EPSRC
Engineering and Physical Sciences Research Council

iGGi

Your future in games research

Game AI Group

General Video Game AI Tutorial

—— www.gvgai.net ——

QMULGRADFEST

Raluca D. Gaina

19 February 2018

Who am I?

Raluca D. Gaina

2nd year PhD Student
Intelligent Games and Games Intelligence (IGGI)

r.d.gaina@qmul.ac.uk
rdgain.github.io
@b_gum22

PhD Topic: Rolling Horizon Evolutionary Algorithms in General Video Game Playing

Why am I here doing this?

~~My supervisor forced me~~
GVGAI is cool!
The new QMUL Games AI group is cool too!

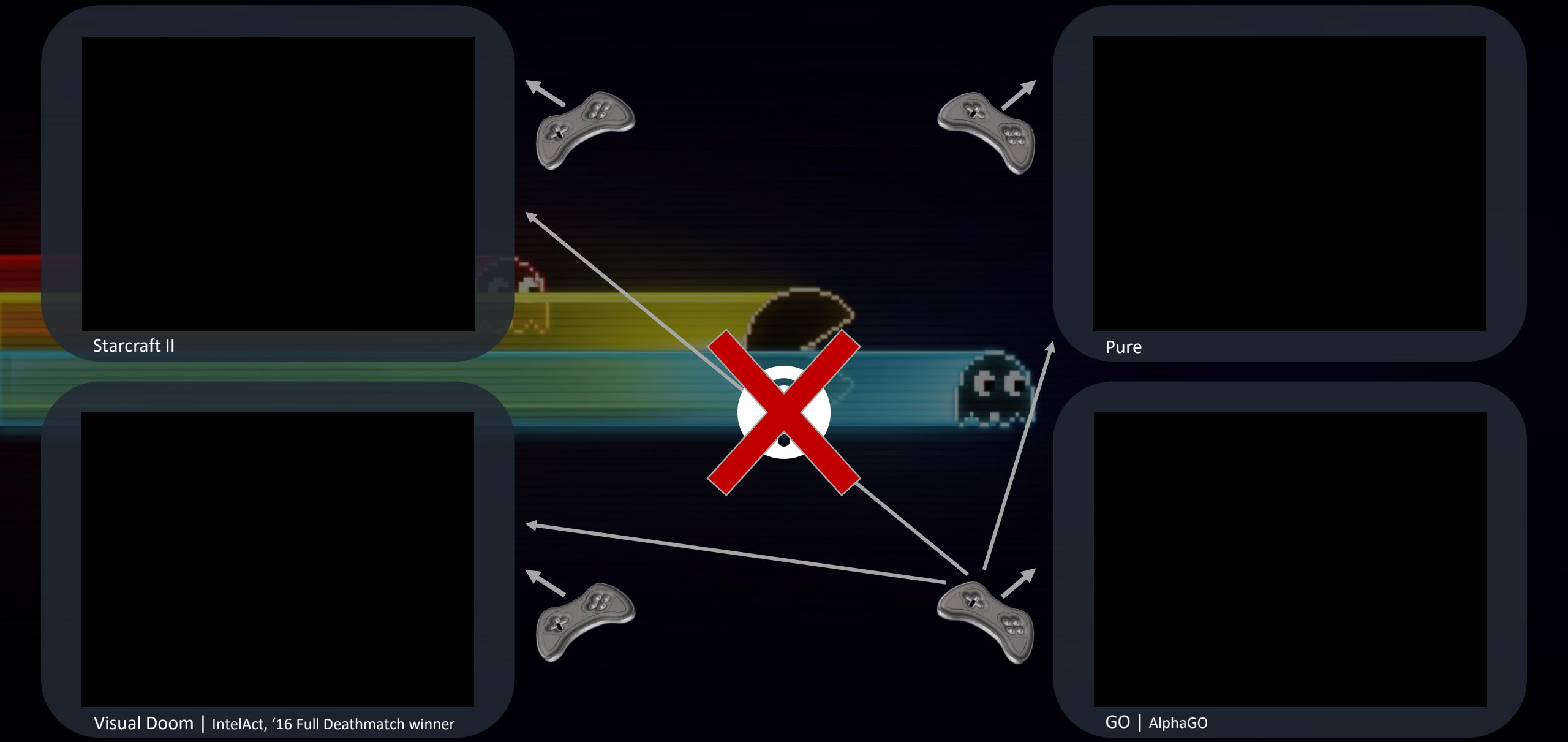
Schedule

11:00 - 11:15	Arrival & coffee
11:15 - 12:00	Introductory talk
12:00 - 12:30	Lunch
12:30 - 14:30	Practical
	Submissions to private league
14:30 - 15:00	Final evaluation private league
15:00	Winners announced

Outline

- Introduction - General Game Playing
- General Video Game AI Research
- GVGAI Framework and Competition
- Sample agents
- Guidelines for practical
- Prizes!

Research on games



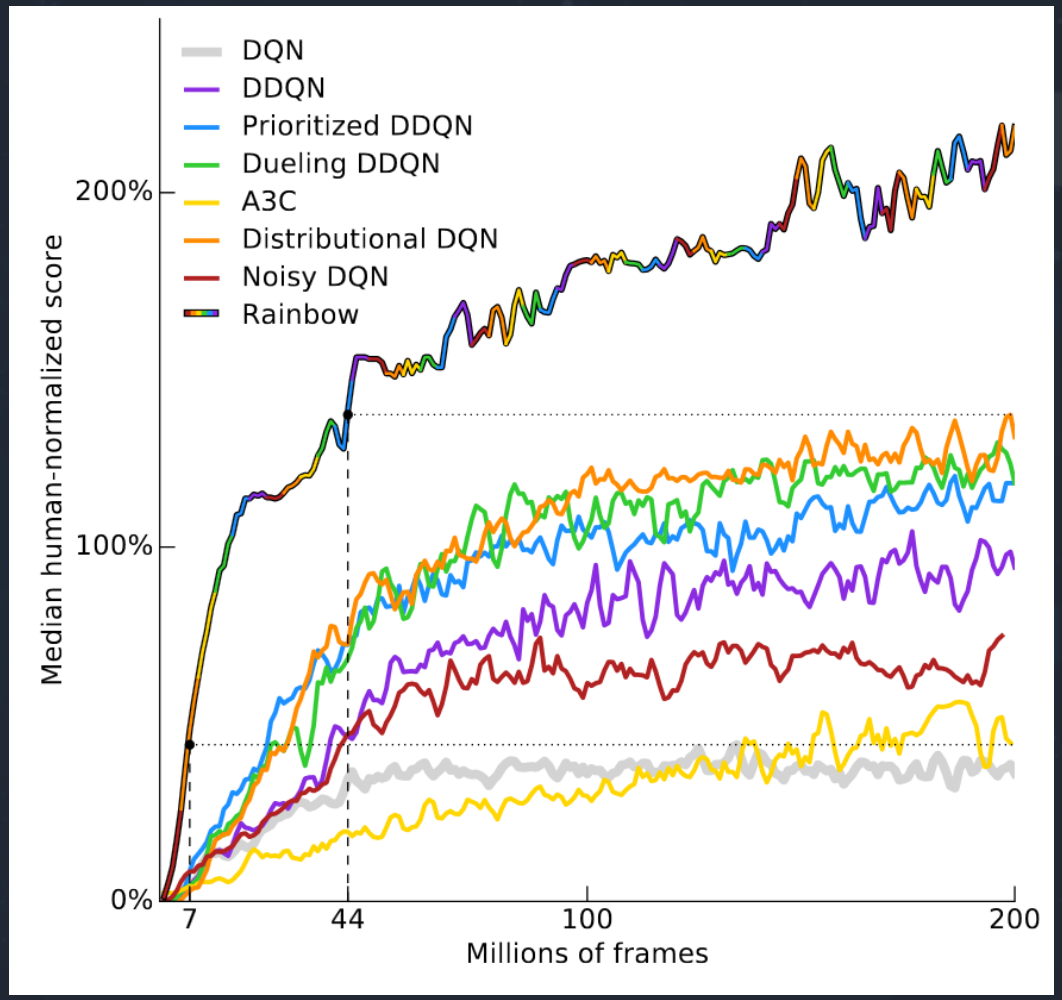
Starcraft II

Pure

Visual Doom | IntelAct, ‘16 Full Deathmatch winner

GO | AlphaGO

General game playing



m/2018/02/14/rl-hard.html

Peter Keevash and Liana Yepremyan, "Rainbow matchings in properly-coloured multigraphs" [1]

General video game playing

One agent plays all (real-time, arcade) games

Arcade Learning Environment (ALE),
Evaluation of AI agents in 55 games of the Atari 2600 Collection

Levine et al. propose the creation of a new benchmark for GVGP [Levine et al., 2013]

Compliments ALE in two ways:

- Creation of games in a more general framework.
- No screen capture analysis needed, information via encapsulated objects.

Video Game Description Language (VGDL) [Tom Schaul, 2013]

- Benchmark for learning and planning problems.
- Base for the GVGAI Framework

Believable Characters in Video Game AI

- Believability Assessment
- Believable Agents

General Video Game AI

- GVGAI -> A competition for game-playing agents, level and rule generators
- Agents are tested on unseen videogames
- 160+ games
- Games are implemented in the Video Game Description Language

Real time -> 40ms as budget time (per action)

- If returns action in 40-50ms
  - NIL is applied
- If returns action after 50ms
  - disqualification and game loss

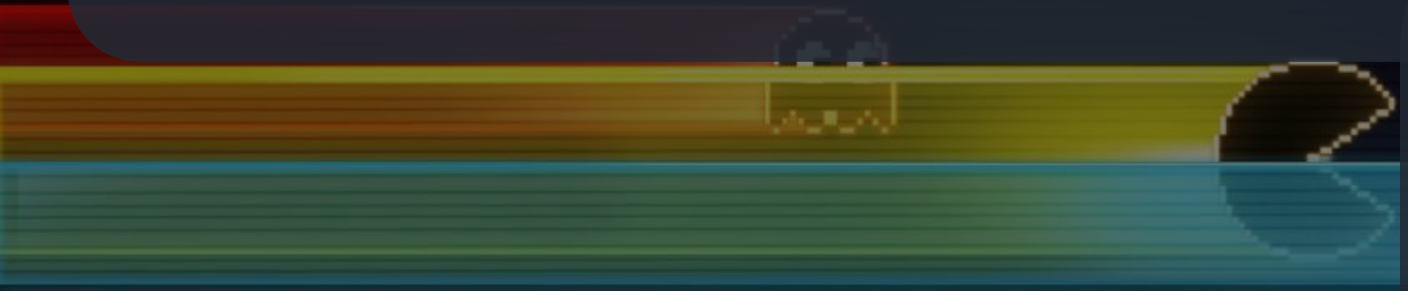
At most 51.2% wins for the best algorithm

- knowledge transfer?
- game identification?

How GVGAI differs…

From General Game Playing:

- We do videogames! Real-time constraints!
- We don’t make the ruleset available



How GVGAI differs…

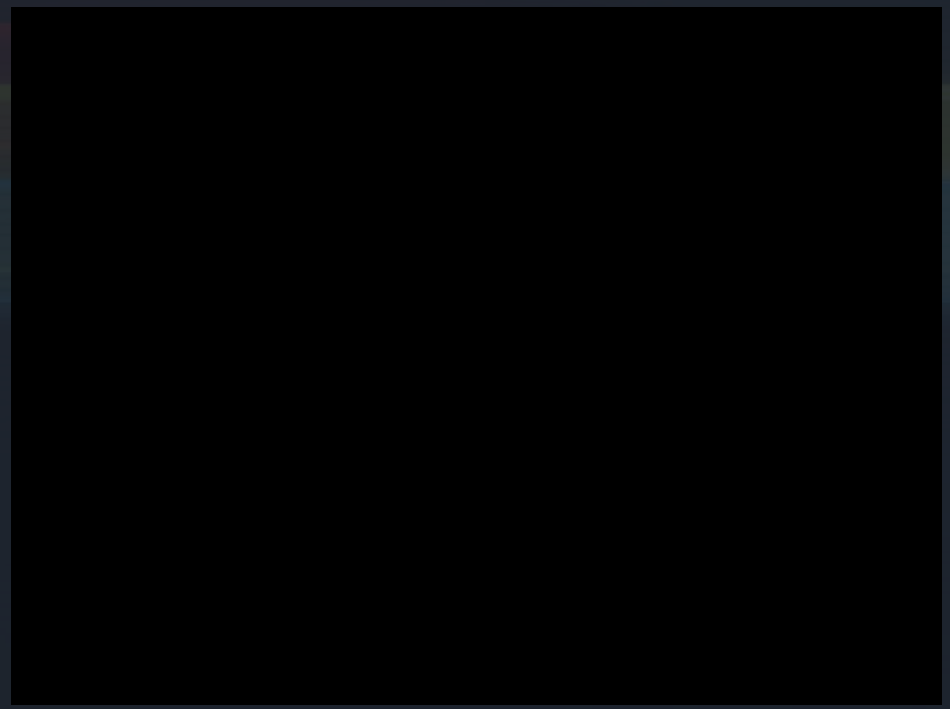
From the Arcade Learning Environment (ALE):

- Structured API (information via Java objects)
- Potentially infinite supply of games
- Agents tested on unseen games

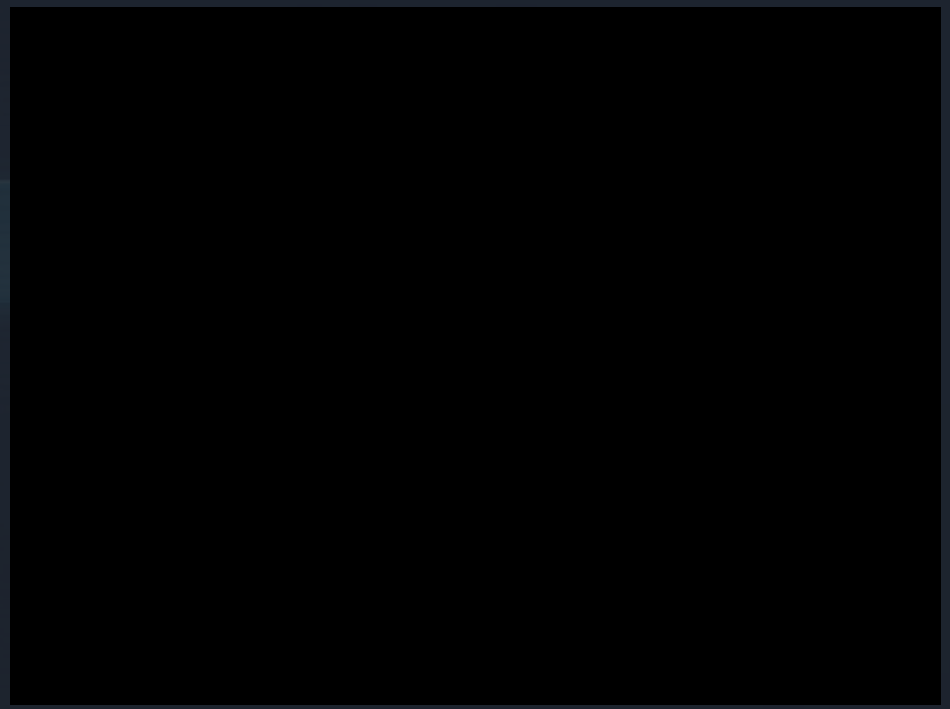
GVGAI for game design

- Use AI agents to evolve game parameters

Evolved Game (20-5) Sample



Evolved Game (5-20) Sample

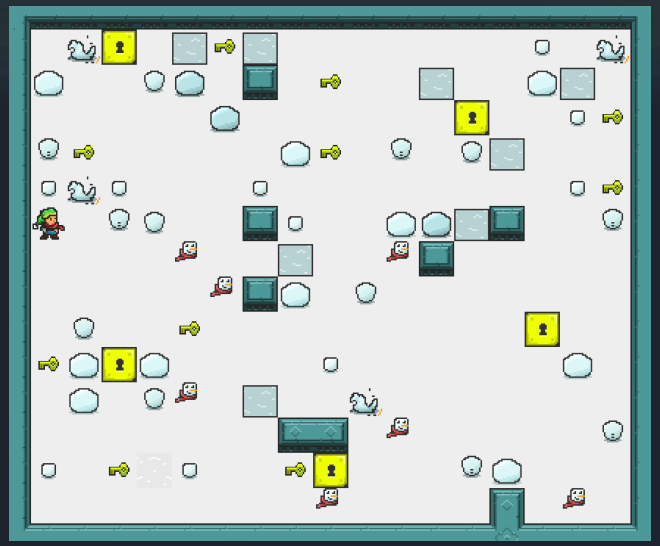


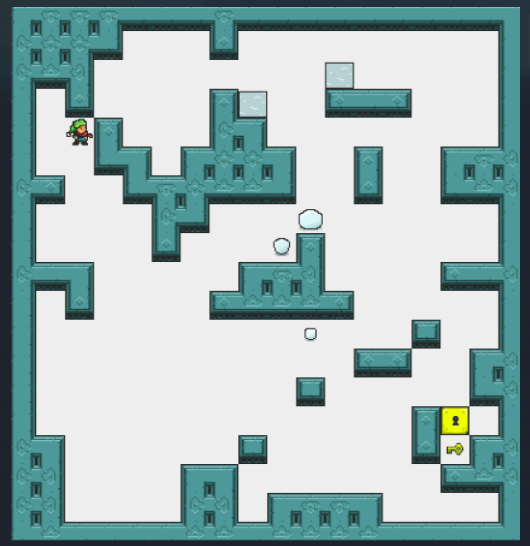
GVGAI for level & rule generation

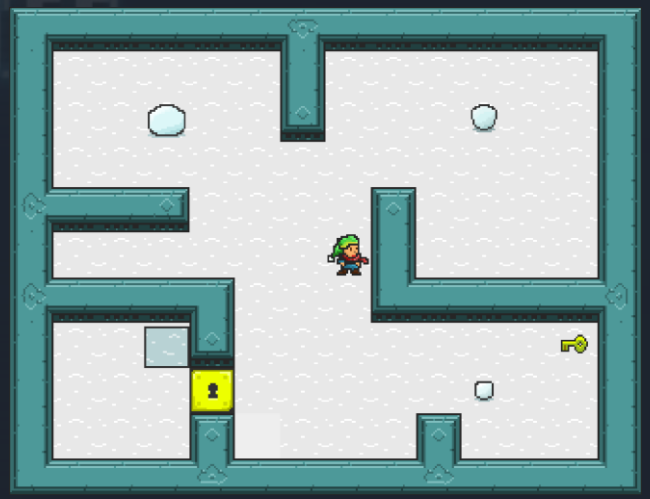
The GVGAI Competition – www.gvgai.net

GVGAI 2016 Competition - Level Generation Track:

Rank	Username	Country	Percentage Preferred (%)	Description	Generator	Screenshots
1	easablade	New Zealand	36	Description	Download	easablade
2	bhorn82287	Afghanistan	12	Description	Download	bhorn82287
2	Jnicho	United Kingdom	12	Description	Download	Jnicho
4	Number13	Germany	8	Description	Download	Number13







GVGAI for learning agents

- No forward model simulations, learn to play from experience
- Game state observation
  - JSON game objects
  - Screen capture (PNG file)
- Java or Python
- 5 minutes for training per game in first 3 levels
  - 10 runs per game in 2 new levels -> validation
- Winner 2017: Q-learning (but barely better than random…)

GVGAI for planning agents

Most successful approaches so far:

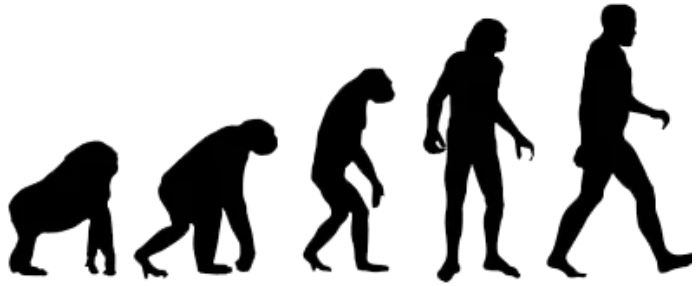
- Monte Carlo Tree Search
- Evolutionary Algorithms
- Hybrids
- Breadth-first search in deterministic games
- Value maps
- Interesting target identification

RHEA in GVGAI

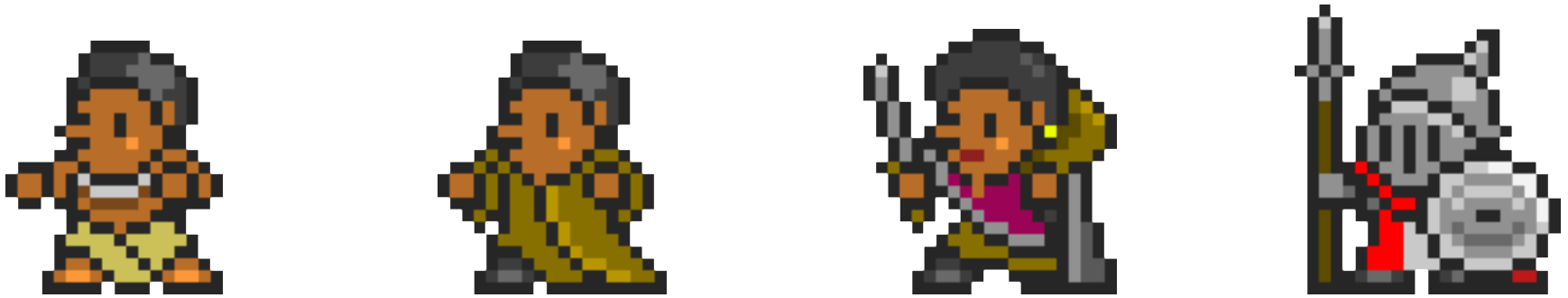
- Rolling Horizon Evolutionary Algorithm [2][3][4]
- Individuals as sequences of actions
- Great in puzzle games
- Not so great in quick reaction games
- Can be enhanced in various ways
  - Parameter optimization
  - Initialization methods
  - Shift buffer
  - Statistical tree
  - Monte Carlo rollouts

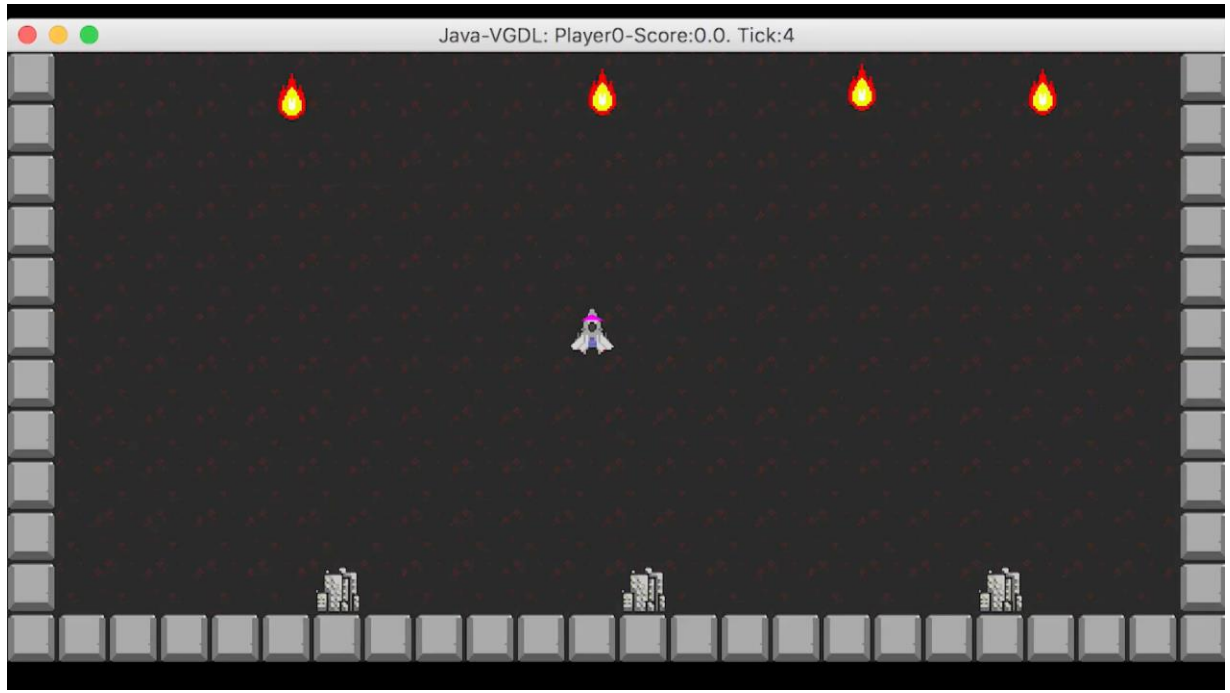
Evolving Game Playing Agents

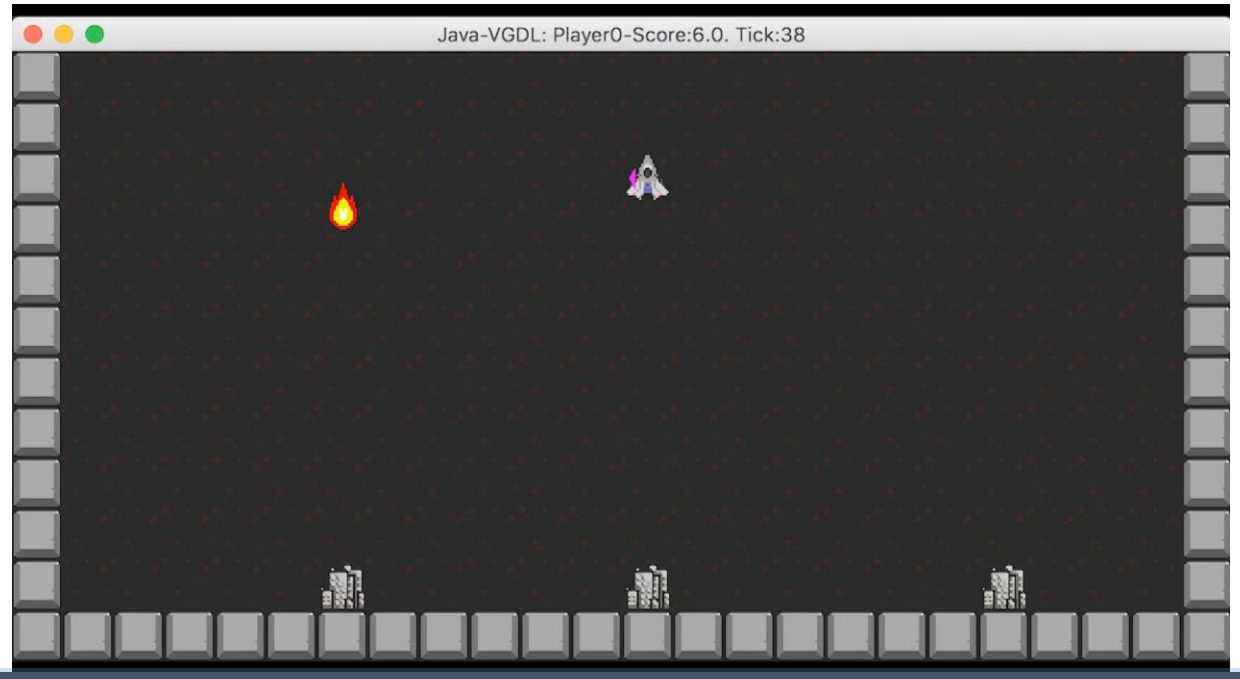
As humans evolved immersed in the environment
and the got better at surviving



It is possible to evolve
game-playing AI Agents that improve their performance
throughout generations





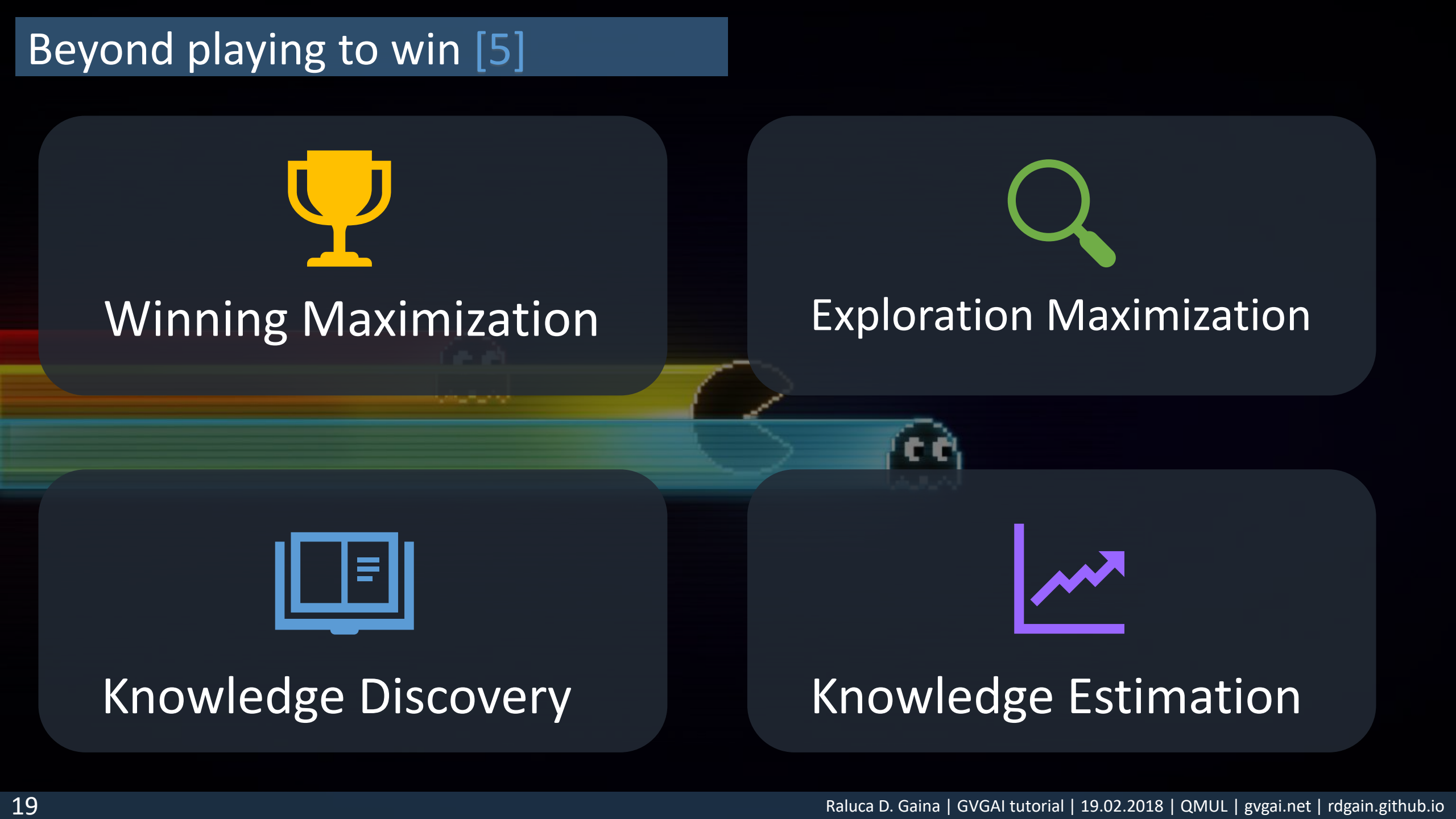


initial formula

evolution

improved formula

Beyond playing to win [5]



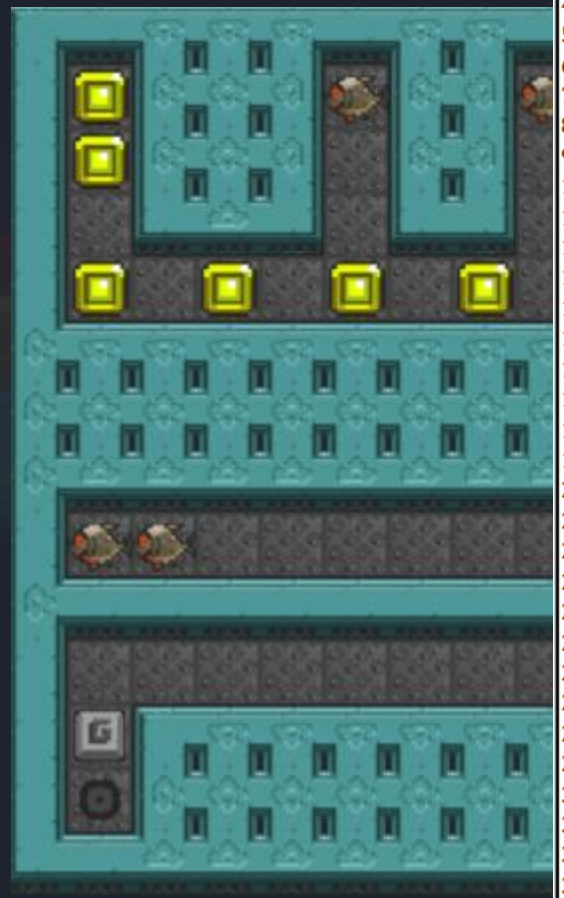
Winning Maximization

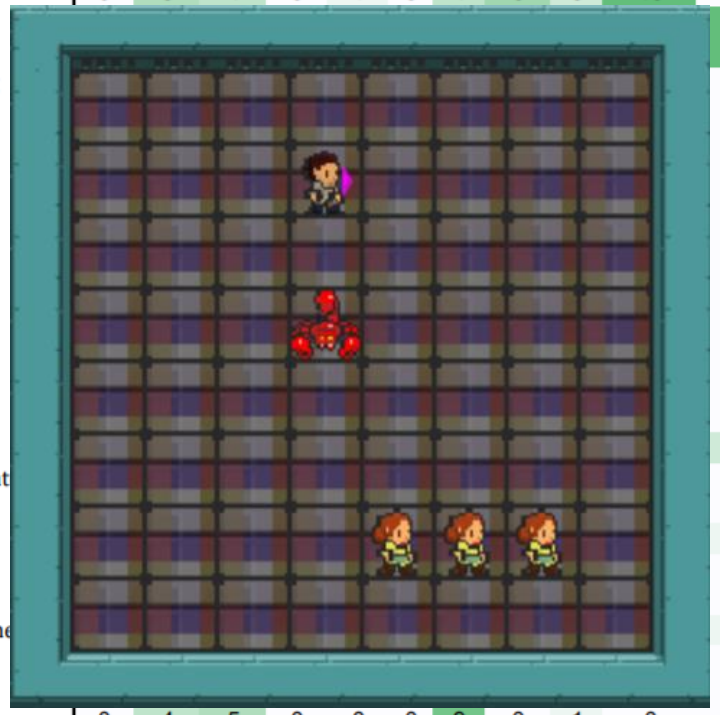
Exploration Maximization

Knowledge Discovery

Knowledge Estimation

Deceptive games [6]





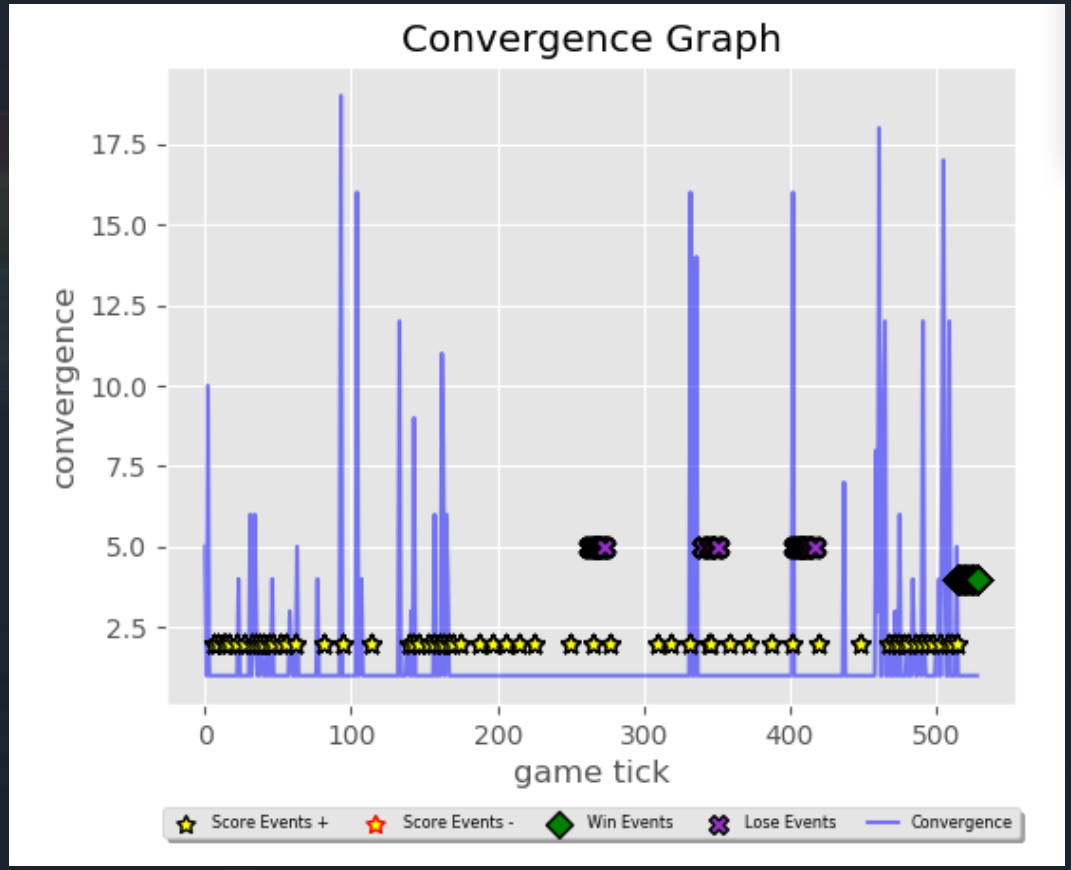
Agent Name	DC1	DC 2	DC 3	DZ 1	DZ 2	SS	BF	Flow	Inv	Mints 1	Mints 2	Rational
1. IceLab	10	2	10	0	0	0	2	10	10	10	10	8
2. Return42	10	1	7	0	0	2	1	10	10	10	0	8
3. MH2015	2	4	5	1	3	0	5	0	1	10	0	8
4. YoloBot	10	3	10	0	0	0	6	6	0	10	10	7
5. jaydee	9	6	9	0	0	0	1	10	0	10	3	7
6. NovTea	3	4	10	0	0	6	3	2	0	10	0	7
7. number27	0	5	4	0	1	0	4	6	3	10	0	7
8. YBCriber											10	6
9. adrienctx											10	6
10. TeamTopBug											0	6
11. Catlinux											0	6
12. muzzle											0	6
13. novelTS											0	6
14. bladerunner											0	6
15. maastCTS2											0	6
16. SJA86											0	6
17. Catlinux3											0	5
18. aStar											0	5
19. AtheneAI											0	5
20. Rooot											0	5
21. SJA862											0	5
22. roskvist											3	5
23. EvolutionStrat											0	5
24. AIJim											0	5
25. HillClimber											1	5
26. MnMCTS											0	5
27. mrtndwrd											0	4
28. simulatedAnne											1	4
29. TomVodo											0	4
30. ToVo1											0	4
31. Thorbjrn	0	4	5	0	0	0	9	0	1	0	0	4
32. BFS	0	1	0	0	0	0	5	0	10	0	0	3
33. Greedy Search	10	0	0	0	0	0	9	0	0	0	0	2
34. IterativeDeepening	0	0	0	0	0	2	1	0	0	0	0	2
35. Catlinux4	0	0	0	0	0	0	0	0	10	0	0	1
36. DFS	0	0	0	0	0	0	0	0	0	0	0	0
Totals	116	79	138	4	13	14	158	133	124	235	48	

GVGAI environments

- Single vs two-player
- Full vs partially observable
- Puzzles vs racing games
- Action/adventure games
- Collecting, shooting, navigation
- Deceptive games
- Competitive vs cooperative

GVGAI gameplay analysis

- Record more data about agent processing and game events
- Adjust strategy dynamically based on gameplay analysis



GVGAI Framework and competition

–––– www.gvgai.net ––––

https://github.com/GAIGResearch/GVGAI

https://github.com/GAIGResearch/GVGAI/wiki

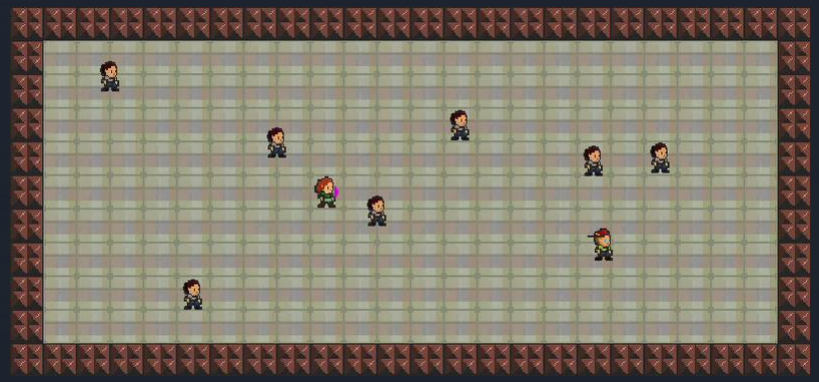
Video Game Description Language

Game description

```
BasicGame
  SpriteSet
    floor > Immovable img=newset/floor6 hidden=True
    human >
        annoyed > RandomNPC speed=0.25 img=newset/cursedman cons=2
        citizen >
            quiet > RandomNPC speed=0.25 img=newset/man2 cons=1
            avatar > ShootAvatar stype=cigarette img=newset/girl1 rotateInPlace=False
    george > Chaser stype=citizen speed=0.25 img=newset/man4 frameRate=8
    cigarette > Flicker limit=5 singleton=True img=newset/cigarette
    wall > Immovable img=oryx/wall6

  TerminationSet
    SpriteCounter stype=avatar  win=False
    SpriteCounter stype=quiet   win=False
    Timeout limit=1000 win=True

  InteractionSet
    quiet george > transformTo stype=annoyed
    avatar george > killSprite scoreChange=-1
    annoyed cigarette > transformTo stype=quiet scoreChange=1
    human wall wall > stepBack

  LevelMapping
    g > floor george
    c > floor quiet
    A > floor avatar
    . > floor
```

Level definition

```
wwwwwwwwwwwwwwwwwwwwwwwww
w.......................w
w..c.........c.....c...w
w.......c..............w
w.......................w
w.......A.......c.....w
w..........c...........w
w.......................w
w....c..............g...w
w.......................w
wwwwwwwwwwwwwwwwwwwwwwwww
```



A GVGAI 1P planning agent

```
public class Agent extends AbstractPlayer{


    /**
     * constructor called once at the beginning of each game.
     * @param stateObs Observation of the current state.
     * @param elapsedTimer Timer when the action returned is due.
     */
    public Agent(StateObservation stateObs, ElapsedCpuTimer elapsedTimer){
    }

    /**
     * act method called at every game tick.
     * @param stateObs Observation of the current state.
     * @param elapsedTimer Timer when the action returned is due.
     * @return ACTION_NIL all the time
     */
    @Override
    public ACTIONS act(StateObservation stateObs, ElapsedCpuTimer elapsedTimer) {
        return Types.ACTIONS.ACTION_NIL;
    }
}
```

StateObservation objects

Available in all agent methods: constructor, act, result

```
ArrayList<Types.ACTIONS> getAvailableActions()
int getNoPlayers()
double getGameScore()
int getGameTick()
Types.WINNER getGameWinner()
boolean isGameOver()
Dimension getWorldDimension()
int getBlockSize()

ArrayList<Observation>[][] getObservationGrid()
TreeSet<Event> getEventsHistory()
ArrayList<Observation>[] getNPCPositions()
ArrayList<Observation>[] getImmovablePositions()
ArrayList<Observation>[] getMovablePositions()
ArrayList<Observation>[] getResourcesPositions()
ArrayList<Observation>[] getPortalsPositions()
ArrayList<Observation>[] getFromAvatarSpritesPositions()
```

StateObservation objects

Available in all agent methods: constructor, act, result

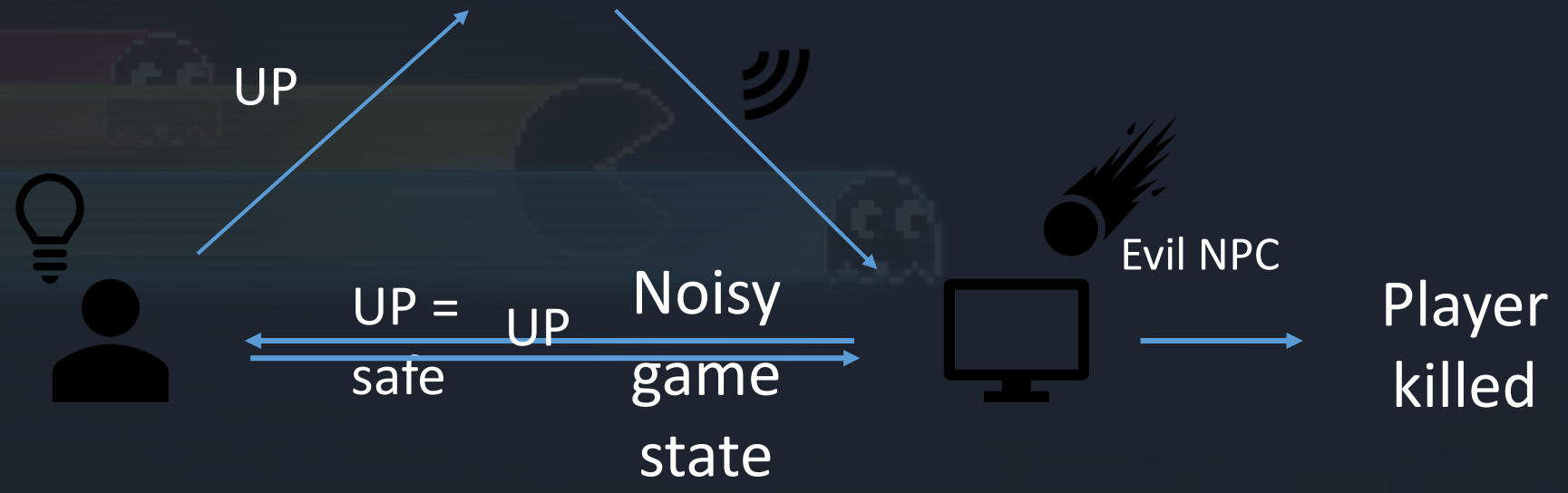
```
Vector2d getAvatarPosition()
double getAvatarSpeed()
Vector2d getAvatarOrientation()
HashMap<Integer, Integer> getAvatarResources()
Types.ACTIONS getAvatarLastAction()
int getAvatarType()
int getAvatarHealthPoints()
int getAvatarMaxHealthPoints()
int getAvatarLimitHealthPoints()
boolean isAvatarAlive()
```

```
void advance(Types.ACTIONS action)
StateObservation copy()
```

ForwardModel

Simulate possible next game states.

```
void advance(Types.ACTIONS action)
```



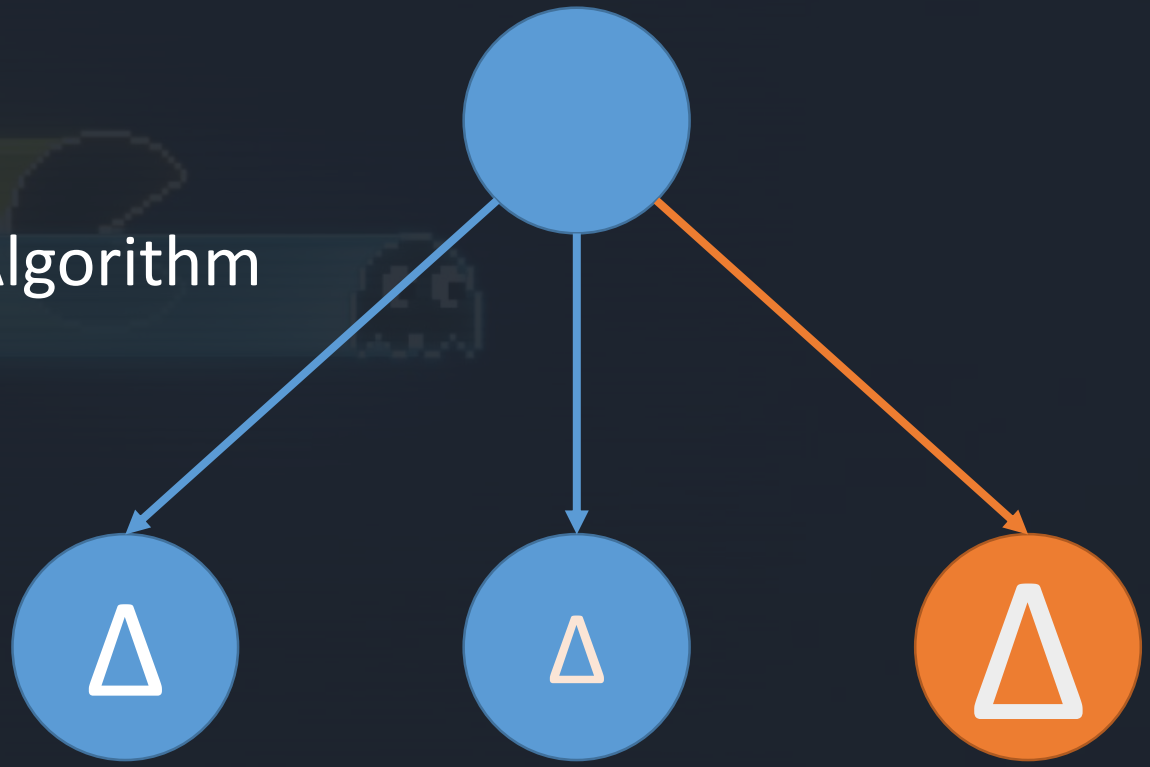
Testing a 1P planning agent

1. Find class src.tracks.singlePlayer.Test.java
2. Set gameIdx and levelIdx variables
3. Create a String variable containing the path to your Agent class, such as:
   - `String sampleMCTSController = "tracks.singlePlayer.advanced.sampleMCTS.Agent";`
4. Select running mode:
   - Play as human
   - Play one game with a controller (pass your agent String to the method)
   - Replay a recorded game
   - Play N games in L levels repeated M times (visuals off)
5. Run Test.java!

Sample agents in the framework

In packages src.tracks.singlePlayer.simple and src.tracks.singlePlayer.advanced

- Do Nothing
- Random
- Simple One Step Look Ahead
- Monte Carlo Tree Search
- Rolling Horizon Evolutionary Algorithm
- Random Search

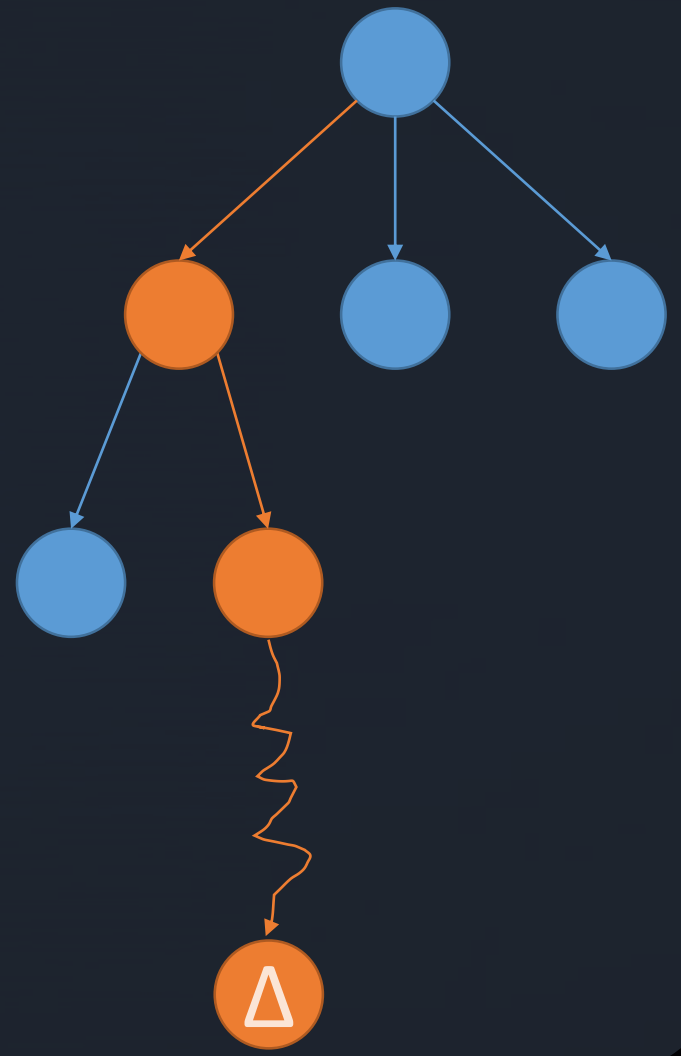


Monte Carlo Tree Search

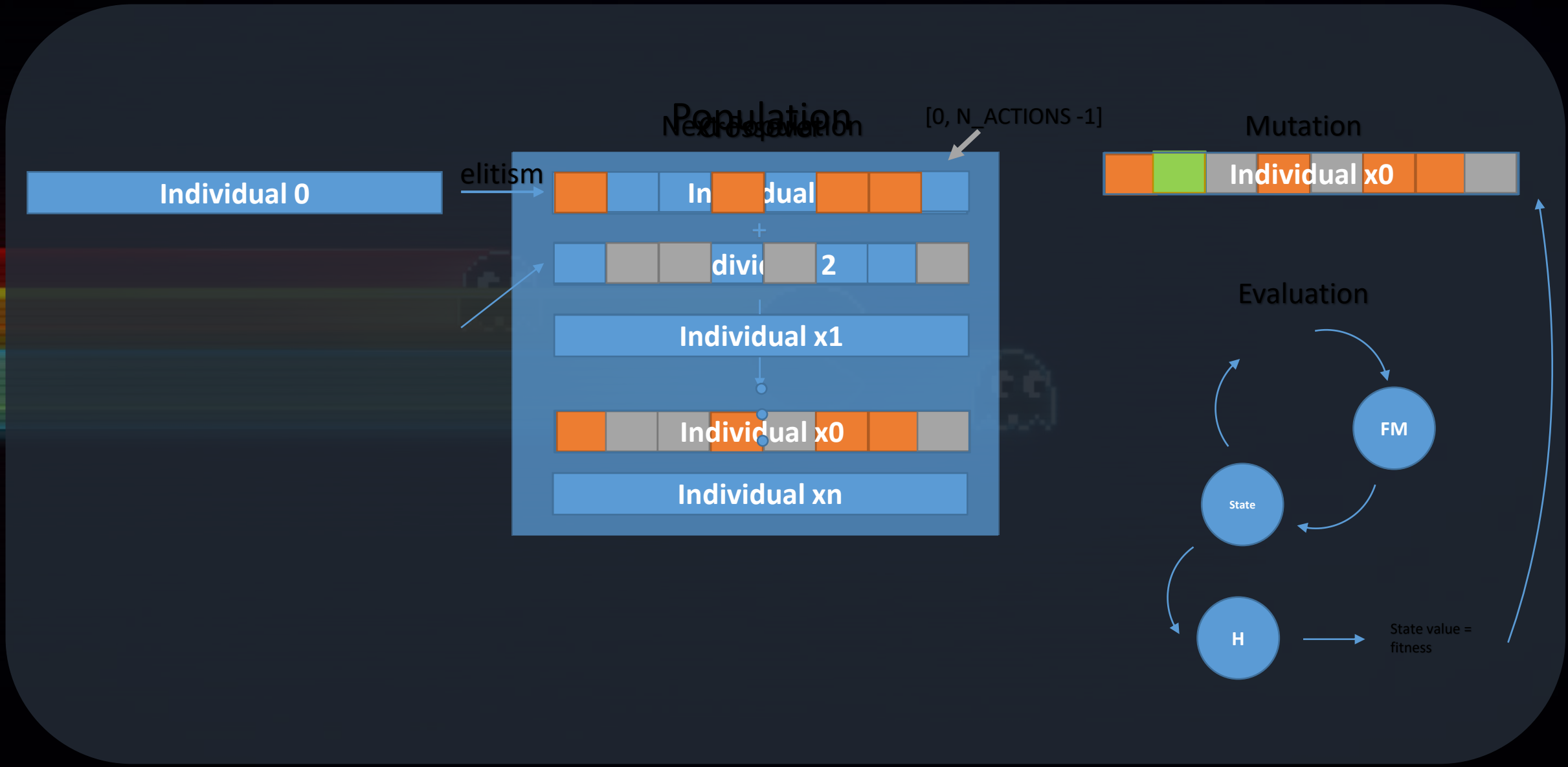
In class sampleMCTS.SingleTreeNode

```
mctsSearch ──→ mostVisitedAction
    ↓
treePolicy ──→ uct
    ↓
expand
    ↓
rollOut ──→ value
    ↓
backUp
```

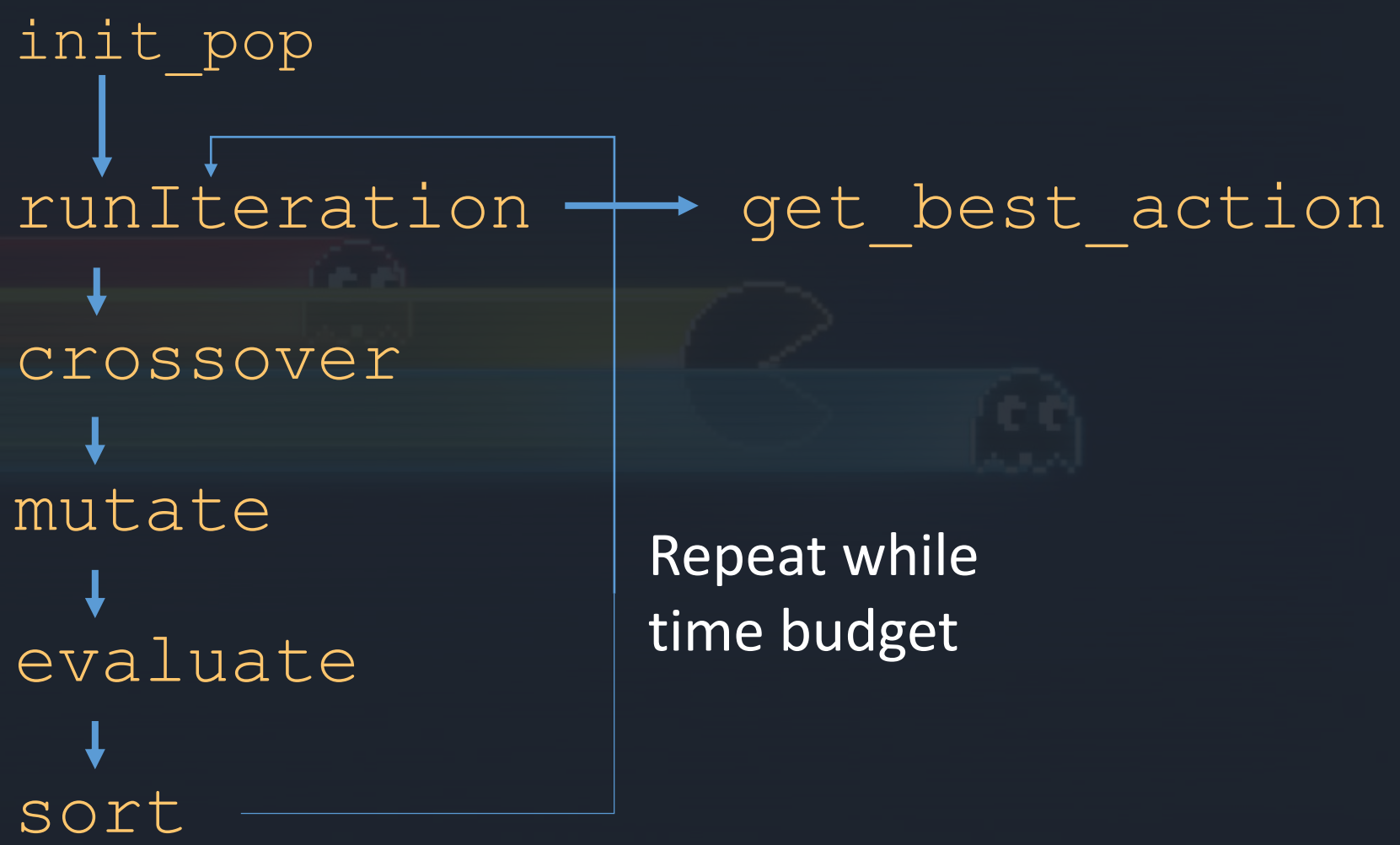
Repeat while time budget



Rolling Horizon Evolutionary Algorithm



Rolling Horizon Evolutionary Algorithm



QMUL GVGAI tutorial private league!

Best AI Agent

£20 Voucher

Best Game

£20 Voucher

QMUL GVGAI tutorial private league!

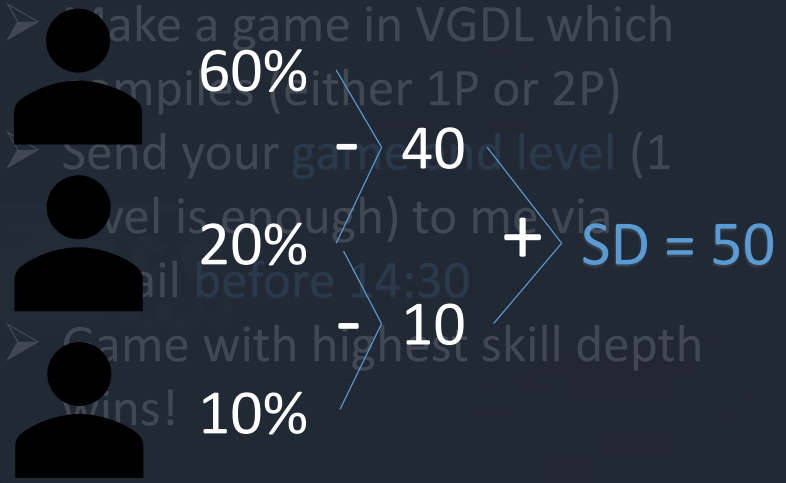
Best AI Agent

- Register on the GVGAI website
- Submit a functioning 1P planning agent to the *QMUL private league* before 14:30
- Agent with most F1 points according to GVGAI ranking system wins!

£20 Voucher

Best Game

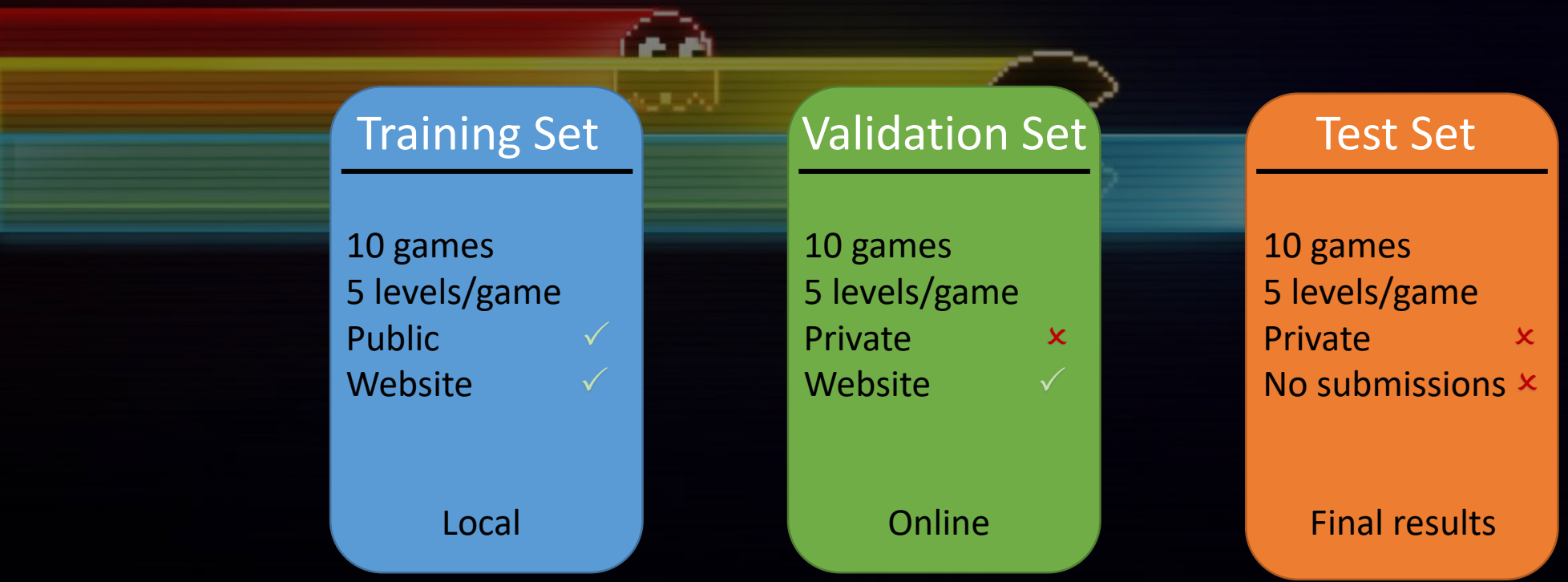
- Make a game in VGDL which compiles (either 1P or 2P)
- Send your game and level (1 level is enough) to me via email before 14:30
- Game with highest skill depth wins!



£20 Voucher

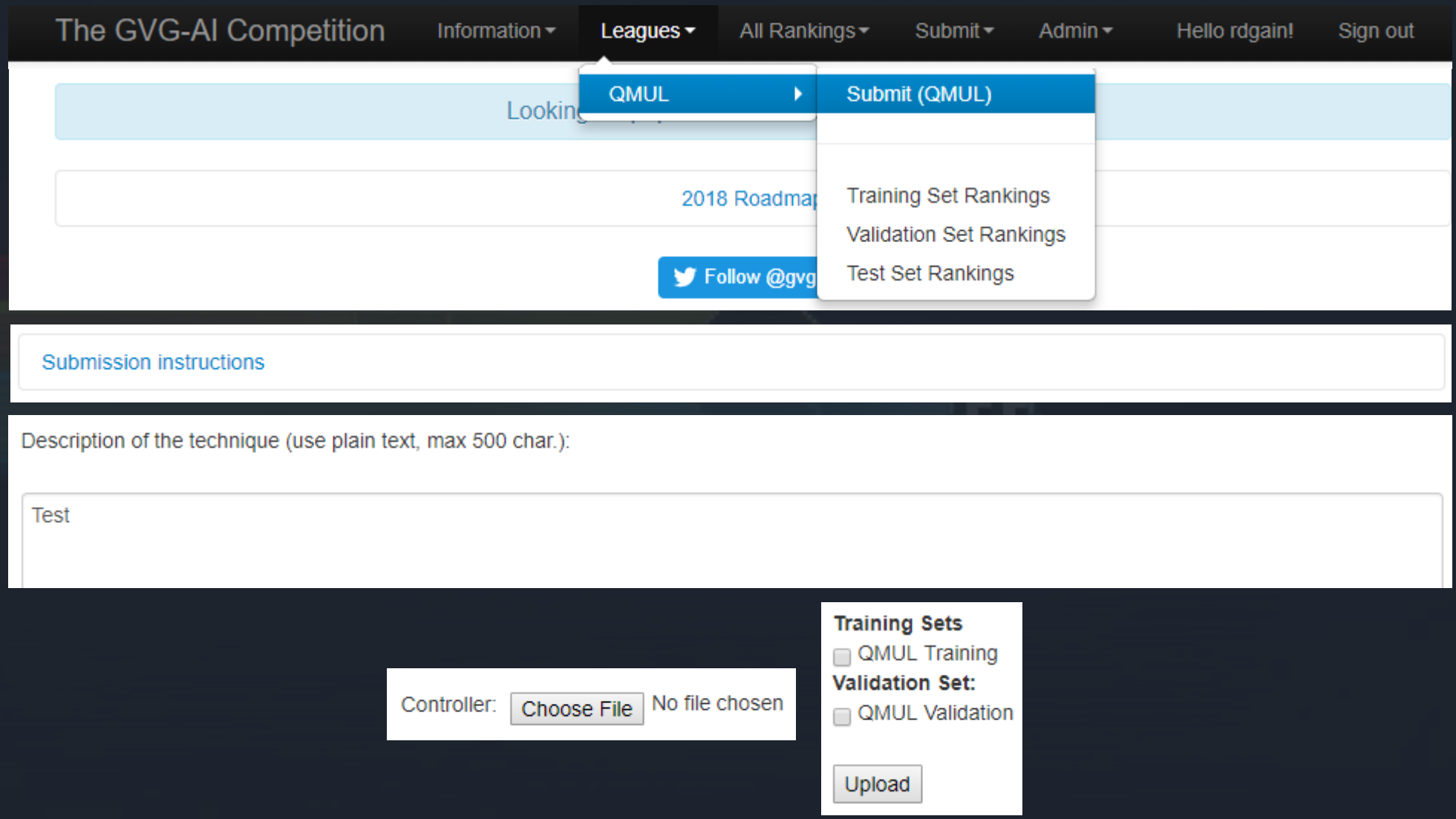
GVGAI website - competition

1. Sign up on gvgai.net to be able to submit your agent
2. Shout when done to collect usernames
3. Log in -> Private Leagues top menu tab -> QMUL



Training Set	Validation Set	Test Set
10 games	10 games	10 games
5 levels/game	5 levels/game	5 levels/game
Public ✓	Private ✗	Private ✗
Website ✓	Website ✓	No submissions ✗
Local	Online	Final results

GVGAI website submission



GVGAI ranking system

Bubble:

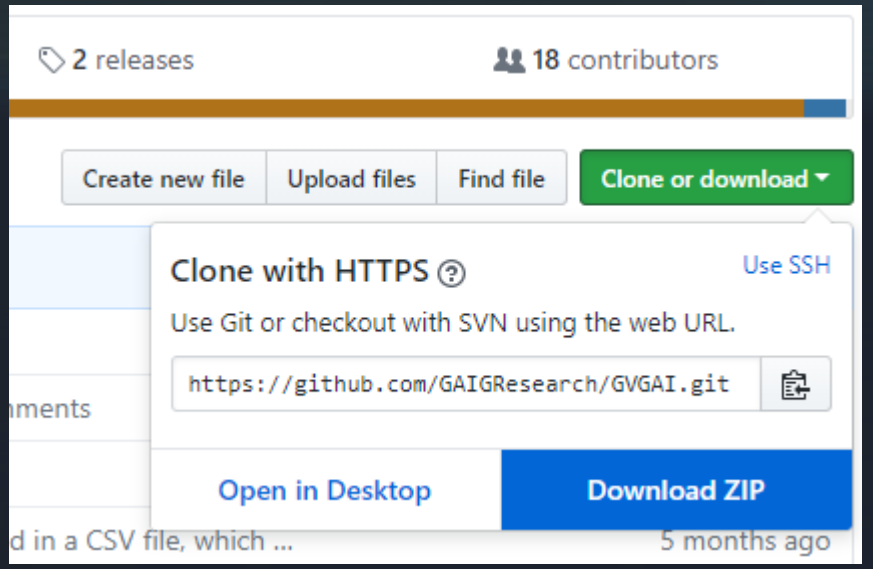
Game	Username	Country	Points	Winner %	Avg. Score	Avg. Timesteps
Bubble	Cyclus	Germany	25	100	18 (3.54)	566.6 (157.68)
Bubble	Jaybot	Germany	18	100	16.6 (3.26)	357 (78.45)
Bubble	sampleOLMCTS	United Kingdom	15	60	14.8 (2.86)	906.2 (175.25)

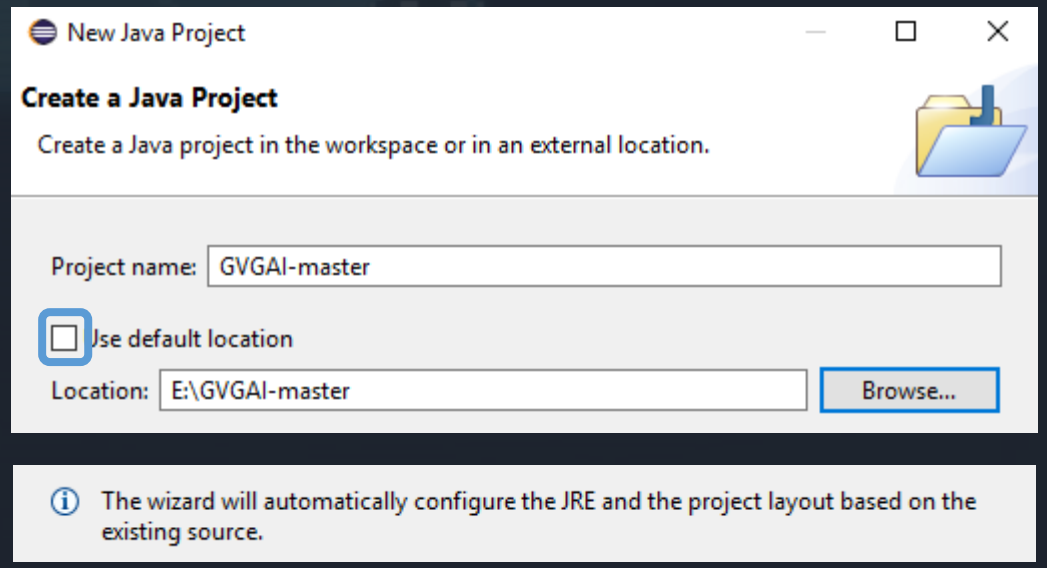
Total | Artillery | Asteroids | Bird | Bubble | Candy | Lander | Mario | Pong | PTSP | Racing

Rank	Username	Country	Description	G-1	G-2	G-3	G-4	G-5	G-6	G-7	G-8	G-9	G-10	Total
1	asd592	Germany	Description	1	12	15	6	18	0	10	8	**25**	**25**	**120**
2	Jaybot	Germany	Description	2	10	**25**	18	8	1	12	18	8	12	**114**
3	JACAM	Germany	Description	**25**	18	0	0	15	0	8	**25**	18	0	**109**

Opening GVGAI in Eclipse

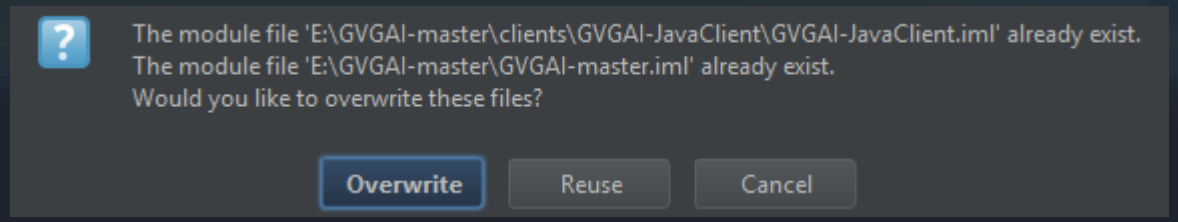
1. Download the GVGAI framework
2. Unzip
3. Open Eclipse and create a new workspace in any other folder.
4. File -> New -> Project… -> New Java project
5. Untick ‘default location’ and select the GVGAI-master directory location -> Finish
6. In project view, navigate to **src.tracks.singlePlayer** -> **Test.java** class -> Run!

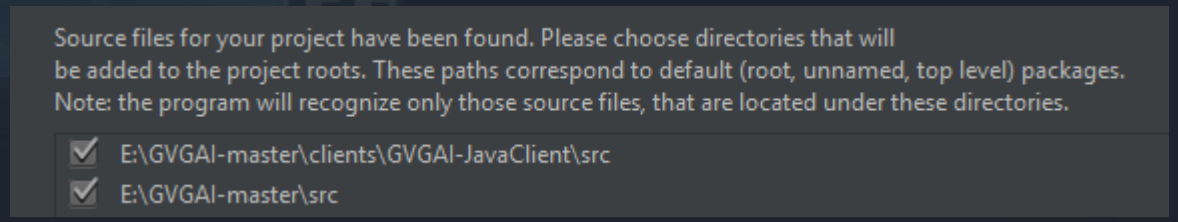


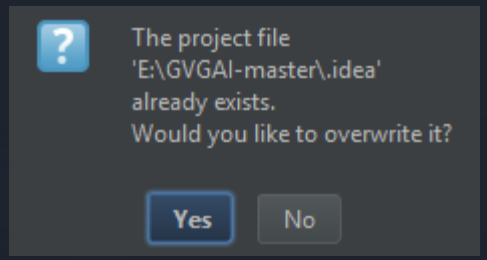


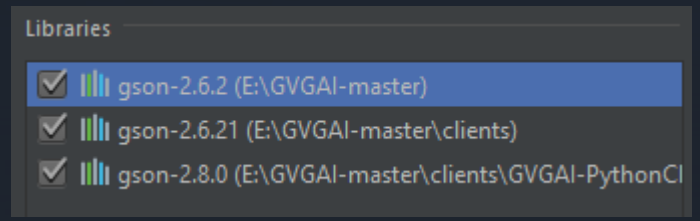
Opening GVGAI in IntelliJ

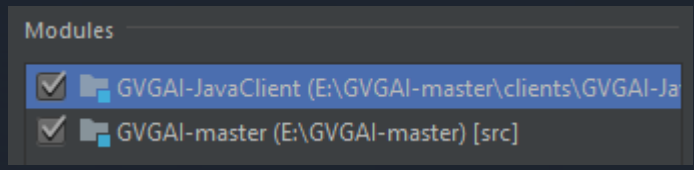
1. Download the GVGAI framework
2. Unzip
3. Open IntelliJ and open another project / create a new one (can be deleted after)
4. File -> New -> Project from Existing Sources …
5. Overwrite project and module files when asked ☑
6. Import all libraries and modules by default (Next … Next -> Finish) ☑
7. In project view, navigate to **src** -> **tracks** -> **singlePlayer** -> **Test.java** class -> Run!











Let’s get hands on

–––– www.gvgai.net ––––

https://github.com/GAIGResearch/GVGAI

https://github.com/GAIGResearch/GVGAI/wiki

Tasks for 2nd part

Root

Make a 1P planning agent for the private league

Make a VGDL game (if 1P, it can be used for the private league, to get the agents tested on it)

Experiment with other tracks, specific game sets or game design features

- Two player planning
- Single player learning
- Level generation
- Rule generation

Work either solo or small teams (3-4 max)

Where to start?!!

Oh no this sounds too difficult…

If you can't program …
… find someone who can!

If you're making a game …
… read the wiki information on VGDL!

If you're making an agent …
… start from a sample agent or previous successful entries

- Change parameters
- Change heuristic (state evaluation)
- Change algorithm logic (e.g. mutation operator for EAs; tree policy for MCTS)
- Mash algorithms together (MCTS as mutation operator in EA?)

But first...

Lunch!

Talk to QMUL Game AI group members

Find potential work buddies

Works Cited

- [1] Peter Keevash and Liana Yepremyan, “Rainbow matchings in properly-coloured multigraphs”, CoRR http://arxiv.org/abs/1710.03041, 2017
- [2] Raluca D. Gaina, Jialin Liu, Simon M. Lucas, Diego Perez-Liebana, Analysis of Vanilla Rolling Horizon Evolution Parameters in General Video Game Playing, Proceedings of EvoApplications 2017 (EvoGames) (2017).
- [3] Raluca D. Gaina, Diego Perez-Liebana and Simon M. Lucas, Population Seeding Techniques for Rolling Horizon Evolution in General Video Game Playing, Proceedings of the Congress on Evolutionary Computation (2017).
- [4] Raluca D. Gaina, Diego Perez-Liebana and Simon M. Lucas, Rolling horizon Evolution Enhancements in General Video Game playing, in Proceedings on the Conference on Computational Intelligence and Games (CIG) (2017).
- [5] Cristina Guerrero-Romero, Annie P. Louis and Diego Perez-Liebana, Beyond Playing to Win: Diversifying Heuristics for GVGAI, in Proceedings on the Conference on Computational Intelligence and Games (CIG) (2017).
- [6] Damien Anderson, Matthew Stephenson, Julian Togelius, Christoph Salge, John Levine, and Jochen Renz, “Deceptive Games”, CoRR https://arxiv.org/pdf/1802.00048.pdf, accepted at EvoStar, 2018

- [Tom Schaul, 2013] Schaul, T. A Video Game Description Language for Model-based or Interactive Learning, Proceedings of the IEEE Conference on Computational Intelligence in Games, 2013, 193-200
- [Perez et al., 2015] Perez, D.; Samothrakis, S.; Togelius, J.; Schaul, T.; Lucas, S.; Couëtoux, A.; Lee, J.; Lim, C.-U.; and Thompson, T. 2015. The 2014 General Video Game Playing Competition. IEEE Transactions on Computational Intelligence and AI in Games (to appear) DOI: 10.1109/TCIAIG.2015.2402393.
- [R Gaina, 2017] Raluca D. Gaina, Adrien Couetoux, Dennis J.N.J. Soemers, Mark H.M. Winands, Tom Vodopivec, ¨ Florian Kirchgeßner, Jialin Liu, Simon M. Lucas, Diego Perez-Liebana. The 2016 Two-Player GVGAI Competition, in Transactions on Computational Intelligence and AI in Games (2017)
- http://gameaibook.org/

Thank you!

Raluca D. Gaina
IGGI PhD Student

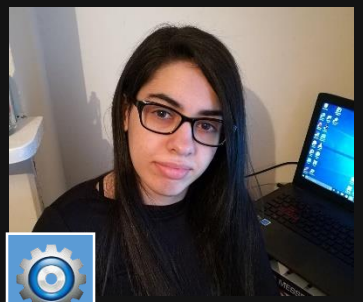
r.d.gaina@qmul.ac.uk
rdgain.github.io
@b_gum22

Ivan Bravi
IGGI PhD Student

i.bravi@qmul.ac.uk
@ivanbravi

Cristina Guerrero Romero
IGGI PhD Student

c.guerreroromero@qmul.ac.uk
@kisenshi



Cristiana Pacheco
IGGI PhD Student

c.pacheco@qmul.ac.uk
@Pacheco_CrisP

Diego Perez Liebana
GVGAI Overlord

diego.perez@qmul.ac.uk
@diego_pliebana