Queen Mary
University of London

STRATEGA - A General Strategy Games Framework

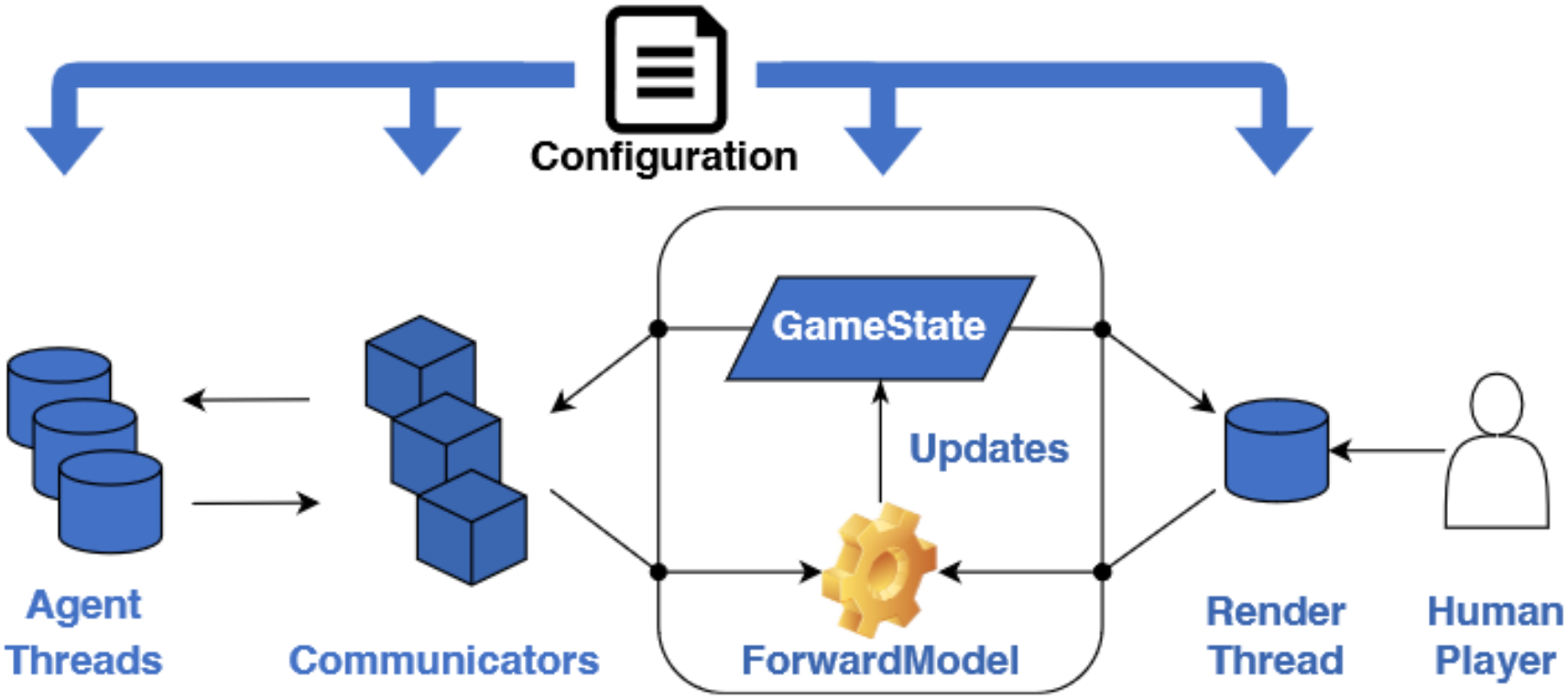
Alexander Dockhorn, Jorge Hurtado-Grueso,
Dominik Jeurissen, Diego Perez-Liebana

AIIDE 2020 Workshop on AI for Strategy Games

Stratega – A General Strategy Games Framework

- A single framework for turn-based and real-time strategy games.
- Easy creation and configuration of games using YAML-files.
- Built for research on Statistical Forward Planning (SFP) agents.

Framework Overview



Built for Statistical Forward Planning Agents

- A framework for research on general strategy game-playing.
  - All games defined in our framework use a common interface!
- Each game offers access to a forward model.
  - The framework has been optimized to maximize the number of possible forward model calls.
  - Observed game-states can be freely manipulated by the agent.
- The framework is implemented in C++ to assure a high execution speed
  - Headless mode for running games with enhanced speed.

Configuration

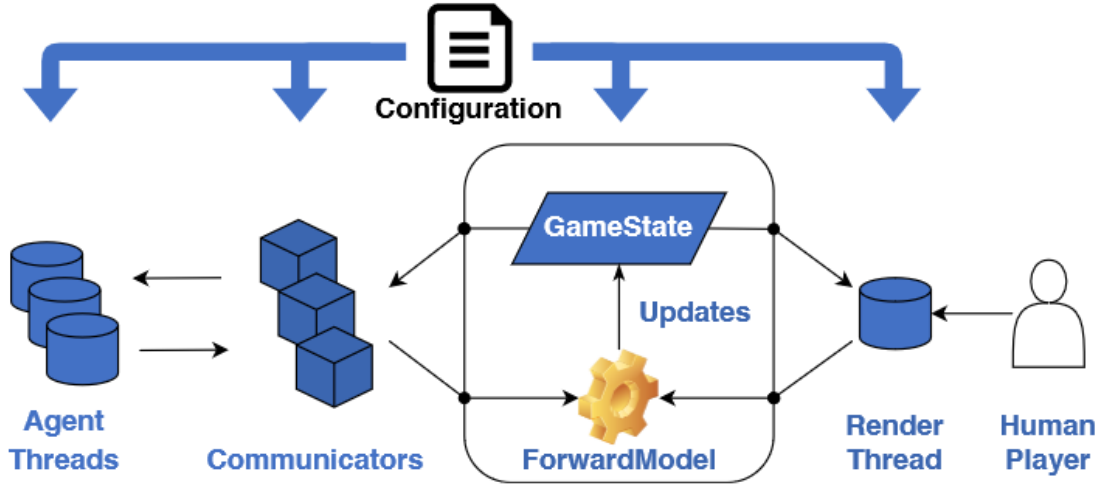
General

Tiles and Boards

Units and Actions

Forward Model

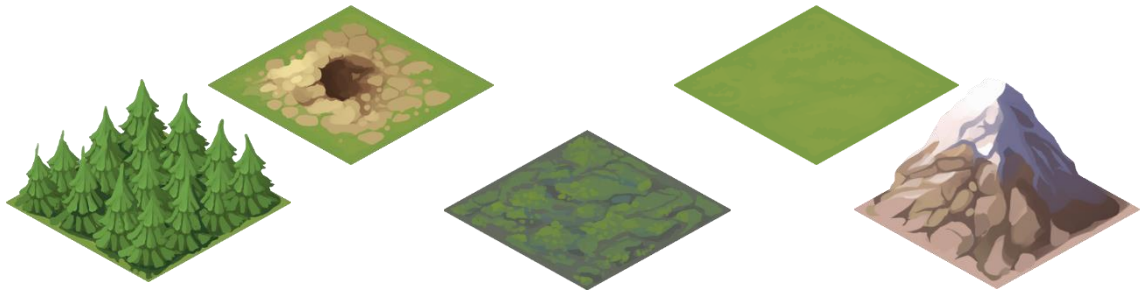
- Games and Agents can be configured using YAML-files
- They can be used to:
  - quickly generate variances of a game
  - balance a game's parameters
  - setup experiments



Configuration

- General
- **Tiles and Boards**
- Units and Actions
- Forward Model

- Users can define their own tiles.
- Each tile can have a variety of properties or tile effects.
- Maps are encoded as tile maps.
- They can be manually defined or automatically generated.



```
Tiles:
    Swamp:
        Symbol: S
        IsWalkable: true
    Mountain:
        Symbol: M
        IsWalkable: false
    Hole:
        Symbol: H
        IsWalkable: true

Board:
    GenerationType: Manual
    Layout: >
        MMMMM
        MSSSM
        MSSHM
        MSSHM
        MSSSM
        MMMMM
```

Configuration

- General
- Tiles and Boards
- **Units and Actions**
- Forward Model

- Similarly to tiles, users can generate their own units.
- Some base-properties are required, e.g. health, movement range, line of sight range and attack damage.
- RTSUnits require some time related properties, e.g. movement speed.
- Introducing new actions requires adding respective code. Parameterized actions will be added soon.

```
Units:
    Warrior:
        Health: 100
        MovementRange: 3
        LineOfSightRange: 4
        AttackDamage: 20
        Actions: [Attack, Move]
    Healer:
        Health: 40
        MovementRange: 5
        LineOfSightRange: 4
        HealAmount: 10
        Actions: [Heal, Move]
```



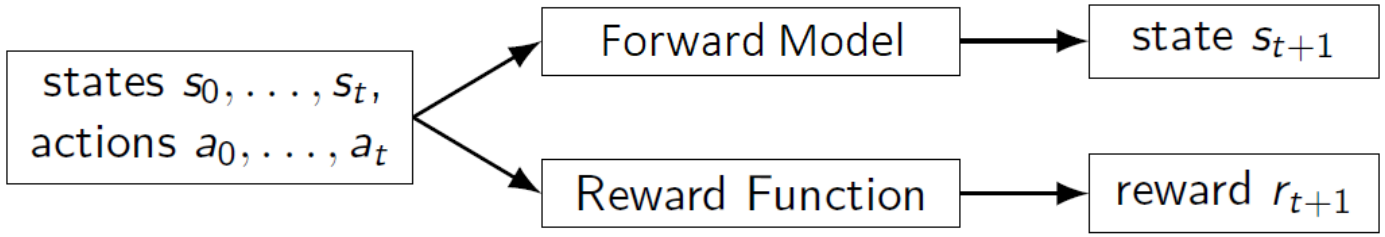


Configuration

- General
- Tiles and Boards
- Units and Actions
- **Forward Model**

- The forward model includes the game's rules.
- Choose among a set of win conditions or define your own.
- Implement unique effects and quickly change their parameters to create a unique game-mode.

```
ForwardModel:
    WinCondition: LastManStanding
    Effects:
        DamageAll:
            Type: Damage
            Trigger: EndOfTurn
            Condition: None
            Amount: 10
        DeadlyHole:
            Type: Death
            Trigger: EnterTile
            Condition: StandingOnTile
            TargetTile: Hole
```



A variety of game-modes

Kings

- Each agent needs to defend its king and kill the opponent king.
- unique win-condition

Pushers

- Units cannot fight but push each other into holes to kill.
- unique abilities

Healers

- Units continuously lose health and need to be healed.
- unique events

Graphical User Interface (GUI)

- View and play games through our GUI
  - Human players can play both game-modes via mouse controls
- Show additional logging information at real-time
- Send game-states to the GUI to visualize a search-path for simplified debugging of SFP agents

Agents

- Each Agent runs in a separate thread.
  - Allows for computation during the opponent‘s turn.
- A communicator object lets them observe the current game-state.
- Agents need to return actions to the communicator which will first be checked for validity and then applied by the forward model.
- A human controller interface is available to play against bots.

Agents

The framework includes many baseline agents and further agents will be added in future updates.

Basic Agents

- Rule-based Agents
- One Step Lookahead
- Breadth-First Search
- Depth-First Search
- Beam Search

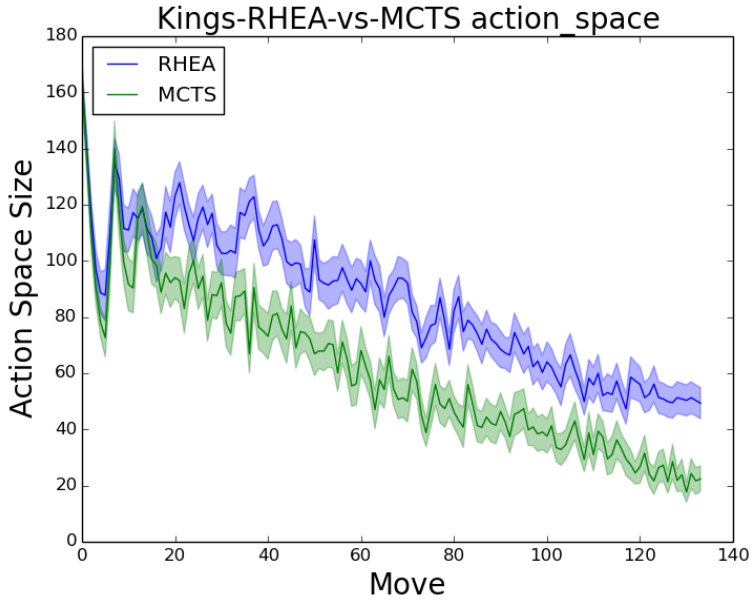
Advanced Agents

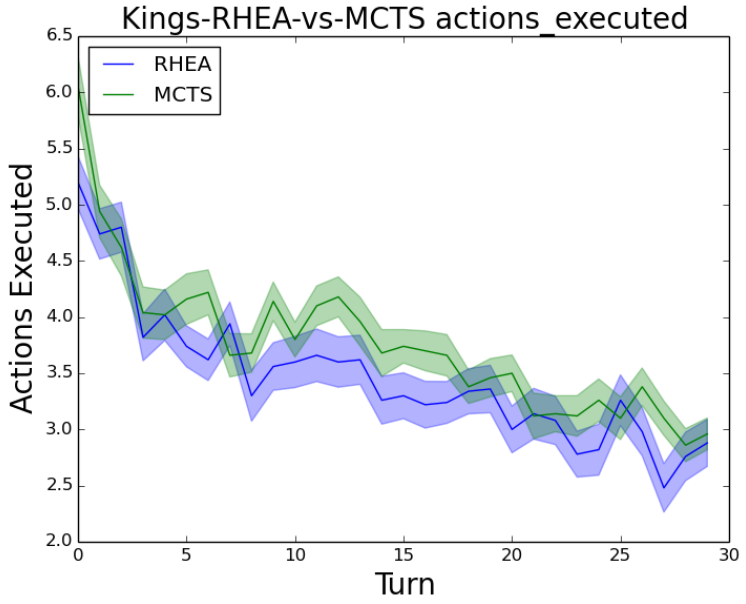
- Monte Carlo Tree Search (MCTS)
- Rolling Horizon EA (RHEA)
- Portfolio Greedy Search
- Portfolio RHEA

Logging

Supporting Debugging and Evaluation

- game-related statistics are tracked automatically
- Agents can log additional information through a logging interface





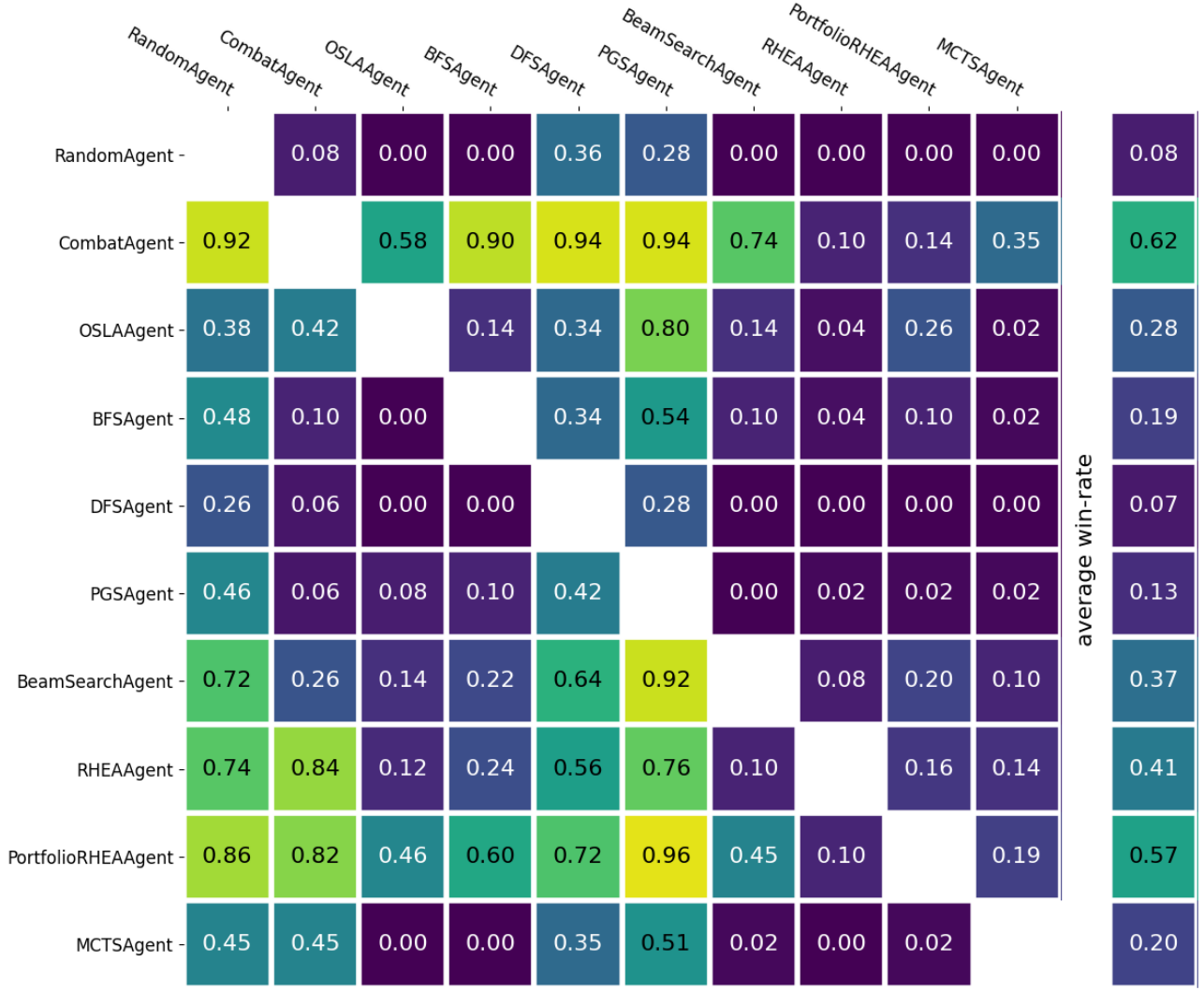
Experiments

All agents have been tested in three game-modes.

- Rule-based agents dominated our experiments.
- Using portfolios increased the performance, but will require a more in-depth analysis.
- Adding an opponent model improved the performance of all search-based agents.

Agents	RBC	OSLA	MCTS	RHEA	Average
	Kings				
RBC	—	1.00	0.86	0.90	0.92
RHEA	0.10	0.98	0.60	—	0.56
MCTS	0.14	0.92	—	0.12	0.39
OSLA	0.00	—	0.02	0.00	0.01
	Healers				
RBC	—	0.98	0.82	0.66	0.82
RHEA	0.34	1.00	0.70	—	0.68
MCTS	0.16	0.94	—	0.26	0.45
OSLA	0.02	—	0.06	0.00	0.03
	Pushers				
RBP	—	1.00	0.46	0.74	0.73
MCTS	0.54	1.00	—	0.30	0.61
RHEA	0.26	0.94	0.40	—	0.53
OSLA	0.00	—	0.00	0.00	0.00

Experiments



Real-Time Strategy Mode in Active Development

Opportunities and Future Work

- We work closely with industry partners to shape the future of our project.

- Future updates will increase the variety of possible game-mechanics.
  - tech trees, (de-)buffs, object pick-ups, inventories, economy management

- We plan to host competitions on general strategy game AI.
  - General Strategy Game-playing
  - Balancing
  - Map/Content Generation

Thank you for your attention!

This work is supported by UK EPSRC research grant EP/T008962/1.

Interested in testing the framework yourself? Download the Stratega framework on Github:
https://github.com/GAIGResearch/Stratega

by Alexander Dockhorn, Jorge Hurtado-Grueso,
Dominik Jeurissen, Diego Perez-Liebana
Email: {a.dockhorn, diego.perez}@qmul.ac.uk