Perkun – a new AI algorithm

www.vampisoft.com

Perkun

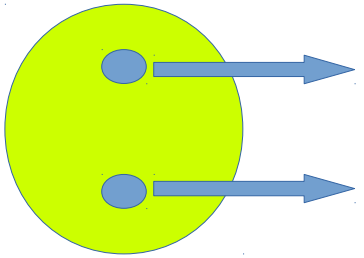
IF THEN vs. Perkun optimizer

- IF THEN is stateless
- Ignores history

- Perkun optimizer has a state (belief)
- Belief is updated
- Maximizes E(payoff)
- Builds a game tree

Hidden variables

- Introduced to generate belief
- A visible state is a vector of input variable values
- Belief is a probability distribution
- Belief domain is a collection of states built upon a visible state
- A state is a vector of hidden variable values



What Perkun needs to know

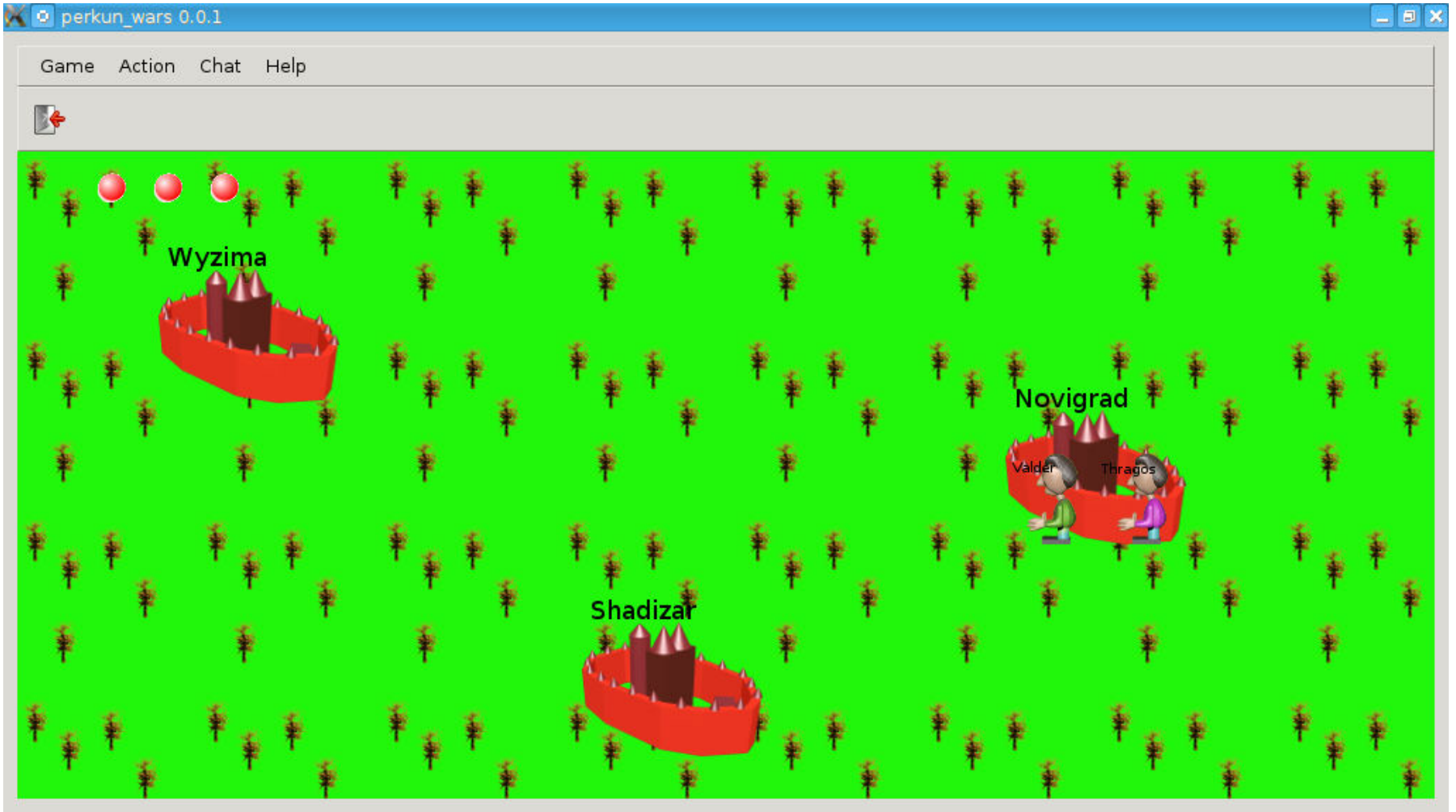
- Values
- Variables
  - Input variables
  - Output variables
  - Hidden variables
- Payoff function
- Model
- Game tree height

What Perkun can do

- Predict result probability
- Calculate result interpretation (new belief)
- Find optimal decisions
  - Performing actions
  - Performing experiments/asking questions

PerkunWars – hunting a vampire

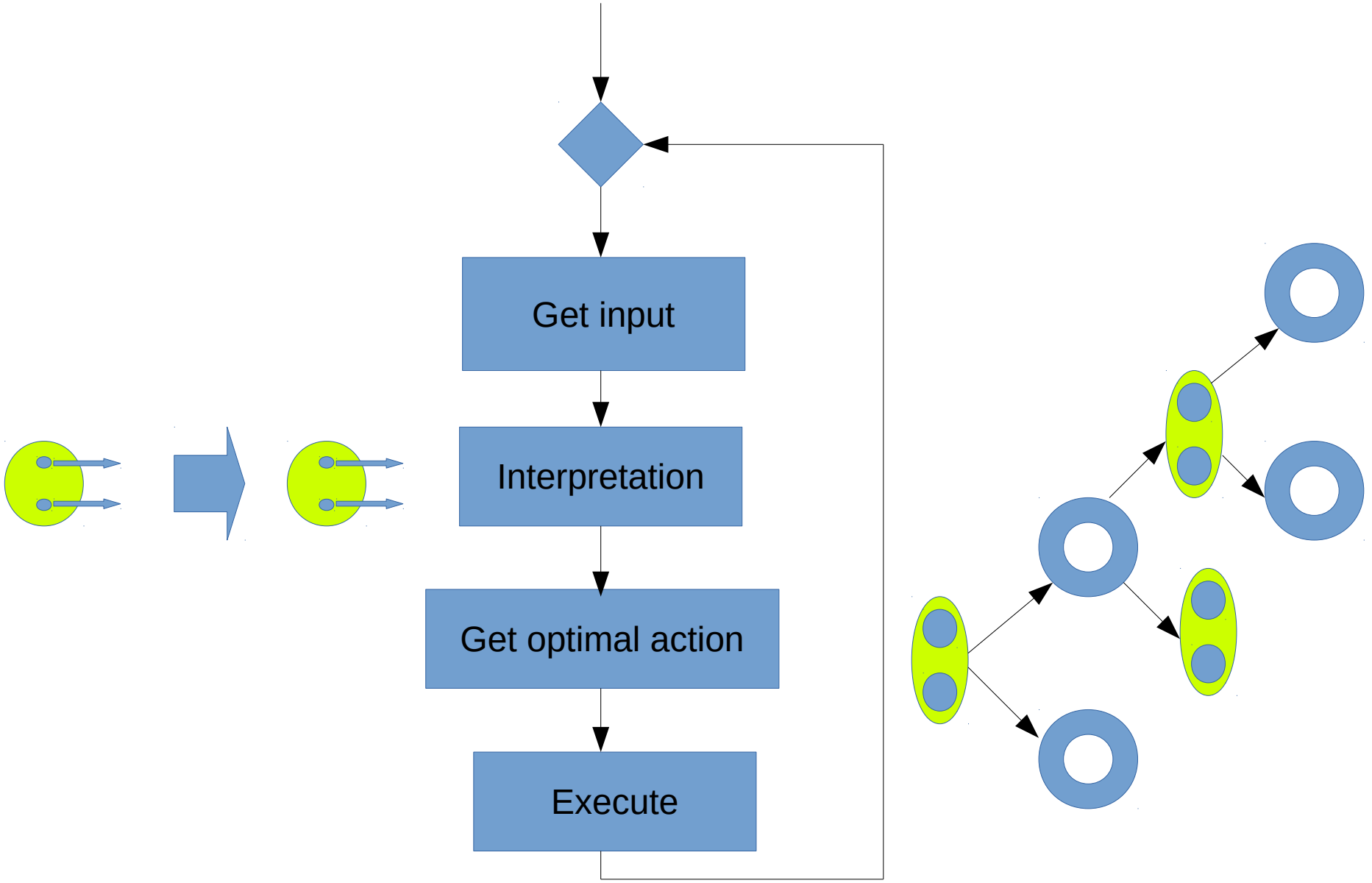
- https://sourceforge.net/projects/perkunwars/



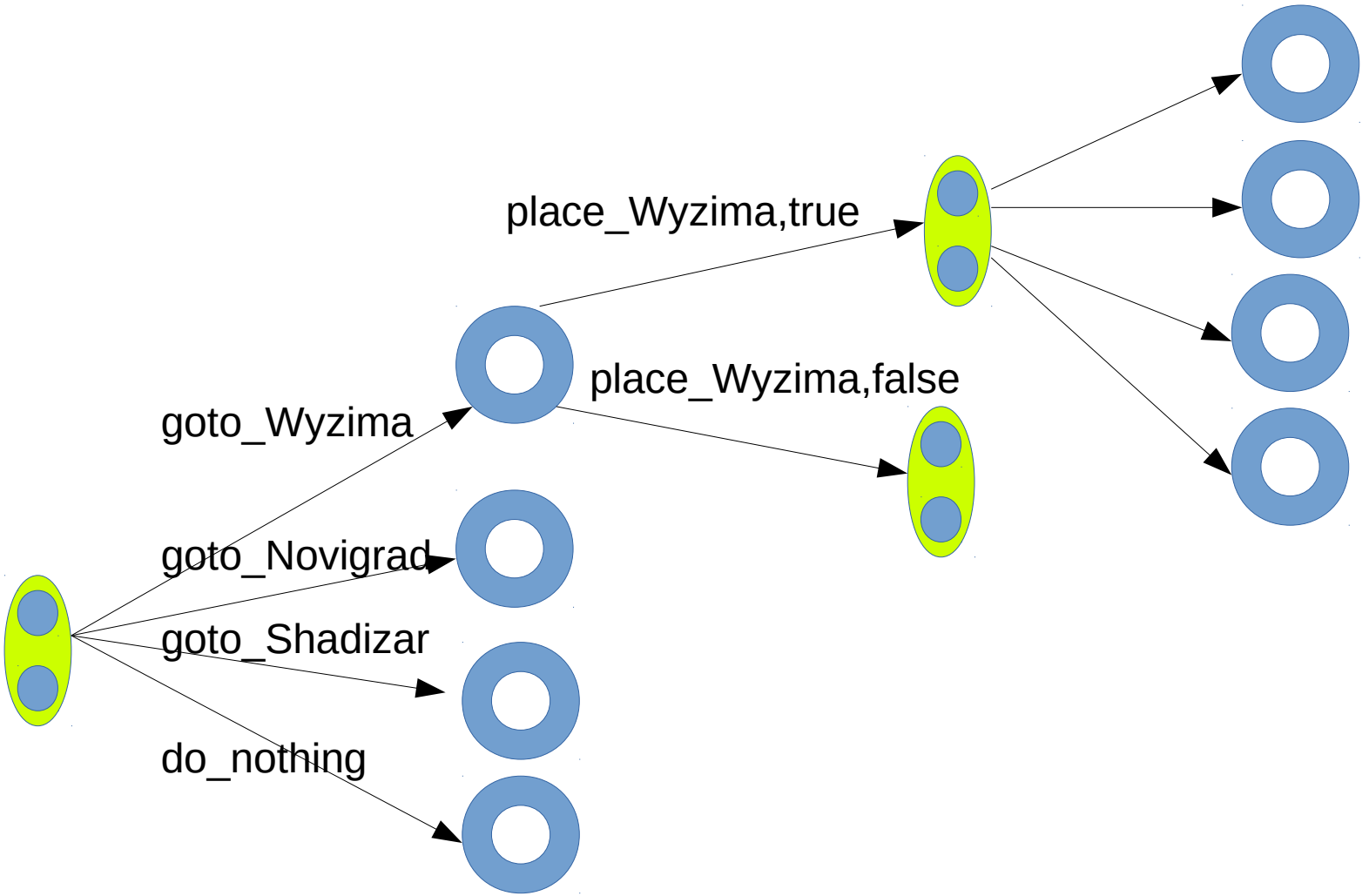
Running perkun session

- cd perkun-0.1.7/examples
- perkun example3_pregor.perkun
  - place_Wyzima true
  - place_Shadizar false
- perkun example2_dorban.perkun
  - place_Wyzima false
  - place_Shadizar false
  - place_Novigrad true

Optimization algorithm



Game tree


place_Wyzima,true
place_Wyzima,false
goto_Wyzima
goto_Novigrad
goto_Shadizar
do_nothing

Command line tools

- Perkun
- Wlodkowic
- Zubr

Perkun specification

values { … }

variables

{

input variable where_is_Dorban:{place_Wyzima,place_Shadizar,place_Novigrad};

input variable do_I_see_vampire:{false,true};

output variable action:{do_nothing,goto_Wyzima,goto_Shadizar,goto_Novigrad};

hidden variable where_is_vampire:
{place_Wyzima,place_Shadizar,place_Novigrad};

}

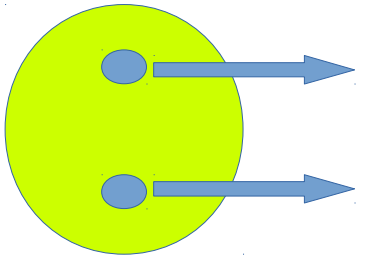
payoff {...}

model {...}

loop(1);

Belief – the optimizer state

- where_is_vampire=place_Wyzima where_is_Dorban=place_Wyzima do_I_see_vampire=false 0.00000
- where_is_vampire=place_Shadizar where_is_Dorban=place_Wyzima do_I_see_vampire=false 0.500000
- where_is_vampire=place_Novigrad where_is_Dorban=place_Wyzima do_I_see_vampire=false 0.500000



Optimal action

- goto_Wyzima
- goto_Shadizar
- goto_Novigrad
- do_nothing

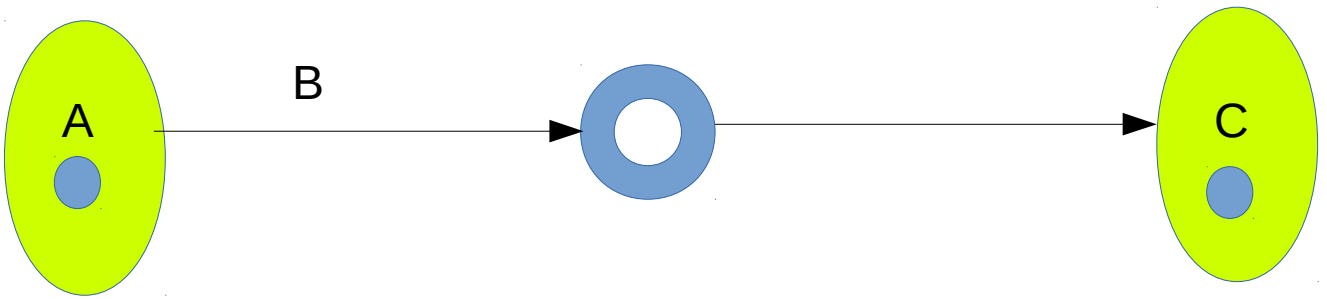
Pregor vs. Dorban

```
payoff
{
set({where_is_Pregor=>place_Wyzima,
do_I_see_vampire=>false}, 100.0);
set({where_is_Pregor=>place_Wyzima,
do_I_see_vampire=>true}, 0.0);
set({where_is_Pregor=>place_Shadizar,
do_I_see_vampire=>false}, 100.0);
set({where_is_Pregor=>place_Shadizar,
do_I_see_vampire=>true}, 0.0);
set({where_is_Pregor=>place_Novigrad,
do_I_see_vampire=>false}, 100.0);
set({where_is_Pregor=>place_Novigrad,
do_I_see_vampire=>true}, 0.0);
}
```

```
payoff
{
set({where_is_Dorban=>place_Wyzima,
do_I_see_vampire=>false}, 0.0);
set({where_is_Dorban=>place_Wyzima,
do_I_see_vampire=>true}, 100.0);
set({where_is_Dorban=>place_Shadizar,
do_I_see_vampire=>false}, 0.0);
set({where_is_Dorban=>place_Shadizar,
do_I_see_vampire=>true}, 100.0);
set({where_is_Dorban=>place_Novigrad,
do_I_see_vampire=>false}, 0.0);
set({where_is_Dorban=>place_Novigrad,
do_I_see_vampire=>true}, 100.0);
}
```

Model

- Illegal actions
- Impossible states
- Transition probabilities
  - set(A,B,C,D);



Interesting achievment

- Game tree depth > 1
- Experiments – actions with neutral payoff differentiating the beliefs
- Actions using the knowledge gathered
- Depending on the belief – decisions to ask questions

Links

- http://www.pawelbiernacki.net/software/perkun
- http://pawel-biernacki.blogspot.fi/
- https://sourceforge.net/projects/perkunwars/
- https://sourceforge.net/projects/thragos/