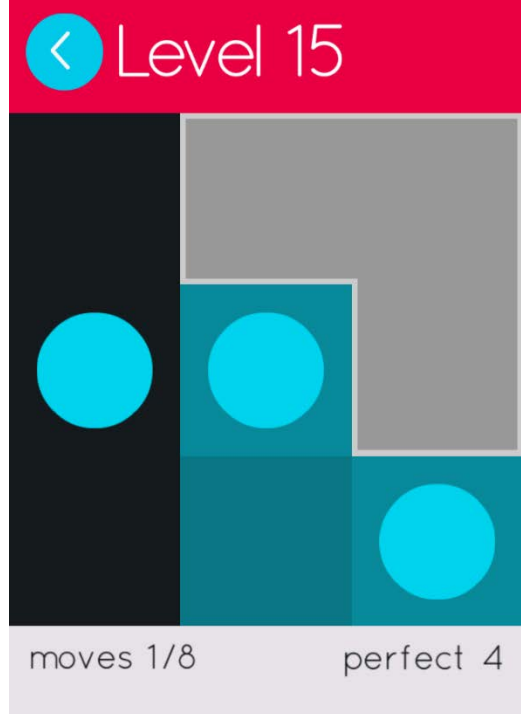
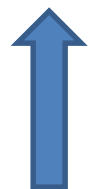
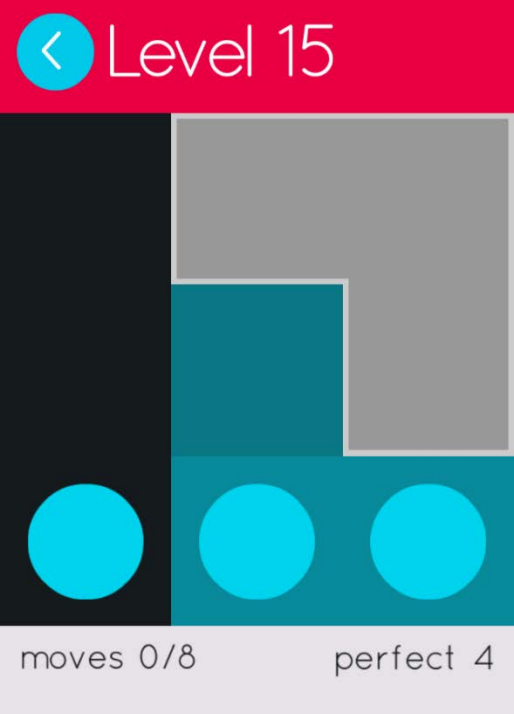
28

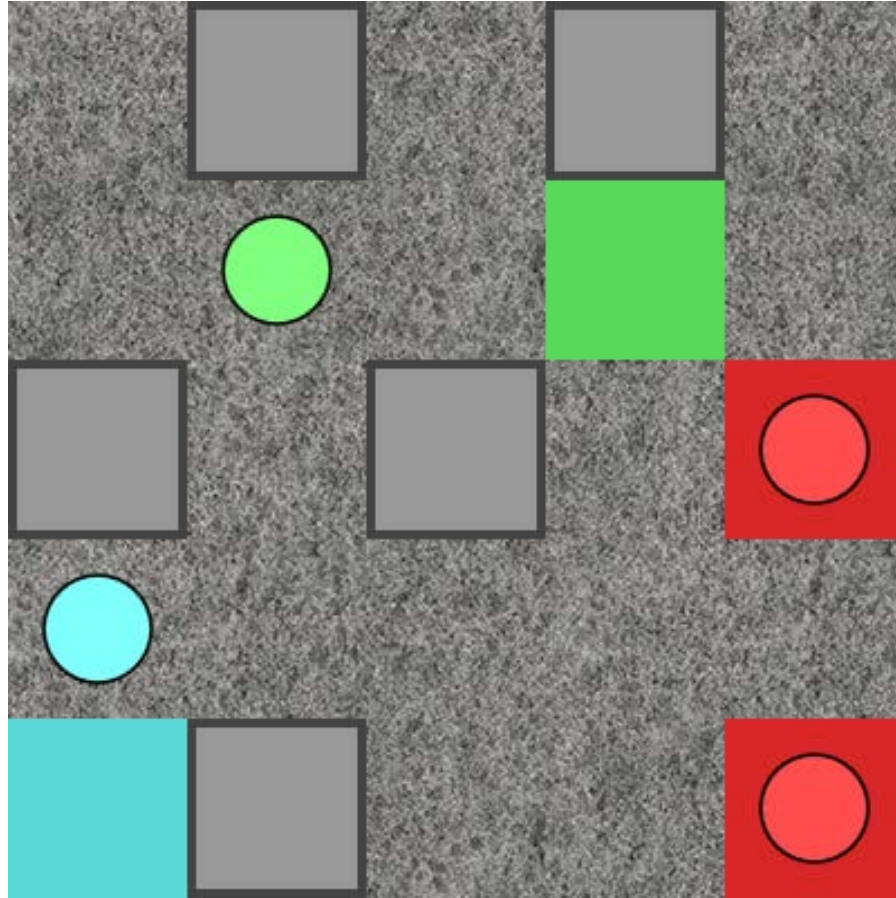
31

36

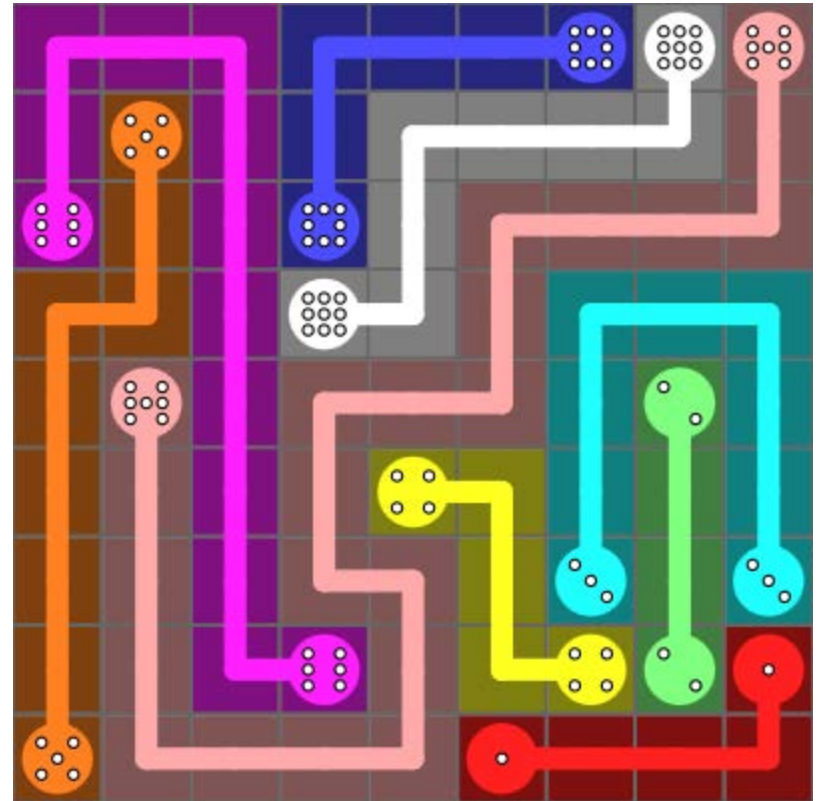
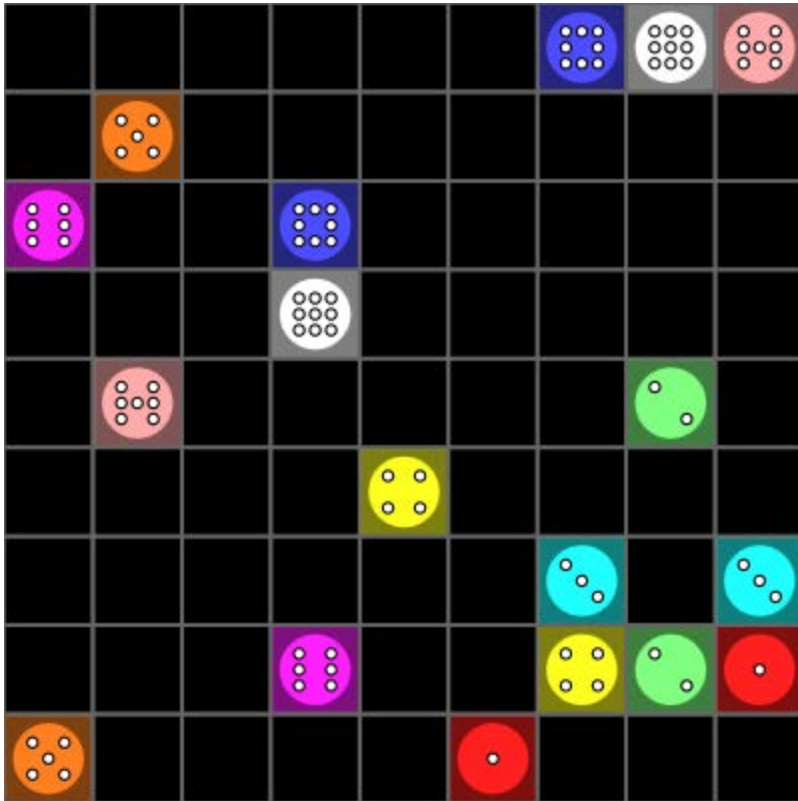
41

46

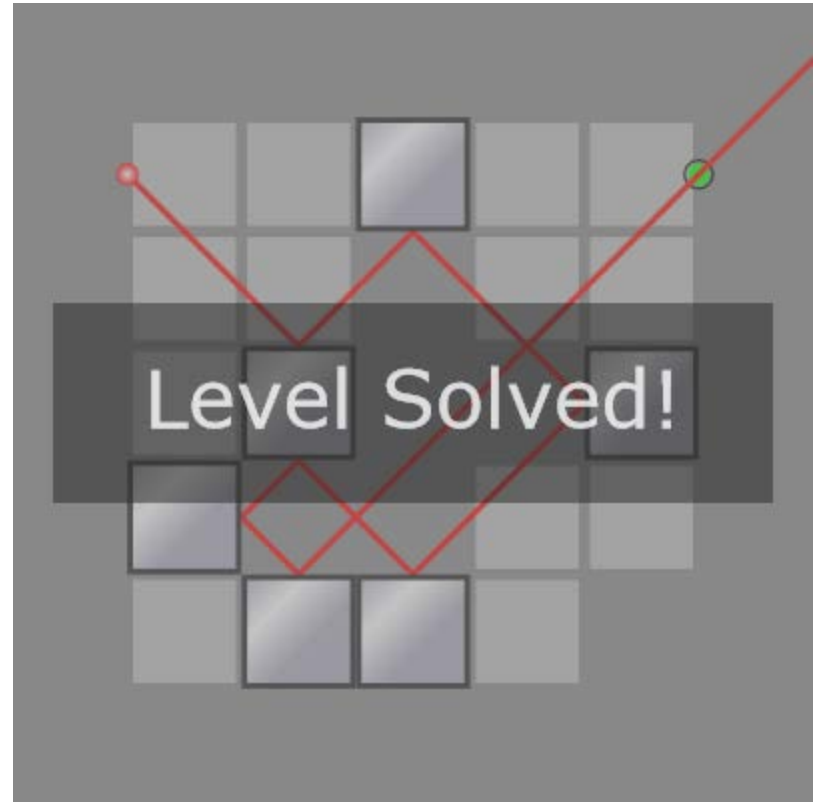
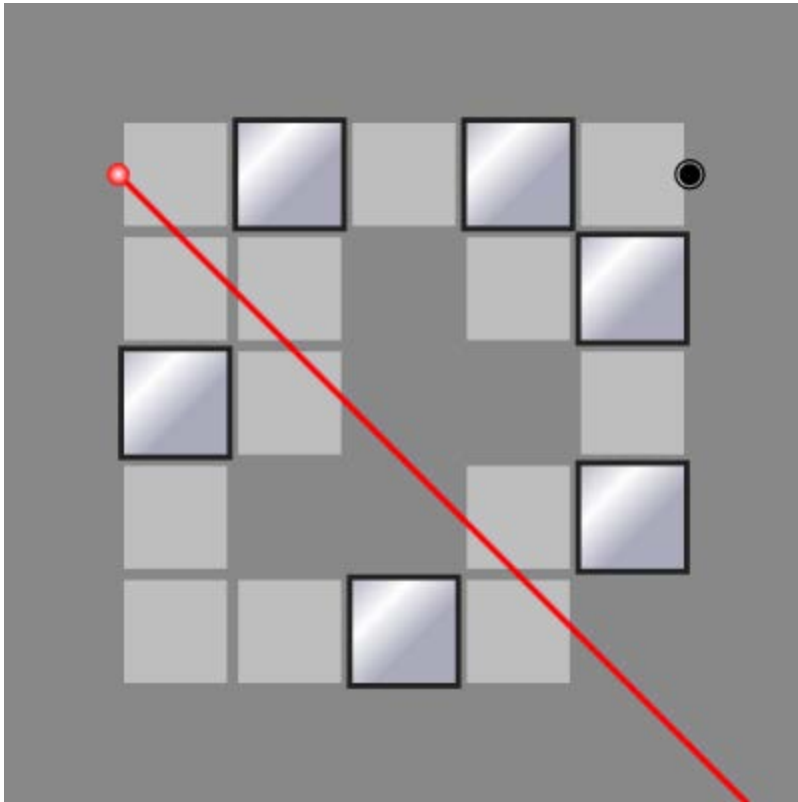




# Flow

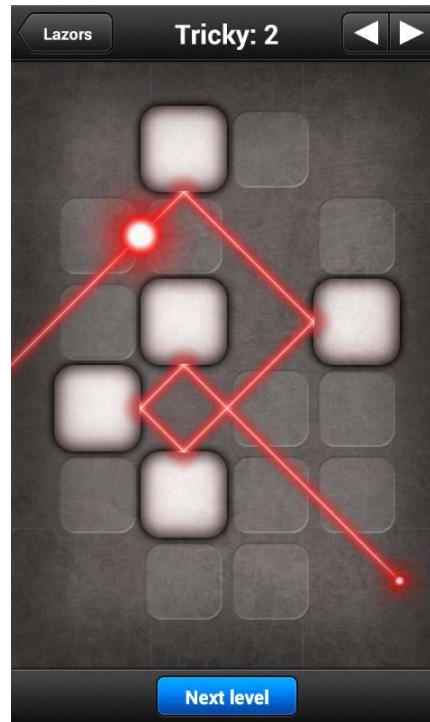


# Lazors

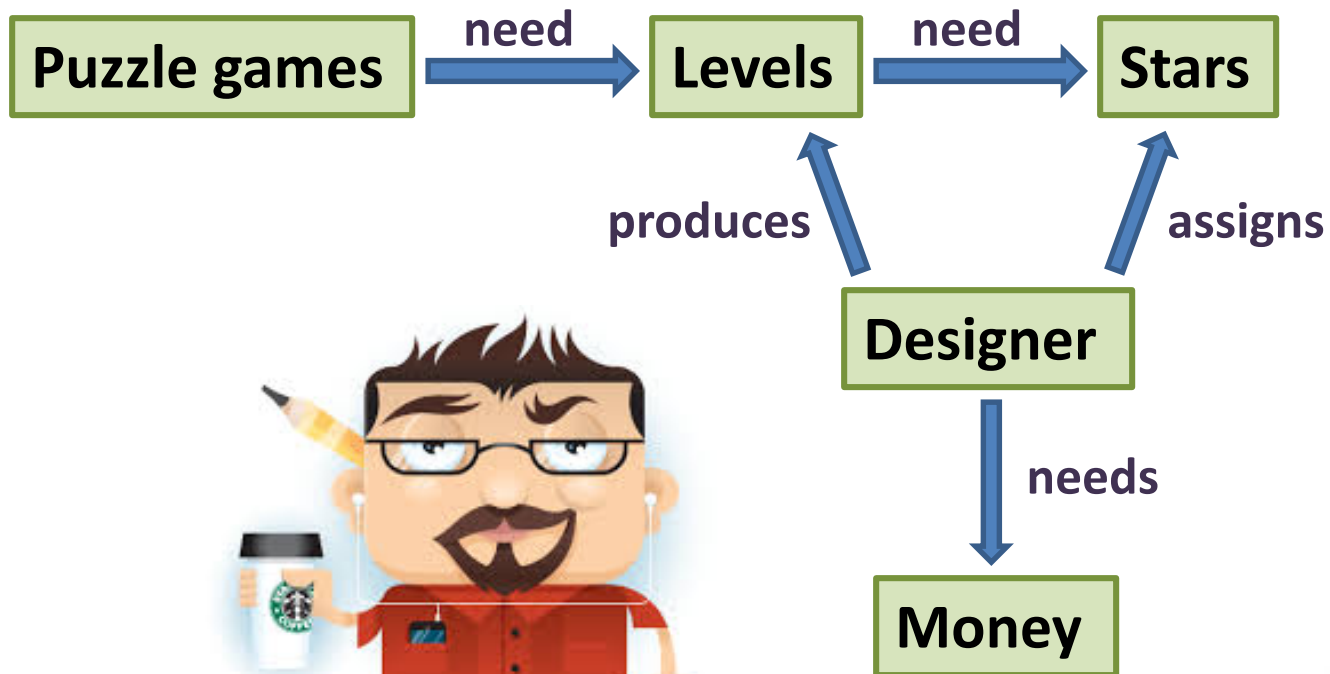




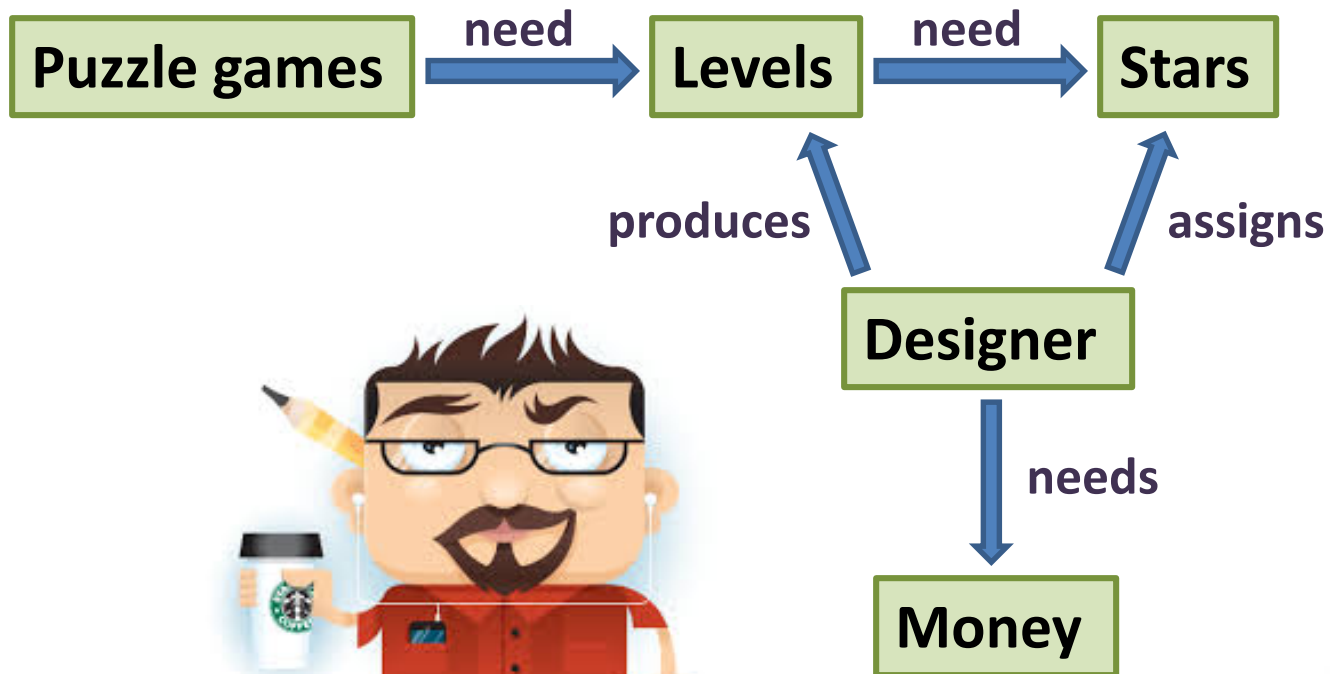
# Games and Levels



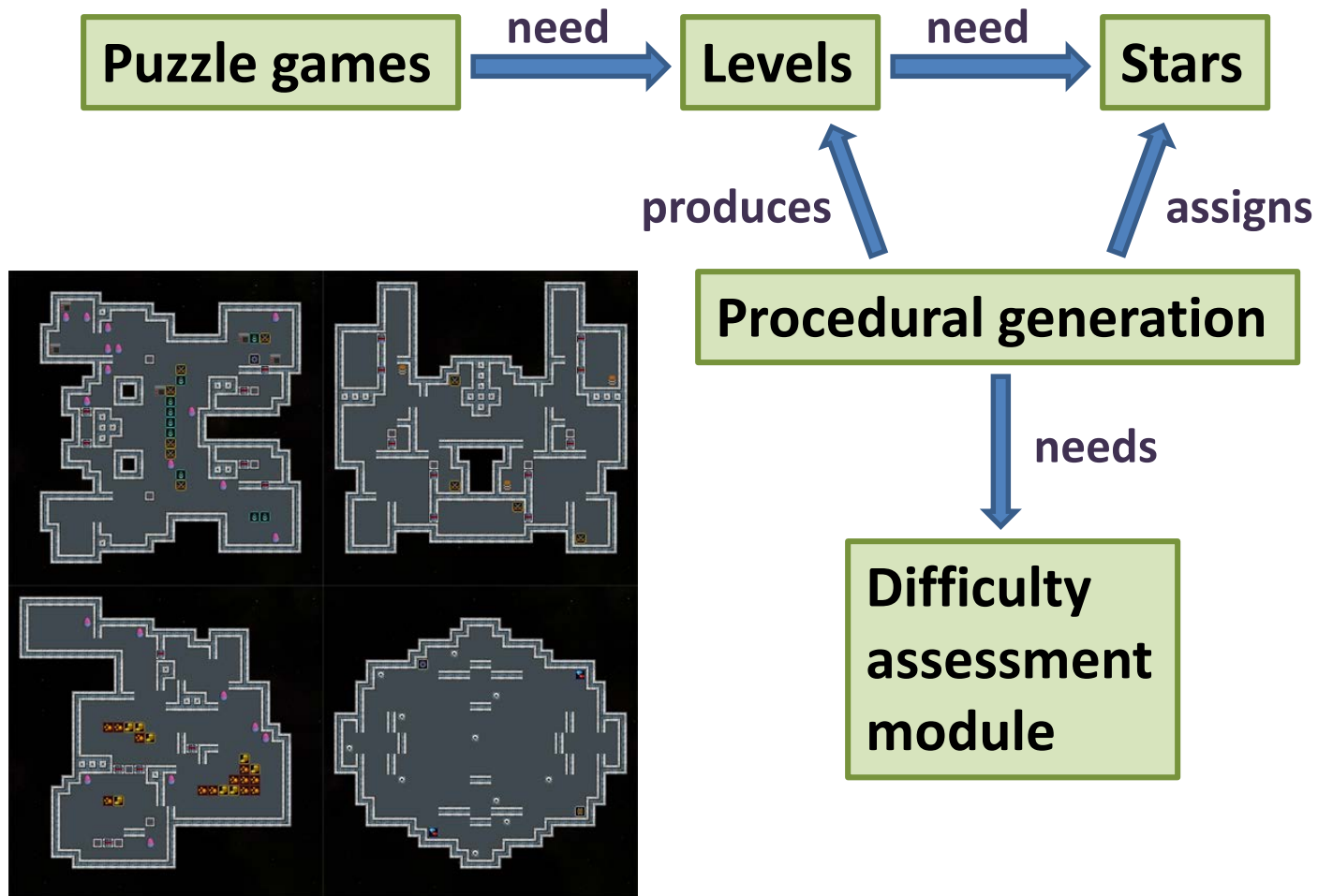
# Games and Levels



# Games and Levels

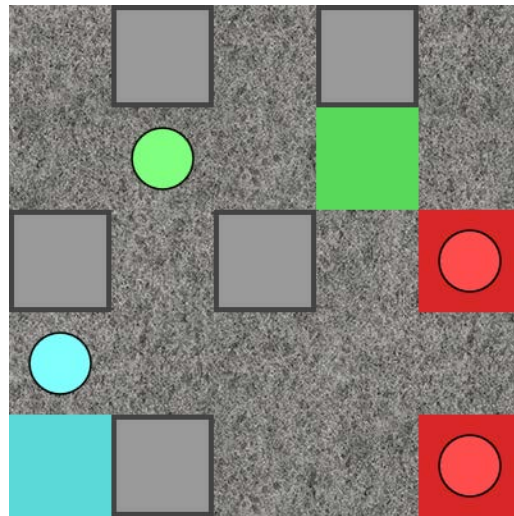


# Games and Levels



# Difficulty Assessment: Just a Function

$$f(\text{“level”}) \rightarrow \mathbb{R}$$



# Difficulty Assessment

Mantere and Koljonen (2007)

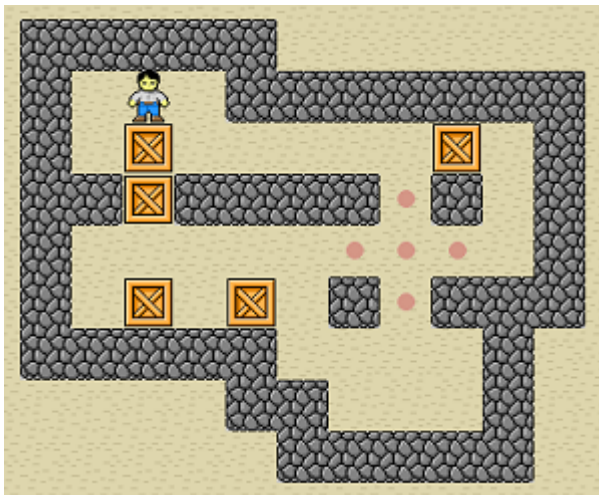
Ashlock and Schonfeld (2010)

Jarušek and Pelánek (2010)

Aponte, Devieux, and Natkin (2011)

András, Sipos, and Sóos (2013)

Guid and Bratko (2013)



*1. Measure **time** taken by **humans***

*2. Measure **time** taken by a **solver***

*3. Count **steps** to a solution*

*4. Use probability that a solution **attempt fails***

# Difficulty Assessment

Browne (2011):

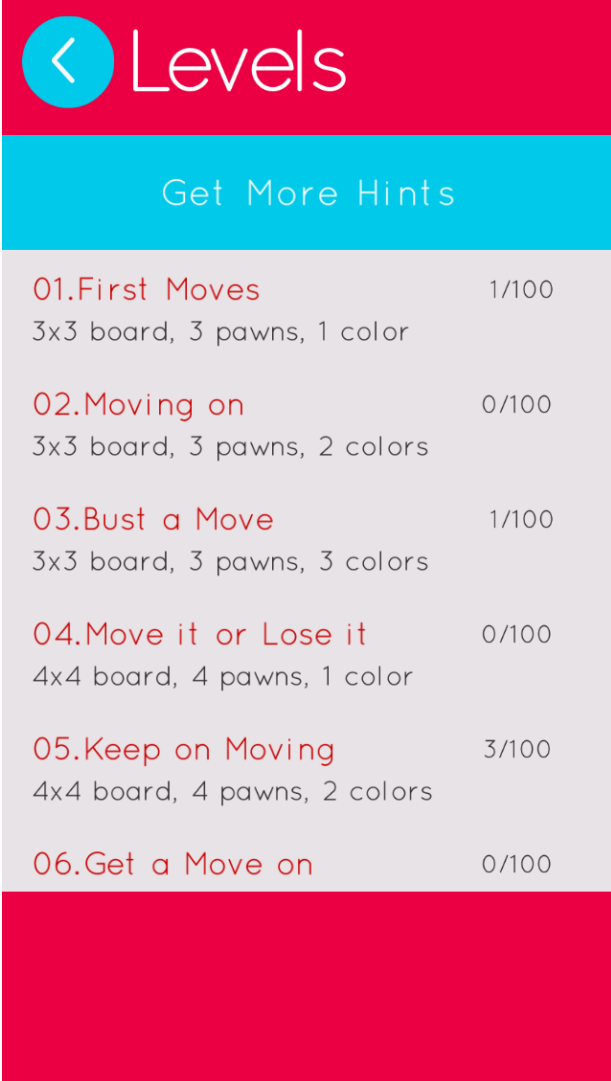
## Quantifying game quality

*Use linear function to combine  
game features*

This should also work for  
level difficulty:

## Quantifying level difficulty

Move

A screenshot of a game's levels menu. The title bar is red with a white back arrow and the word "Levels" in white. Below the title bar is a cyan button labeled "Get More Hints". The main content area is a list of levels with their names in red and progress indicators on the right. The background is light gray.

Level Name	Progress
01.First Moves 3x3 board, 3 pawns, 1 color	1/100
02.Moving on 3x3 board, 3 pawns, 2 colors	0/100
03.Bust a Move 3x3 board, 3 pawns, 3 colors	1/100
04.Move it or Lose it 4x4 board, 4 pawns, 1 color	0/100
05.Keep on Moving 4x4 board, 4 pawns, 2 colors	3/100
06.Get a Move on	0/100

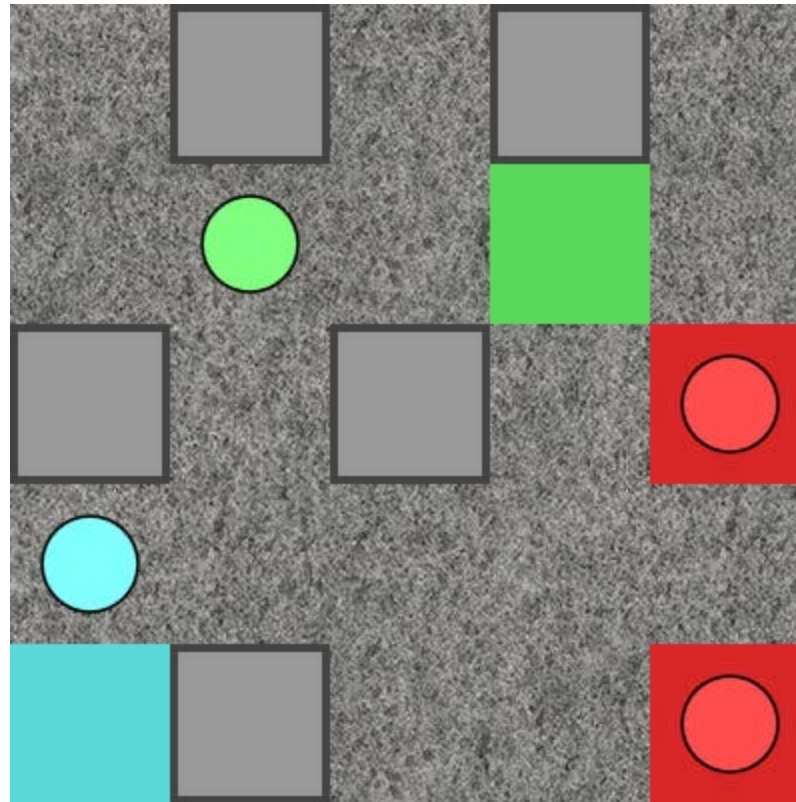
# Difficulty by Game Features

5x5 grid

5 blocks

4 balls

3 colors



not visible: 6 steps to the solution (moves)  
2 counter-intuitive moves



# Difficulty by Game Features

$$f(\text{“level”}) \rightarrow \mathbb{R}$$

$$\begin{aligned} f(\text{Move level}) = & W_1 * \text{grid-size} + W_2 * \text{balls} + \\ & W_3 * \text{colors} + W_4 * \text{blocks} + \\ & W_5 * \text{min-moves} + W_6 * \text{ci-moves} \\ & + W_0 \end{aligned}$$

$W_0, W_1, W_2, W_3, W_4, W_5$  and  $W_6$  are unknown weights

# Setting up the Difficulty Function

How do we get the weights?

How do we test whether a function exists that predicts the difficulty of a level well?

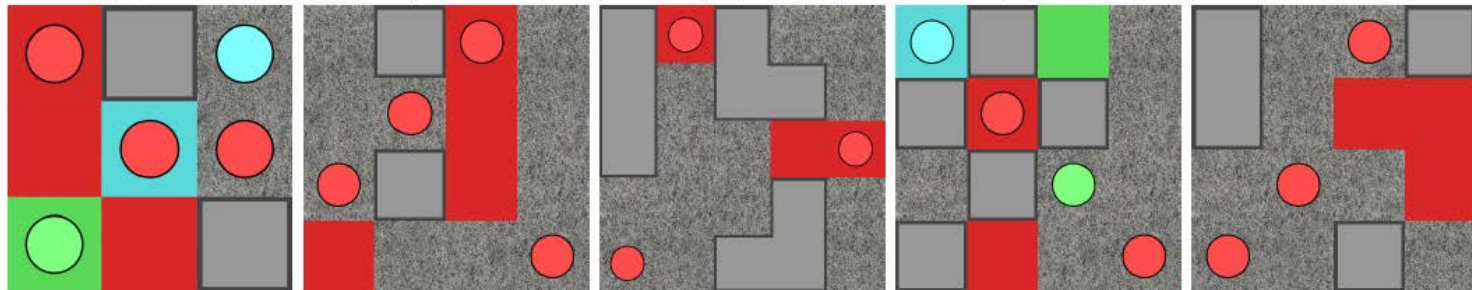


# Setting up the Difficulty Function

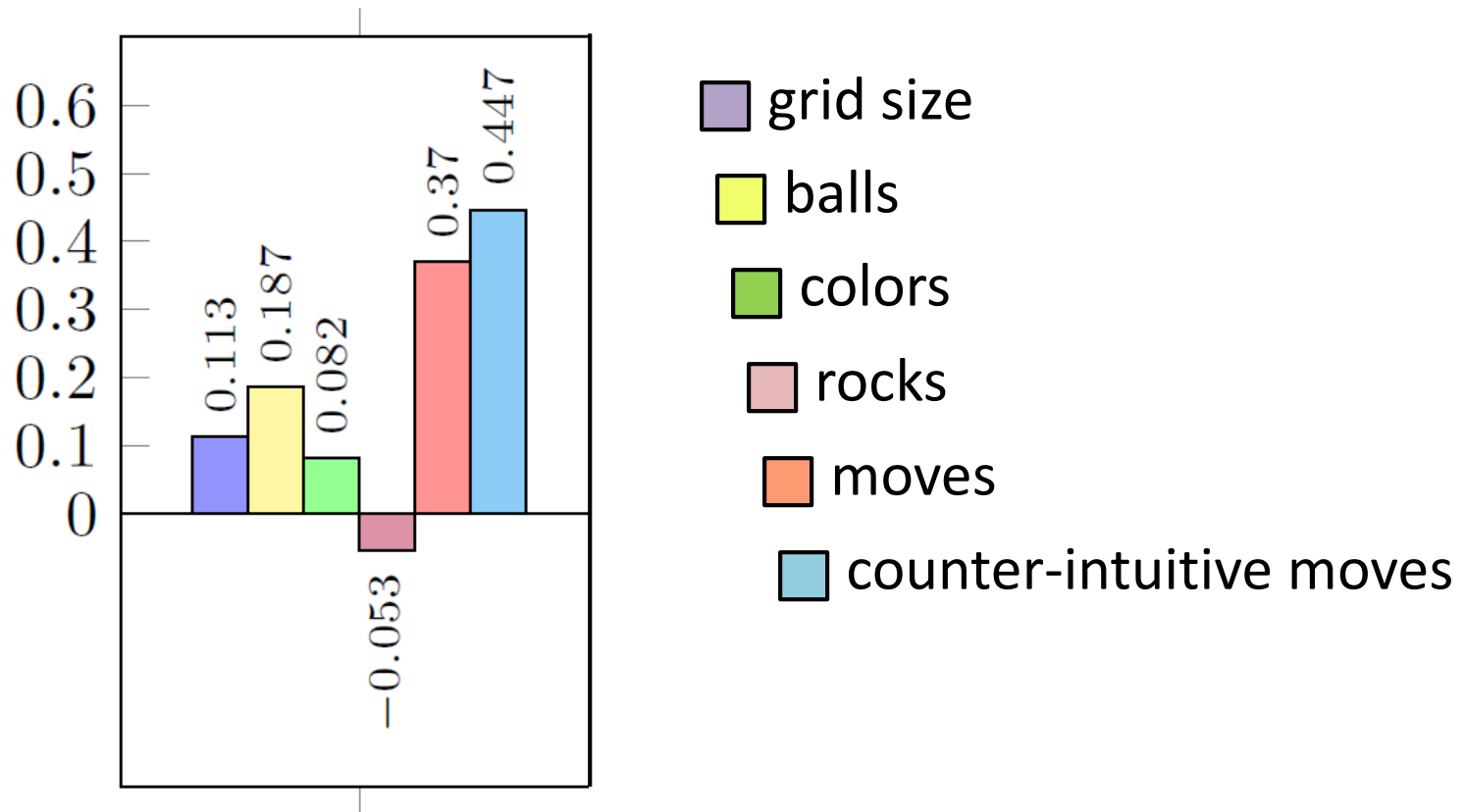
Web-based user-assisted difficulty rating:

*learn the game – play a level – rate its difficulty*

- 80 levels
- random selection and order for participants
- 6 or 7 ratings per level (57 different people)



# Correlation Results



# Setting up the Difficulty Function

$$f(\text{“level”}) \rightarrow \mathbb{R}$$

$$\begin{aligned} f(\text{Move level}) = & 0.06 * \text{grid-size} + 0.14 * \text{balls} + \\ & 0.16 * \text{colors} - 0.15 * \text{blocks} + \\ & 0.17 * \text{min-moves} + 0.46 * \text{ci-moves} \\ & - 0.19 \end{aligned}$$

with standard multiple linear regression

# Setting up the Difficulty Function

Web-based user study

- 80 levels
- random selection and order for participants
- 6 or 7 ratings per level (different people)

Analysis by **cross-correlation**: 5 groups of 16 levels

- get the weights (fit function) using 4 groups = 64 levels
- measure error on other group = 16 levels  
( error of level  $L$  is  $| \text{avg-rating } L - f(L) |$  )
- do this 5 times to measure all 80 levels → average error

# Difficulty Prediction Error

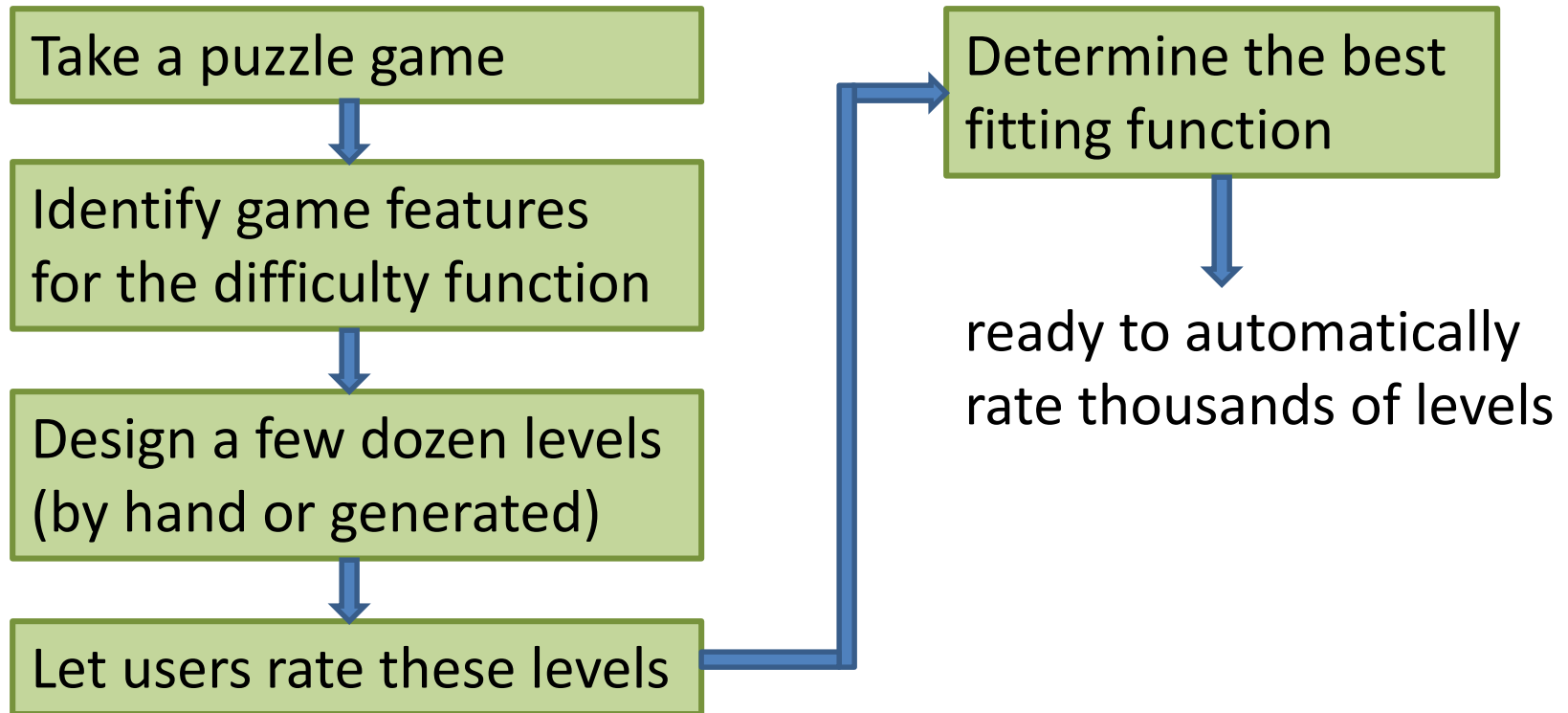
---

Rating scale is 1 – 10; average prediction error is **0.93** over the 80 levels

With the learned weights, any *new* level can be rated fully automatically with reasonable precision using  $f$

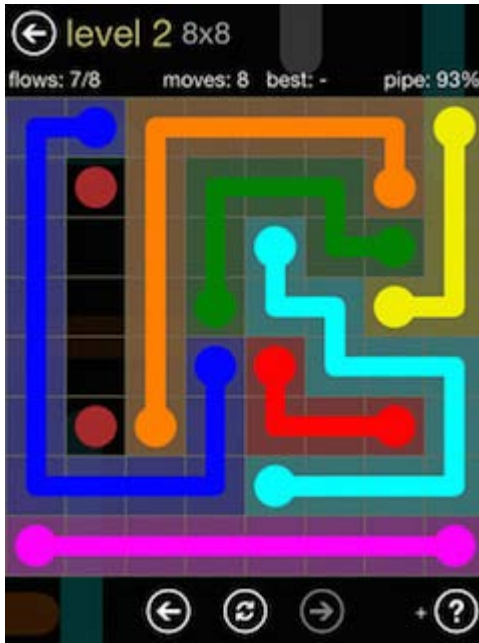
# The Level Difficulty Assessment Pipeline

---



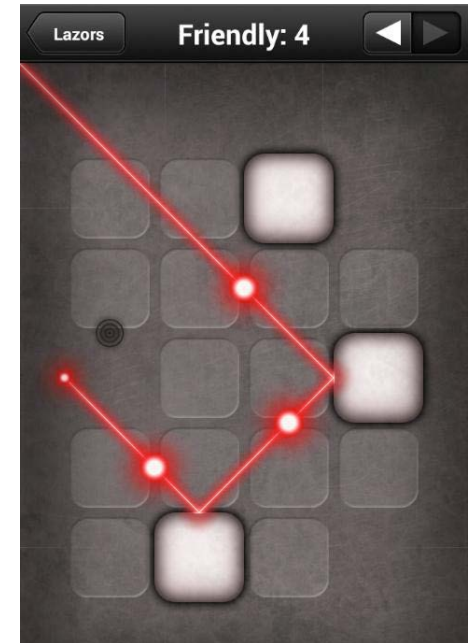


# More Results



flow

4 game features  
40 levels,  
played ~30 times each  
Average error **0.40**



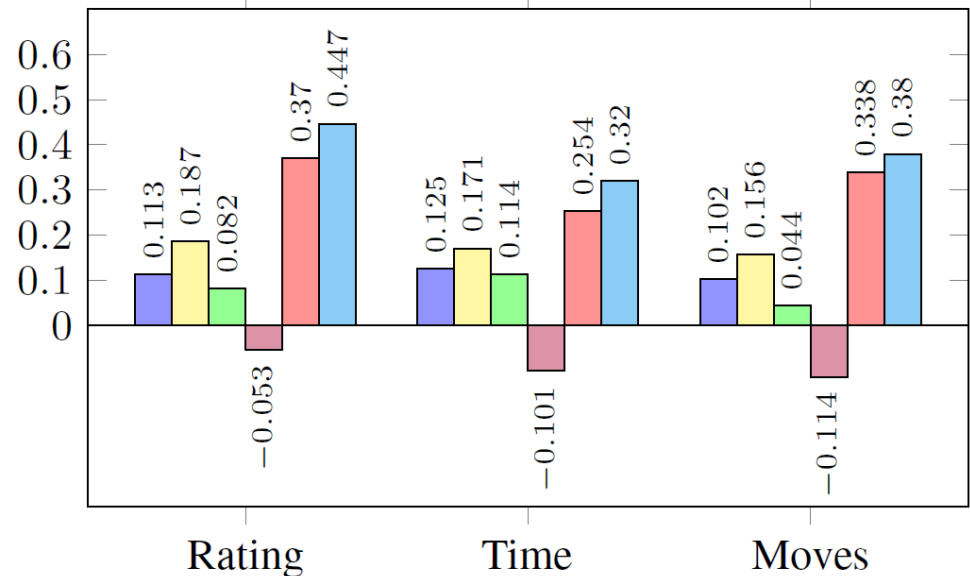
lazors

7 game features  
65 levels,  
played ~10 times each  
Average error **1.01**

# More Results

We also measured time taken by the users and number of interactions/moves done by the users

They also correlate with the difficulty ratings, but a function to predict them performs poorly (avg error 80% for time and 60% for #moves)



# Shortcomings

- We need to set up difficulty function anew for every puzzle game
- People are still needed, albeit not level designers
- Choice of game features requires feeling for the game
- Why would dependency on a game feature be linear? Why would game features be independent?
- It is likely that the approach will not work well for many puzzle games

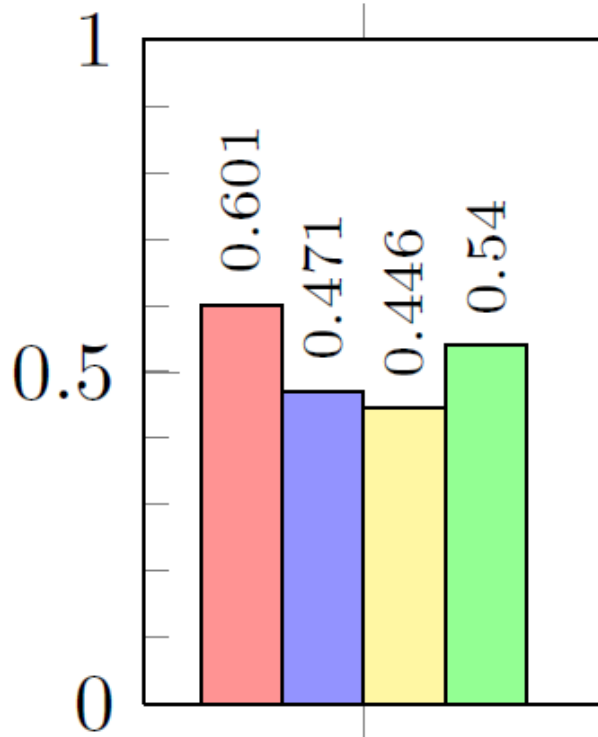


# Discussion

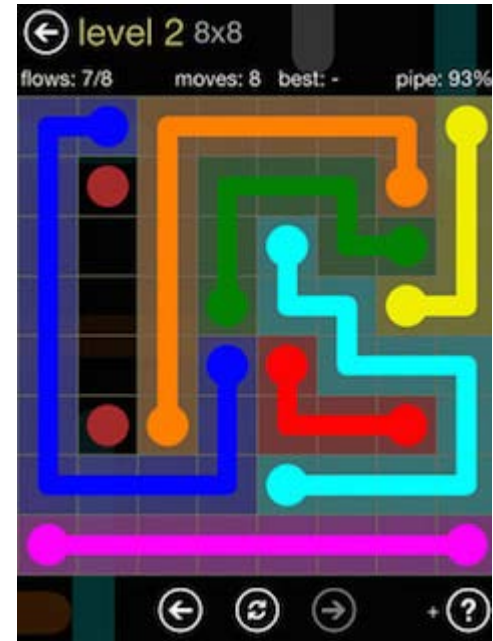
- Approach seems wrong for **physics-based puzzles** (Angry Birds, Cut the Rope, Cover Orange, etc.)  
How do we assess difficulty of levels of such puzzles?
- Can we say that a puzzle game is **more interesting** if the difficulty of its levels can be **less well predicted**?
- How can we make **level generation** benefit from knowing how difficulty correlates with game features?



# Correlation for Flow



- grid size
- colors
- distance
- turns



# Correlation for Lazors

