RoboCup Project Presentation
SYSC 5103F

Aron Arany-Takacs, Bardia Parmoun, & Dylan Leveille

Department of Systems and Computer Engineering
Carleton University, Ottawa, ON, Canada

aaran043@uottawa.ca
bardiaparmoun@cmail.carleton.ca
dylanleveille@cmail.carleton.ca

December 6th, 2024

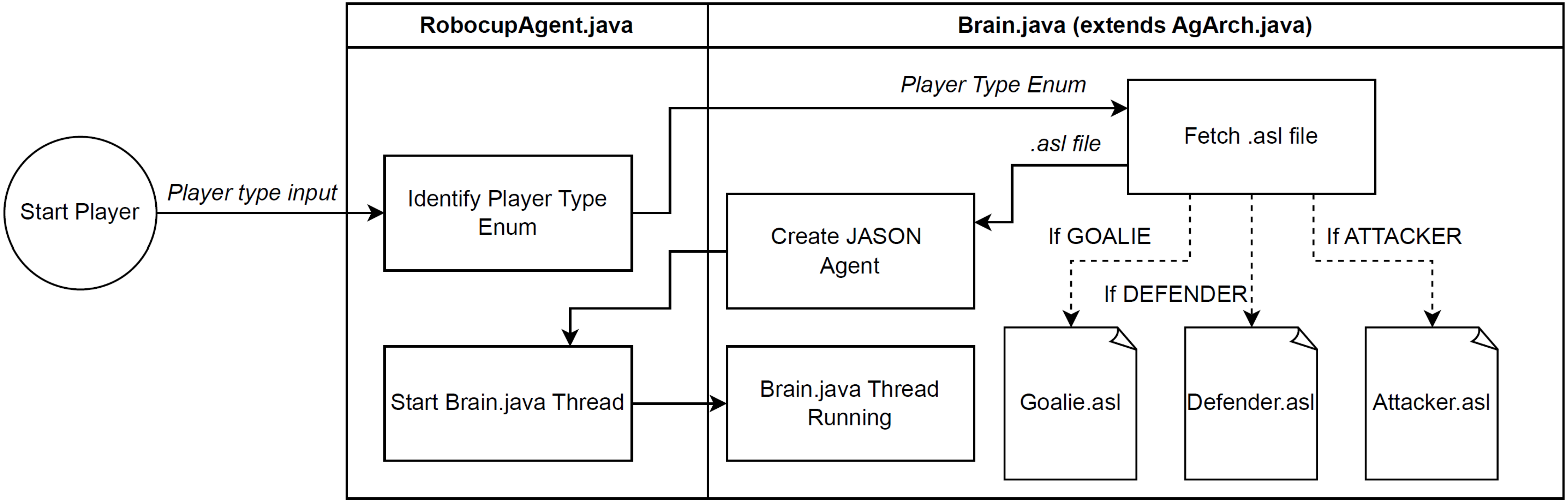
Outline

- Primary Contribution
- Program Design
- Player Functionalities
    - Goalie
    - Defender
    - Attacker
- Agent Software Design

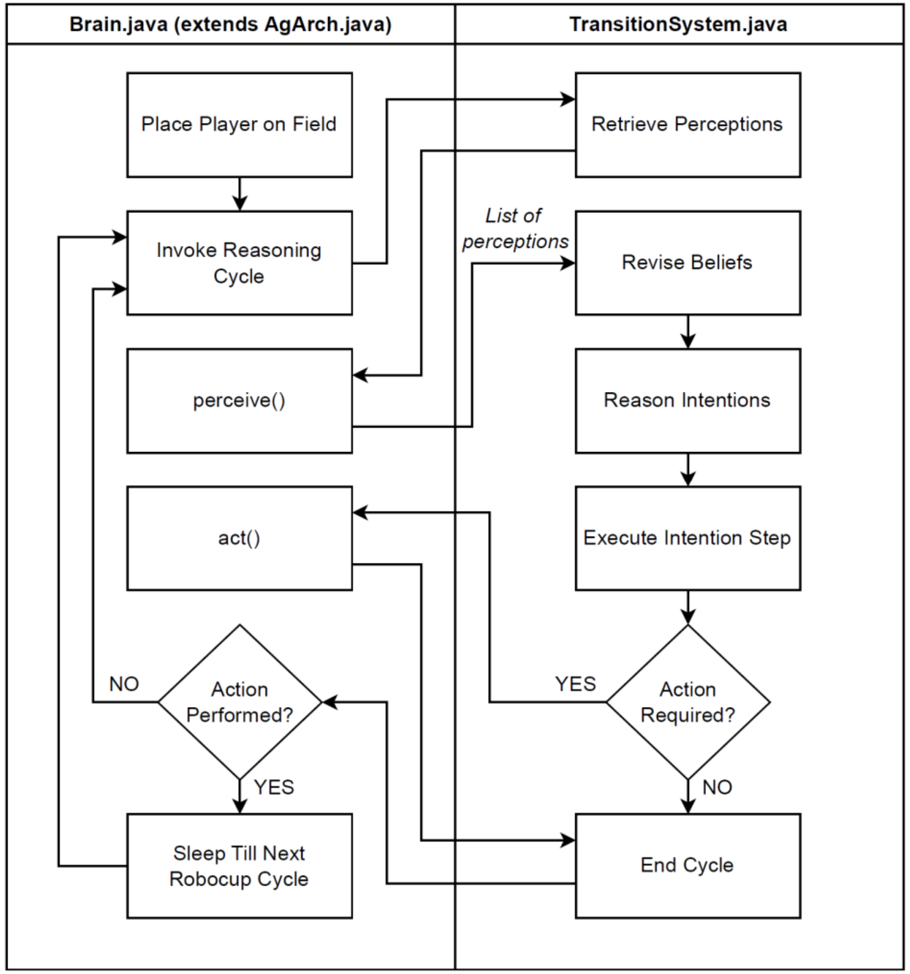
Primary Contribution

- The development of RoboCup players/agents that prioritize their positioning on the field.
  - Achieve by ensuring players maintain a heightened sense of spatial awareness!

Program Design - Initializing a RoboCup Player

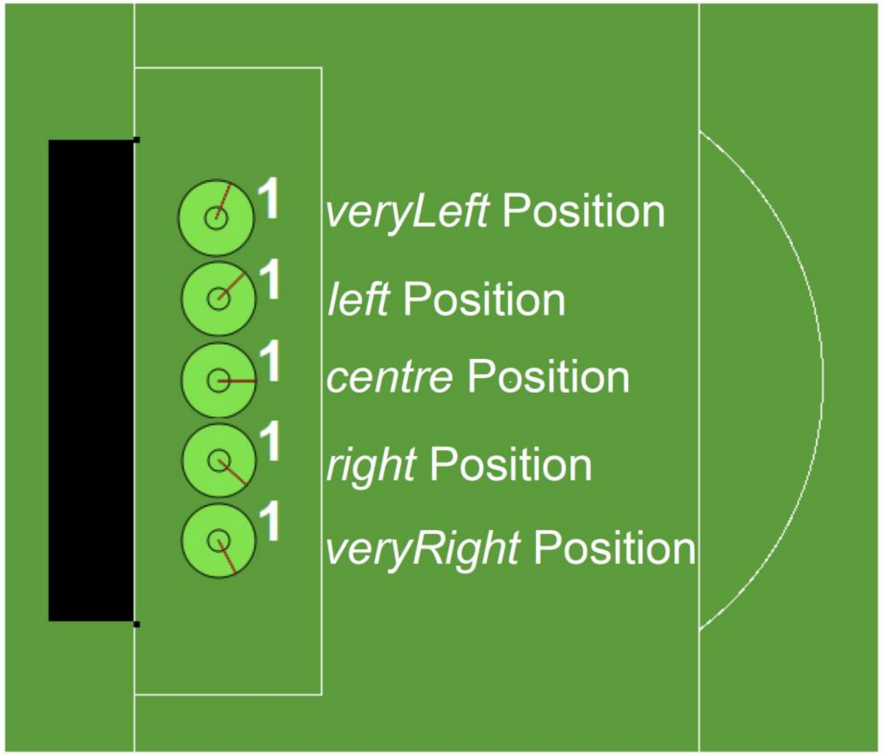


Program Design – Execution of a RoboCup Player



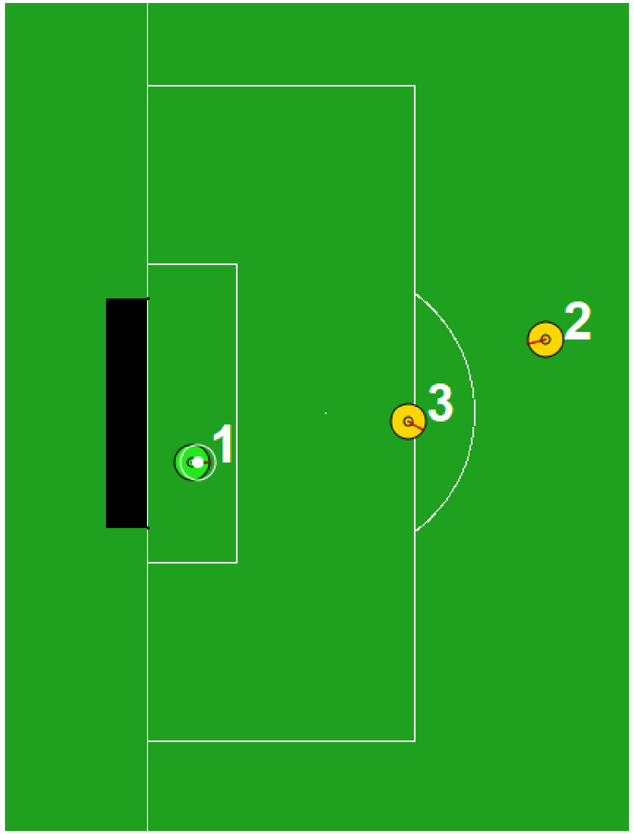
Player Functionalities/Behaviours – Goalie

- Initial Goal: !wait
- Change position based on ball angle.
  - centre: 20 degrees from centre flag (flag c 0)
  - left: 5 degrees from centre-top flag (flag c t)
  - right: 5 degrees from centre-bottom flag (flag c b)
  - veryLeft: 20 degrees from centre-top flag (flag c t)
  - veryRight: 20 degrees from centre-bottom flag (fla



Player Functionalities/Behaviours – Goalie

- Ball within 10 meters and angle to ball is 7.5 degrees
  - Run to ball, and attempt catch
- If caught:
  - Kick to furthest defender if available
  - Kick to nearest attacker if available
  - Else, kick at a 45 degree angle



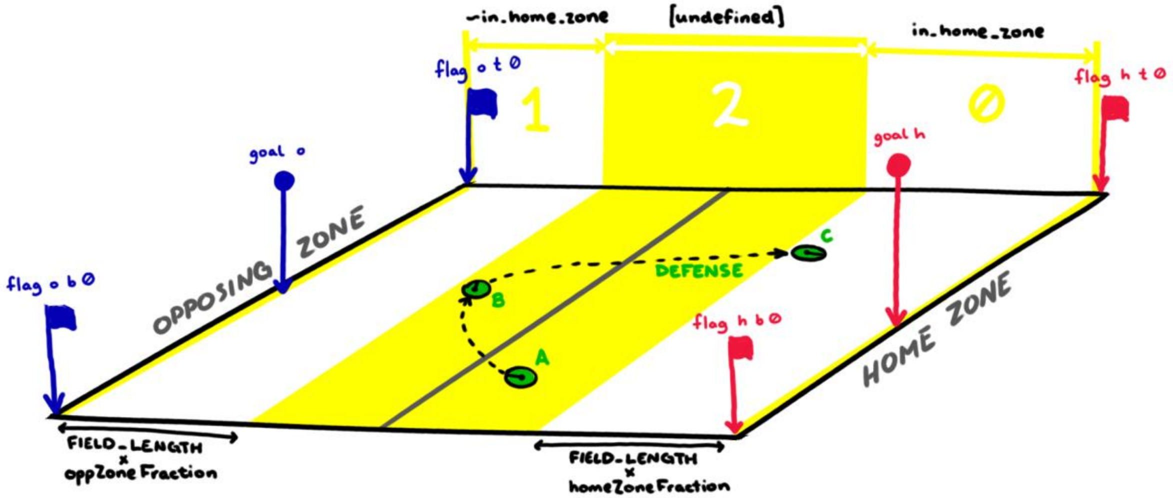
Player Functionalities/Behaviours – Defender

- Three main goals:
  - Get ball and score/pass → !wait
  - Stay in home zone → !run_to_home_zone
  - Wait for kickoff → !give_goalie_space

- !wait
  - Very similar to Krislet!
  - With the addition of passing to an attacker if no goal is visible

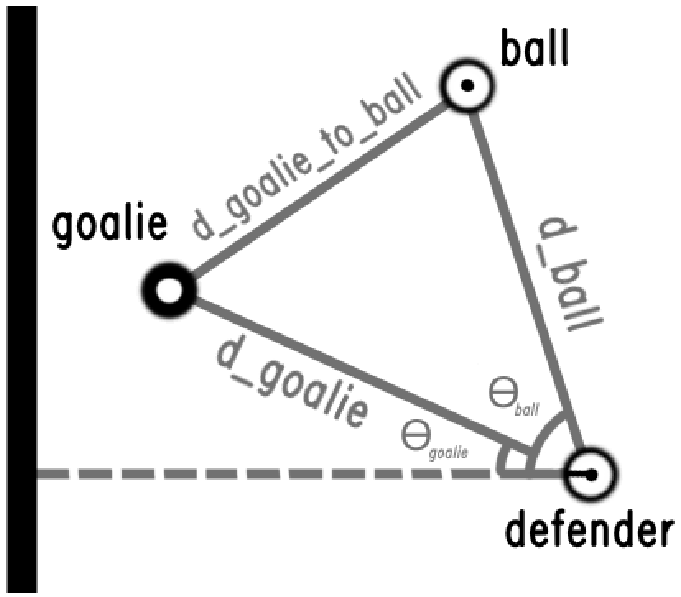
Player Functionalities/Behaviours – Defender

- !run_to_homezone
  1. Determine if you are outside home zone
  2. Find your own goal
  3. Align with the goal (± 40 degrees)
  4. Run until back in the zone



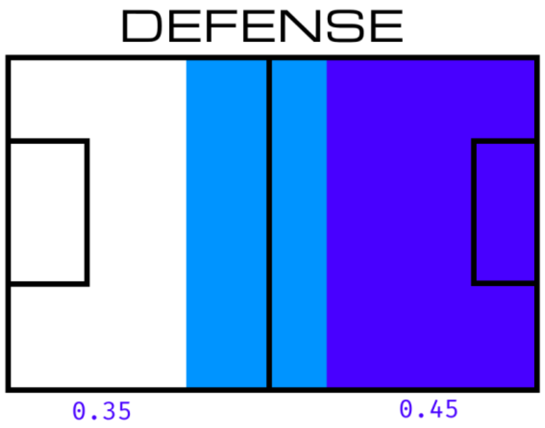
Player Functionalities/Behaviours – Defender

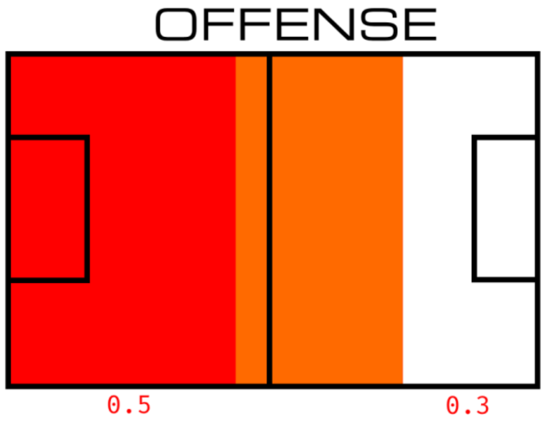
- !give_goalie_space
  1. Anticipate if goalie catches the ball
  2. Check if close to the goalie
  3. If close, head back to center
  4. Wait



Player Functionalities/Behaviours – Attacker

- Positioning & Aggression
  - Uses inverse positioning logic to defense
    - (`~in_home_zone` target)
  - Move back home and undefined zones to position the agent more effectively

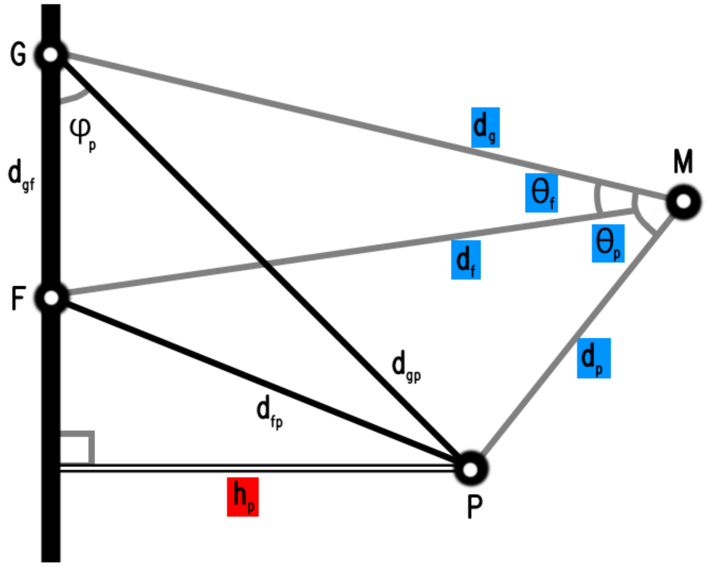




Carleton University

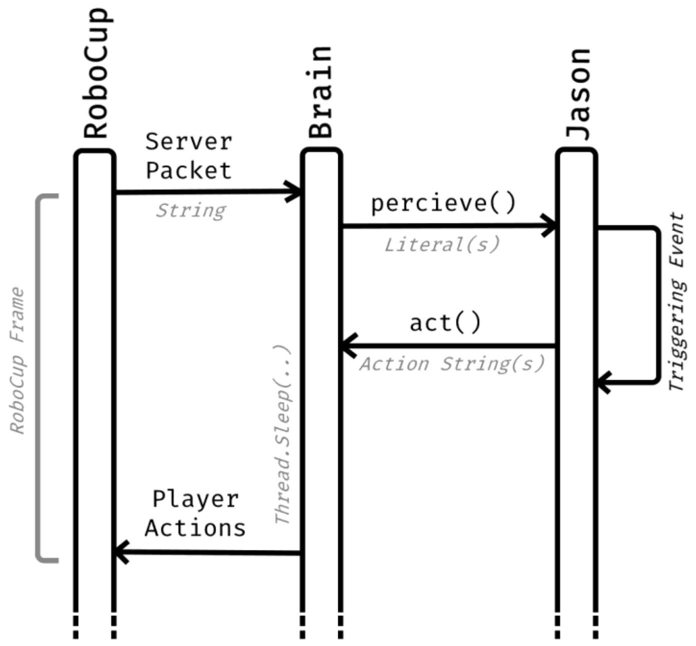
Player Functionalities/Behaviours – Attacker

- Offside Foul Mitigation
  - Prevent over-aggression by offence
  - Implemented as a simple belief to be processed by the Jason agent (`ball_offside`)
- If offside detected, run at reduced speed



Agent Software Design

- Focus on segmentation
  - Simplify belief inputs to Jason
  - Processing & Synchronization in Brain.java
- Issues
  - Coordinating between the two can be finicky
  - All logic must be bilaterally mirrored



Agent Software Design

- BDI Catchment and Pathways
  - Implementation of a "Hub-and-Spoke" control flow
  - Multiple hubs can be tied to one another (e.g., Goalie positions)
- Approach allows for greater control over belief catchment
- Comes at the cost of expandability for complex reasoning

Thank You