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
3x3 board, 3 pawns, 1 color

02. Moving on
3x3 board, 3 pawns, 2 colors

03. Bust a Move
3x3 board, 3 pawns, 3 colors

04. Move it or Lose it
4x4 board, 4 pawns, 1 color

05. Keep on Moving
4x4 board, 4 pawns, 2 colors

06. Get a Move on
0/100

This pack has 100 levels

03. Bust a Move

1-25

1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

16 17 18 19 20

21 22 23 24 25

26 27 28 29 30

31 32 33 34 35

36 37 38 39 40

41 42 43 44 45

46 47 48 49 50

51 52 53 54 55

56 57 58 59 60

61 62 63 64 65

66 67 68 69 70

71 72 73 74 75

76 77 78 79 80

81 82 83 84 85

86 87 88 89 90

91 92 93 94 95

96 97 98 99 100
Flow
Lazors
Games and Levels

Puzzle games need Levels need Stars

Lazors
© 2010-2013 Pyrosphere

Difficulty Rating Scale:
- Beginners
- Novice
- Intermediate
- Advanced
- Expert
Games and Levels

- Puzzle games
- Levels
- Stars

- need
- need
- produces
- assigns

- Money

Designer needs Money
Games and Levels

Puzzle games need Levels, which need Stars. Levels produce Designer, who needs Money. Money assigns Designer.
Games and Levels

Puzzle games → Levels → Stars

Procedural generation

Difficulty assessment module
Difficulty Assessment: Just a Function

\[ f \left( \text{“level”} \right) \rightarrow \mathbb{R} \]
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
Difficulty by Game Features

5x5 grid

4 balls

3 colors

not visible: 6 steps to the solution (moves)
2 counter-intuitive moves
Difficulty by Game Features

\[ f \left( \text{“level”} \right) \rightarrow \mathbb{R} \]

\[
f \text{ (Move level) } = W_1 \times \text{grid-size} + W_2 \times \text{balls} + \\
W_3 \times \text{colors} + W_4 \times \text{blocks} + \\
W_5 \times \text{min-moves} + W_6 \times \text{ci-moves} \\
+ W_0
\]

\( W_0, W_1, W_2, W_3, W_4, W_5 \) 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

- Grid size
- Balls
- Colors
- Rocks
- Moves
- Counter-intuitive moves
Setting up the Difficulty Function

\[ f\left(\text{“level”}\right) \to \mathbb{R} \]

\[ f\left(\text{Move level}\right) = 0.06 \times \text{grid-size} + 0.14 \times \text{balls} + 0.16 \times \text{colors} - 0.15 \times \text{blocks} + 0.17 \times \text{min-moves} + 0.46 \times \text{ci-moves} - 0.19 \]

with standard multiple linear regression
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
  \[
  \text{error of level } L = | \text{avg-rating } L - f(L) | \]
- do this 5 times to measure all 80 levels \(\rightarrow\) 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

- Take a puzzle game
- Identify game features for the difficulty function
- Design a few dozen levels (by hand or generated)
- Let users rate these levels
- Determine the best fitting function
- ready to automatically rate thousands of levels
More Results

flow

4 game features
40 levels,
   played ~30 times each
Average error \textbf{0.40}

lazors

7 game features
65 levels,
   played ~10 times each
Average error \textbf{1.01}
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

- level size
- usable tiles
- emitters
- receivers
- mirrors
- reflections
- intersections