The 2010 Mario AI Championship
Learning, Gameplay and Level Generation tracks
WCCI competition event

Sergey Karakovskiy, Noor Shaker,
Julian Togelius and Georgios Yannakakis
How many of you saw the paper about the 2009 Mario AI Competition yesterday?
What was the Mario AI Competition?

- A competition based on “Super Mario Bros”...
- ...designed to test and rank AI methods for game playing...
- ...where researchers submitted their best Mario-playing controllers...
- ...and the best Mario AI won?
Why bother?

• Problem faced (voluntarily) by hundreds of millions of gamers around the world since 1985

• Games are designed to challenge human cognitive skills and learning abilities

• Could help improve game design/development (e.g. for PCG)

• Similarity to robotics problems etc.
Competition objectives

• Ease of participation
• Transparency
• Ease of finding a winner
• Depth of challenge
Infinite Mario Bros

- by Markus Persson
- quite faithful SMB 1/3 clone
- in Java
- random level generation
- open source
Making a benchmark

- Control loop rewritten
- Tunable FPS, up to 1000 times faster than real-time
- Removed stochasticity
- Created and interface for agents/controllers
Last year’s winner

- Robin Baumgarten of Imperial College, UK
- Approach based on A*
- Goal: get to the right edge of the screen
- Internal model of the game’s physics
A* IN MARIO: CURRENT POSITION

Goal: right border of screen

current node
A* IN MARIO: CHILD NODES
A* IN MARIO: BEST FIRST

current node

right, speed
A* IN MARIO: BACKTRACK

current node

right, jump, speed

right, speed
A* IN MARIO: BEST FIRST

current node

right, jump, speed

right, speed
A* IN MARIO: CREATE CHILDREN

Current node

Friday, September 11, 2009
Sunday, July 25, 2010
Some takeaways from last year

- A* plays this type of levels very well, but not in a human-like manner
- The task did not test learning
- And was too easily beaten
The 2010 Mario AI Championship

- An attempt to...
- ...deepen the challenge: *Gameplay track*
- ...test learning as well as controller design: *Learning track Cancelled for this event*
- ...test the capabilities of CI in game design: *Level generation track Demo only for this event*
The 2010 Mario AI Championship

• Gameplay track
  Same as last year, except that the toughest levels are much tougher.

• In particular: dead ends that force backtracking, meant to be lethal for A*

• Learning track
  Allows the controller 1000 runs on each level to allow it to learn the level; scored on the 1001st attempt
The 2010 Mario AI Championship

- Level generation track
  Competitors submit level generators that output *fun* levels for particular players, based on measurements of playing style.
  Live judging!

(We need you! *Do you have Java installed?*)
The 2010 Mario AI Championship

• EvoStar (April, Istanbul): Gameplay track
• WCCI (July, Barcelona): Gameplay, Learning and Level Generation tracks
• CIG (August, Copenhagen): Gameplay, Learning and Level Generation tracks
• ICE-GIC (December, Hong Kong): Turing Test track
Gameplay track

Sergey Karakovskiy and Julian Togelius
Agent goals

• Develop an agent that gets as far and as fast as possible...
• ...on as many levels as possible...
• ...which are previously unseen
• Scoring: progress on 40 randomly generated levels (of different difficulty, length, type) with seed 17564
• If two agents complete all the levels: tiebreakers
Challenges

- Handle a large state/observation space
- Handle very different situations (now more different than before)
- Tactical tradeoffs (e.g. go back and get the power-up or continue forward?)
Interface
Environment Interface

- 22x22 arrays describing
  - landscape features (e.g. walls, cannons, gaps)
  - creatures
  - Fine position of Mario and creatures
  - Booleans: mario is on the ground, may jump, is carrying a shell, is small/big/fire
Mario AI in a nutshell

Score: 13998.645
Levels cleared = 9
Total time left = 6780
Total Kills = 87
Mario mode = 32
TOTAL SUM = 20906.645

Your Agent

22x22 byte arrays
observation
float[] positions
(float[] rewards
action
(0, 1, 0, 1, 1)

benchmark outputs

Sunday, July 25, 2010
Very simple agent

```
public boolean[] getAction(Environment observation)
{
    action[Mario.KEY_SPEED] = action[Mario.KEY_JUMP] =
        observation.isMarioAbleToJump() ||
        !observation.isMarioOnGround();
    return action;
}
```
Differences from last year

• Framework more developed, now with better support for non-Java agents (e.g. Python)

• Tighter bound on time taken/action

• More difficult levels!
  • Level generator augmented to generate really hard levels on higher difficulties
  • Some levels are impossible
Evaluation setup

- total episodes: 126
- Main score: distance passed
- Tie-breakers: speed, creatures killed, mode
- all 3 types of levels, fixed seed, difficulties = \{0, 1, 3, 5, 12, 16, 20\}
- 42 ms per action (violating the limit results in disqualification for the entire level)
- tweaked JVM run to skip the GC executions.
Results
Evaluation parameters

- Seed 429766
- 576 trials
- Remember: a controller is disqualified on a level if it takes more than 42 ms (real time) in any frame
<table>
<thead>
<tr>
<th>Team</th>
<th>Members</th>
<th>Score</th>
<th>Disc</th>
<th>Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robin Baumgarten</td>
<td></td>
<td>1537834</td>
<td>312</td>
<td>A*</td>
</tr>
<tr>
<td>Sergey Polikarpov</td>
<td>Robert Reynolds, Leonard Kinnaird-Heether, Tracy Lai</td>
<td>1637935</td>
<td>1</td>
<td>CyberNeuron (RL)</td>
</tr>
<tr>
<td>wSUMariO-CAT</td>
<td></td>
<td>1113437</td>
<td>0</td>
<td>Elman network / cultural algorithm</td>
</tr>
<tr>
<td>Alexander Buck</td>
<td></td>
<td>991372</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>Eamon Wong</td>
<td></td>
<td>972341</td>
<td>0</td>
<td>Q-learning</td>
</tr>
<tr>
<td>Mathew Erickson</td>
<td></td>
<td>634239</td>
<td>0</td>
<td>Genetic Programming</td>
</tr>
</tbody>
</table>

Sunday, July 25, 2010
Winner of the Gameplay track: Sergey Polikarpov
What can we learn?

- A* is not invincible!
- At least not *on its own*
- All entrants fail at dead ends, but fail in different ways
- We probably need to combine micro- and macro-strategy
- Still time left to compete at CIG event
Learning track

Sergey Karakovskiy and Julian Togelius
Please compete!

- The interface is almost as simple as for the GamePlay track (and almost the same)
- Allows you 1000 runs to optimize your controller
- Scores you on the 1001st
- Too few entrants for the WCCI event
- Still time left to compete...

Sunday, July 25, 2010
Level generation track

Noor Shaker, Julian Togelius and Georgios Yannakakis
Submit a level generator that generates fun levels, personalized for individual players.

The levels generated also has to adhere to constraints to force the generation of diverse levels.
The process

• Each judge plays a test level, and his performance on this level is recorded (various metrics such as jumps, deaths etc.)

• Each level generator generates a level tailored to each judge

• The judges play “their own” levels and rank them
Setup
Interface

GamePlay.java contains information about the level design and how the testers played.

```java
public int totalEnemies; //total number of enemies
public int totalEmptyBlocks; //total number of empty blocks
public int totalPowerBlocks; //total number of power blocks
public int totalCoins; //total number of coins
public int GreenTurtlesKilled; //number of Green Turtle Mario killed
public int ArmoredTurtlesKilled; //number of Armored Turtle Mario killed
public int GoombasKilled; //number of Goombas Mario killed
public int timeRunningLeft; //total time spent running to the left
public int emptyBlocksDestroyed; //number of empty blocks destroyed
public int coinsCollected; //number of coins collected
```

Interface

• LevelInterface.java provides a simple interface that should be implemented when constructing your customized level:
  public byte[][][] getMap();
  public SpriteTemplate[][][] getSpriteTemplates()

• Constructed levels should communicate with the simulation only through the LevelGenerator interface:
  public LevelInterface generateLevel (GamePlay playerMetrics);
• All submitted levels should satisfy the constraints defined in the Constraints interface. Example:

```java
public static int levelWidth = 320;
public static int gaps = 10;
public static int turtles = 7;
public static int coinBlocks = 10;
```
Instructions

• Download the jar file
• Run it (double click)
• Play the first level (controls A, S, arrows)
• Play two additional levels
• Rank the levels: most and second most fun
• Turn your laptops towards me!
• Scoring: 2 point for most fun level, 1 for second most fun
Scores:

- Random: 12
- Optimized: 7
- Nathan: 20
- Demo winner: Nathan Sorenson
Optimized gaps level

- Refinement of the original Infinite Mario level generator
- Placement, number and width of gaps generated using a model based on preference learning from 240 players
- Also forthcoming AIIDE paper (Shaker et al)
Nathan Sorenson

- Simon Fraser University
- Level generator based on GA (high level) and constraint solver (low level)
http://www.marioai.org

http://groups.google.com/group/mariocompetition/