FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

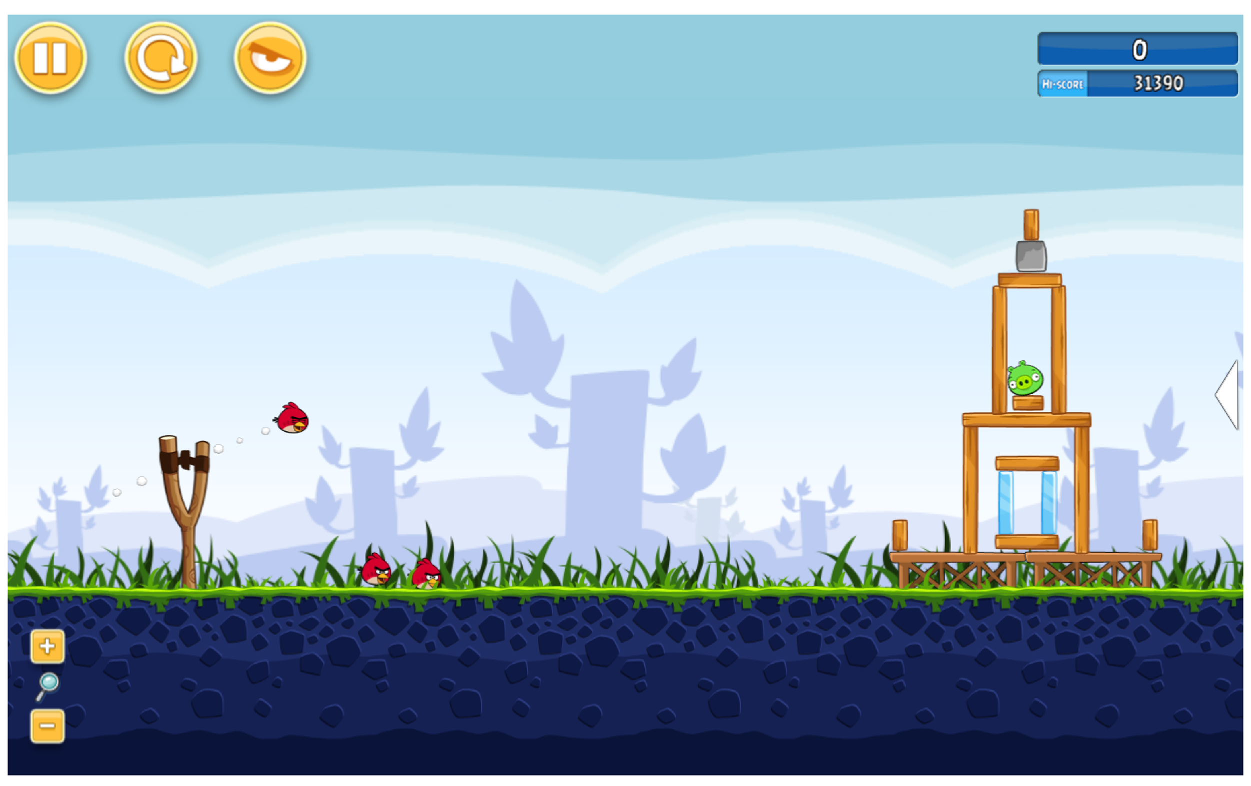
Università della Calabria

AngryHEX: An Angry Birds-playing Agent based on HEX-Programs

Francesco Calimeri Michael Fink Stefano Germano Andreas Humenberger Giovambattista Ianni
Christoph Redl Daria Stepanova Andrea Tucci

1. Motivation

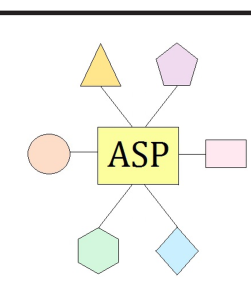
- **Angry Birds** (`http://www.angrybirds.com`) is a strategic arcade video game where the player uses a slingshot to shoot a limited number of birds at constructions aiming to destroy all pigs in the field



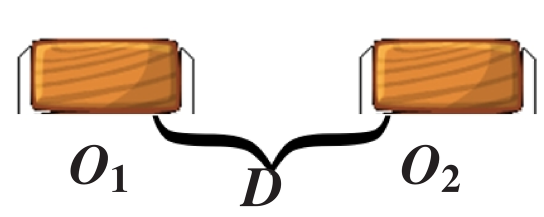
- **Goal:** Construct a declarative agent which plays the game
- **Challenge:** Plan optimal shots under consideration of physics
- **Our means:** HEX-programs, i.e., Answer Set Programming (ASP) with external sources and other extensions

2. HEX-Programs

- HEX-programs extend ASP by external sources
- Rule bodies may contain external atoms, e.g.



  - $\&distance[O_1, O_2](D)$ is true iff distance between O_1 and O_2 is D
  - $\&canpush[ngobject](O_1, O_2)$ is true iff O_1 can push O_2 given additional info in extention of $ngobject$



Example

- Estimate likelihood that object O_2 falls when object O_1 is hit

$$\begin{aligned} \mathbf{r1}:\ & pushDamage(O_2, P_1, P) \leftarrow pushDamage(O_1, \_, P_1), P_1 > 0 \\ & \&canpush[ngobject](O_1, O_2), \\ & pushability(O_2, P_2), P = P_1 * P_2/100. \end{aligned}$$

3. Architecture of our Agent

- We use the provided framework (browser plugin, vision module etc.)
- Agent builds on tactic and strategy both realized declaratively
- **Tactic:** reasoning about the next shot is done in a HEX-program Π
  - **Input**: scene info from the vision module (facts of Π)
  - **Output**: desired target (extracted from models of Π)
- **Strategy:** next level to played is computed in an ASP program Π'
  - **Input:** info about the number of times levels were played, best scores achieved, scored of our agent, etc. encoded as facts
  - **Output:** next optimal level to be played

4. HEX-Encoding for Angry Birds Tactic

- **Physics simulation results** are accessed via external atoms, e.g.,
  - decide whether O_1 falls whenever O_2 falls
  - decide which O' intersect with trajectory of a bird after hitting O
  - compute distances between O_1 and O_2
  - ...
- **Tactic in details:**
  - Consider each shootable target (objects which have a direct and unobstructed path from the slingshot)
  - Compute the estimated damage on each non-target object (discrete values), taking different bird types into account
  - Rank the targets (=answer sets) using weak constraints: add malus points for each pig, where the number of added malus points decreases with increasing likelihood that the pig is destroyed
  - Learn from history: never play a level in the same way more then once, look for new shots

5. ASP encoding for Angry Birds Strategy

- **Strategy in details:**
  - First play each level once
  - Then play levels in which our score maximally differs from the best
  - Play levels in which we played best and the difference to the second best score is minimal

5. Preliminary Benchmark Results

Level	ABC-AI	ABC-IS	HEX(2013)	HEX(2014)[n]
level 1	27550	30490	32090	31540 [7]
level 2	52420	34600	53460	44330 [10]
level 3	33460	41070	42370	41910 [6]
level 4	18690	27990	27970	**28520** [18]
level 5	36280	62780	63300	**69260** [13]
level 6	17870	17500	34810	34890 [9]
level 7	22510	20560	45710	45690 [11]
level 8	47400	40440	38730	**57070** [11]
level 9	35600	42500	43160	**51560** [12]
level 10	41530	43970	55660	55000 [9]
Sum	333310	361900	437260	**459770**

- ABC-AI: plain ASP
- ABC-IS: procedural implementation
- HEX(2013): HEX at IJCAI'13
- HEX(2014): HEX as by 17.08.14
- $[n]$: number of runs (strategy)

6. Results and Outlook

- **Results:**
  - Agent is realized using declarative programming means
  - New vision module provided by the organizers is integrated
  - Declarative strategy is realized (used to be in java)
  - Fixes and improvements
- **Possible improvements:**
  - Combine objects which behave like a single one
  - Plan over multiple shots

7. References

- Eiter, T., Ianni, G., Schindlauer, R., and Tompits, H. (2006). Effective Integration of Declarative Rules with External Evaluations for Semantic-Web Reasoning. ESWC'06 volume 4011, pages 273–287.
- Angry Birds AI competition Benchmark (`http://aibirds.org/benchmarks.html`)

FWF Der Wissenschaftsfonds.