Predictive Act-R (PACT-R)

Using A Physics Engine and Simulation for Physical Prediction in a Cognitive Architecture

David Pentecost[1], Charlotte Sennersten[2], Robert Ollington[1], Craig A. Lindley[2], Byeong Kang[1]
Information and Communications Technology[1]
Robotic Systems and 3D Systems[2]
University of Tasmania[1]
CSIRO Data 61[2]
Hobart, Tasmania
Australia
email: davidp12@utas.edu.au
email: charlotte.sennersten@csiro.au
email: robert.ollington@utas.edu.au
email: craig.lindley@csiro.au
email: byeong.kang@utas.edu.au

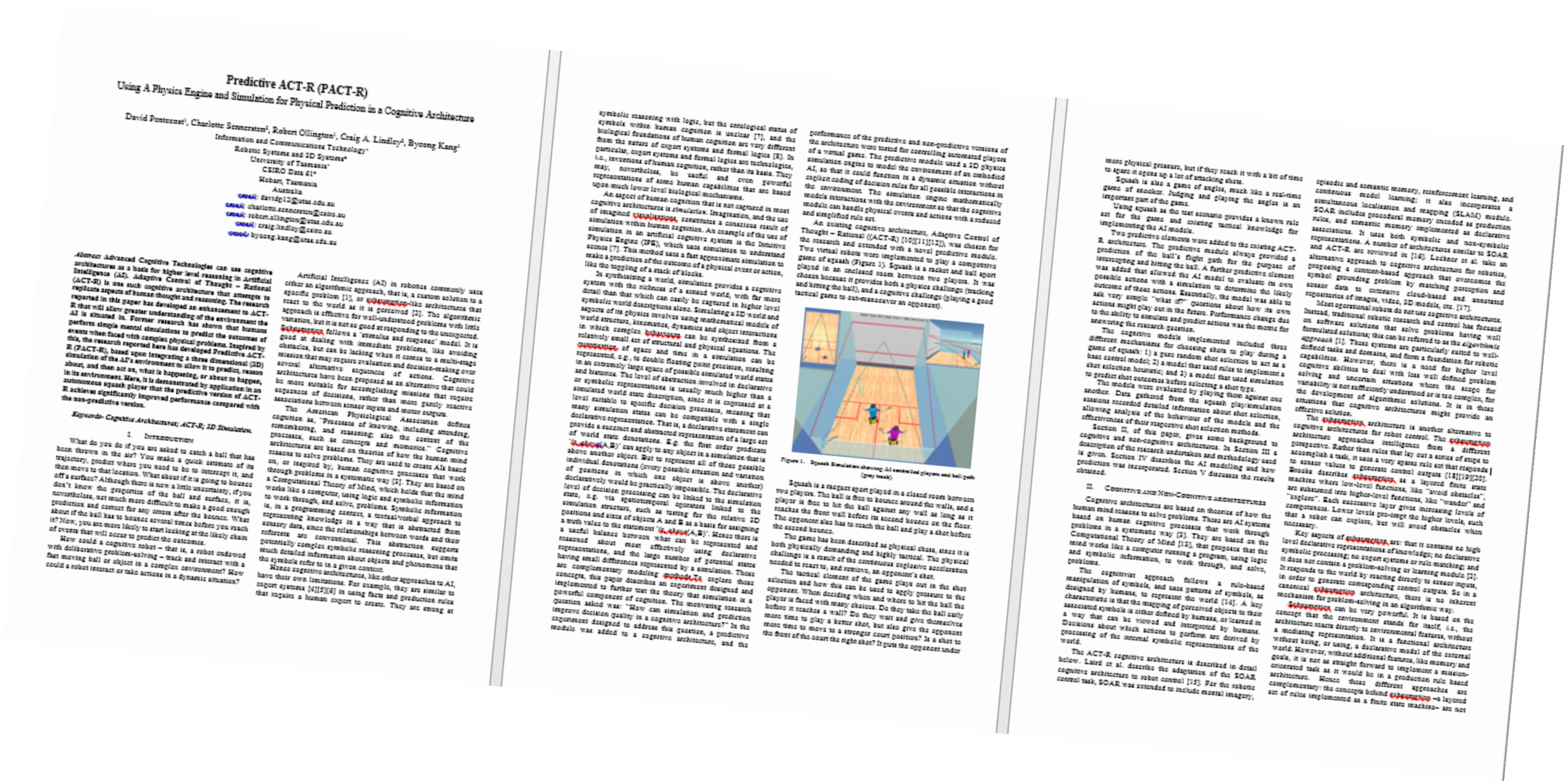
DAVID PENTECOST

Short intro

[redacted]

The Research Question:

How can simulation and prediction improve decision quality in a cognitive architecture?


Predictive ACT-R (PACT-R)
Using A Physics Engine and Simulation for Physical Prediction in a Cognitive Architecture

UNIVERSITY of TASMANIA

DATA 61

CSIRO

1. INTRODUCTION

The Research Question:

How can simulation and prediction improve decision quality in a cognitive architecture?

UNIVERSITY of TASMANIA

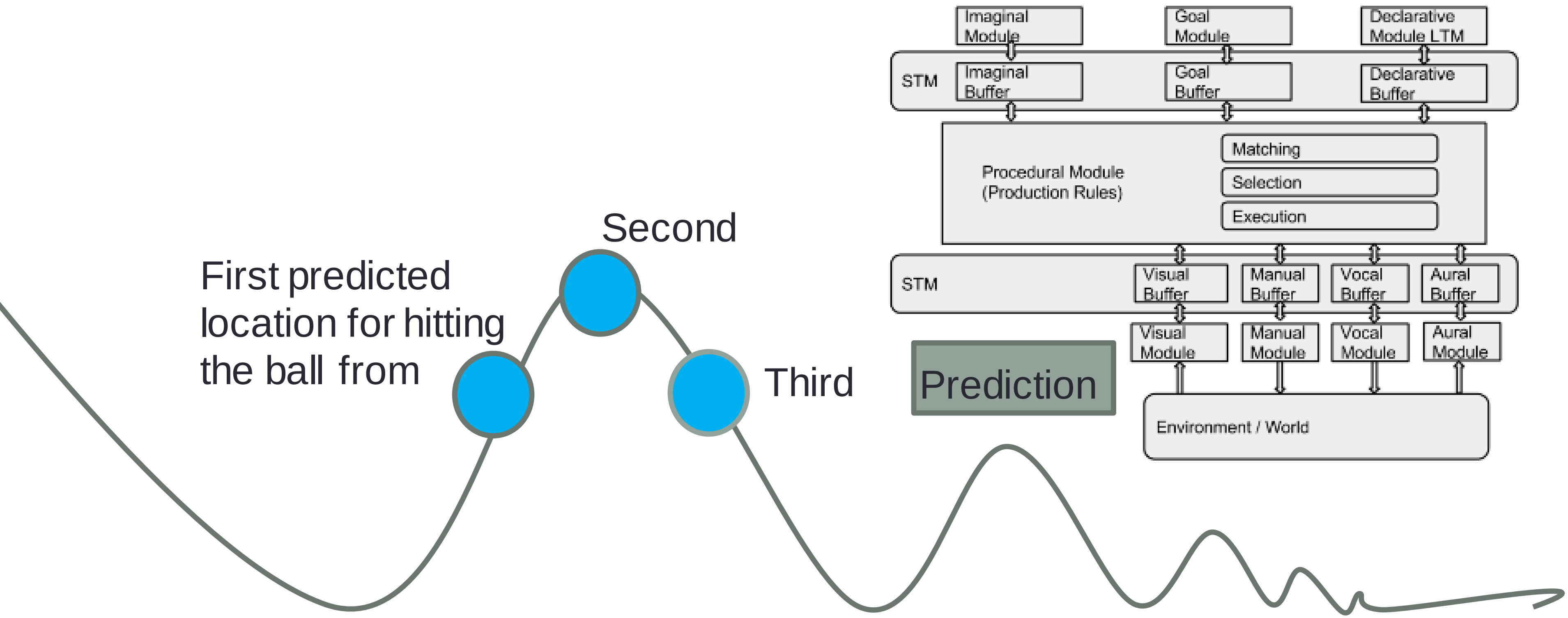
DATA 61

CSIRO

1. INTRODUCTION

The Research Question:

How can simulation and prediction improve decision quality in a cognitive architecture?

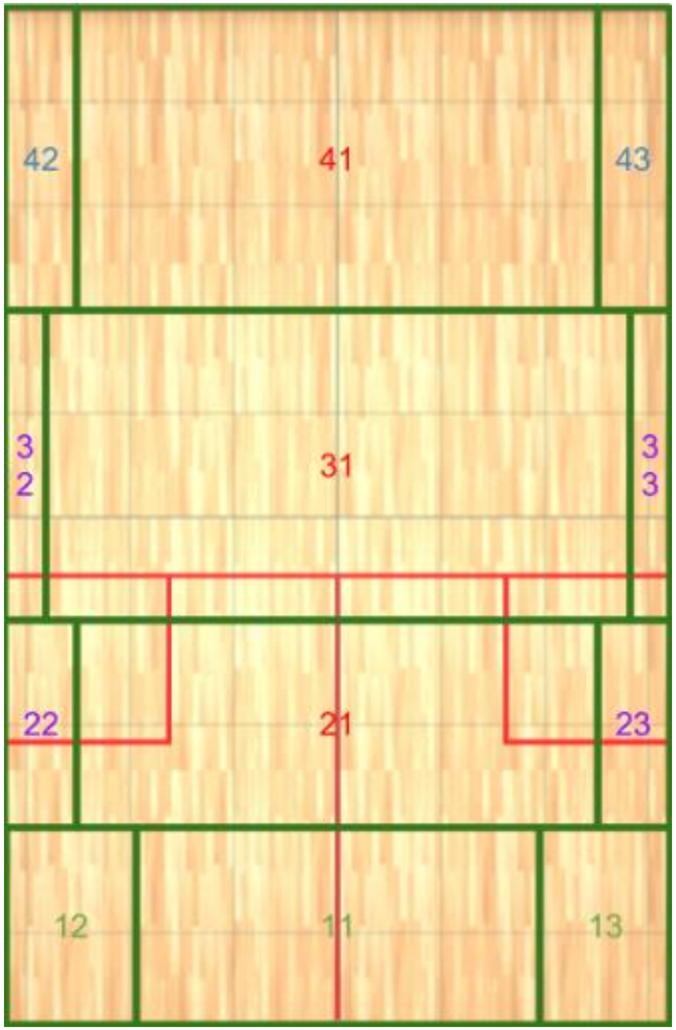


Zones For Shot Selection

Shot selection based on a limited choice per zone

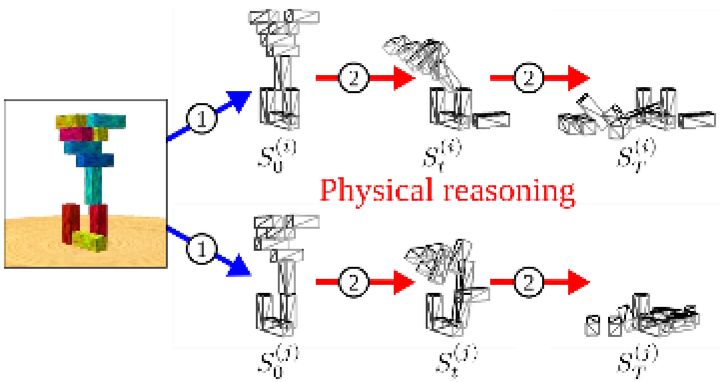
Based on Squash training patterns

Direction	Type	Description
Straight	Drop	Risky attacking shot, only played from a strong position
Cross Court	Drop	Very high risk attacking shot
Straight	Lob	Defensive high shot to back of court, safety shot gives time to recover
Cross Court	Lob	Defensive shot, safer than straight lob as there is less likelihood of the ball going out on the side wall
Boast	Lob (high)	High defensive boast. A shot played into the side wall that gives time to recover but often leads to a weak position for a player.
Boast	Short (low)	Attacking boast that stays low. Usually played from a strong position as a changeup shot.
Straight	Short	Attacking shot played low and hard that stays short and near the side wall.



1. INTRODUCTION

An aspect of human cognition that is not captured in most cognitive architectures is *simulation*. Imagination, and the use of imagined visualisations, constitutes a conscious result of simulation within human cognition. An example of the use of simulation in an artificial cognitive system is the Intuitive Physics Engine (IPE), which uses simulation to understand scenes [7].



Simulation as an engine of physical scene understanding

[Peter W. Battaglia](), [1] [Jessica B. Hamrick](), and [Joshua B. Tenenbaum]()

http://web.mit.edu/~pbatt/www/

Simulation For Scene Understanding

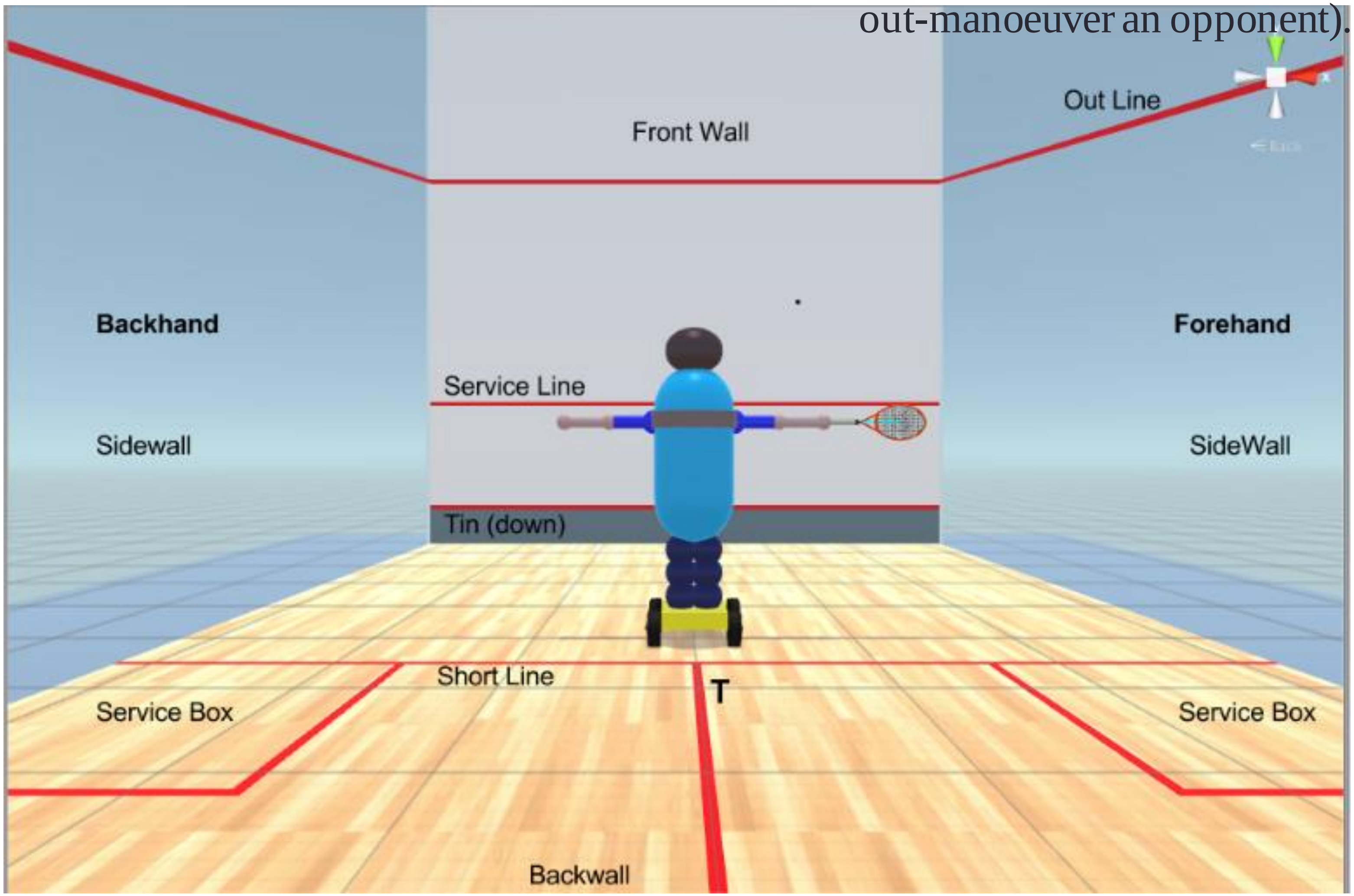
- Imperfect Perception
- Non Deterministic Dynamic Worlds
- (In mathematics and physics, a *deterministic* system is a system in which no randomness is involved in the development of future states of the system).

- Using a 3D Engine
- Low Fidelity Simulation
- How good does it need to be to be useful?

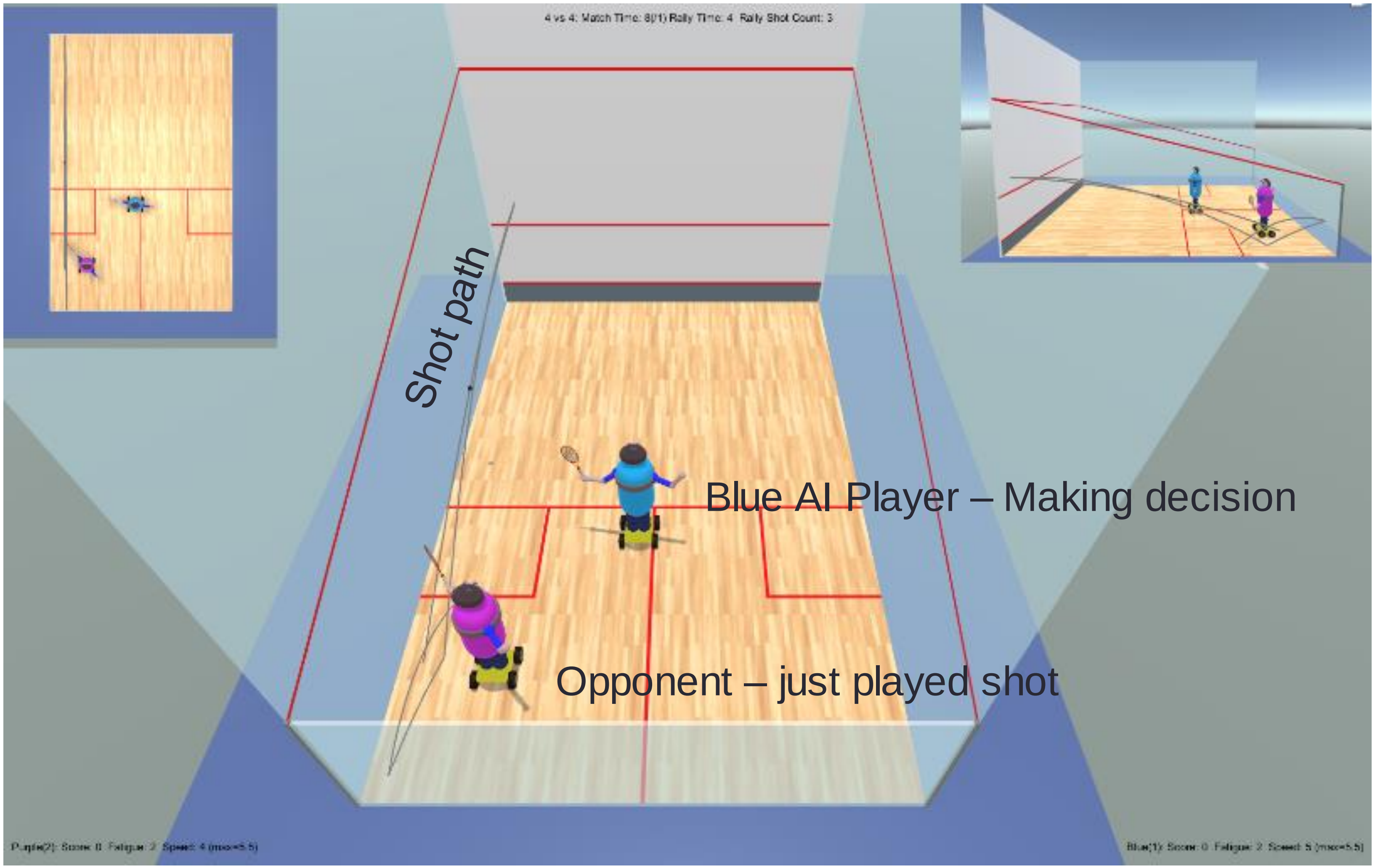
- Giving the AI Something to Reason With

Squash Terminology

It provides both a physics challenge (tracking and hitting the ball), and a cognitive challenge (playing a good tactical game to out-manoeuver an opponent).

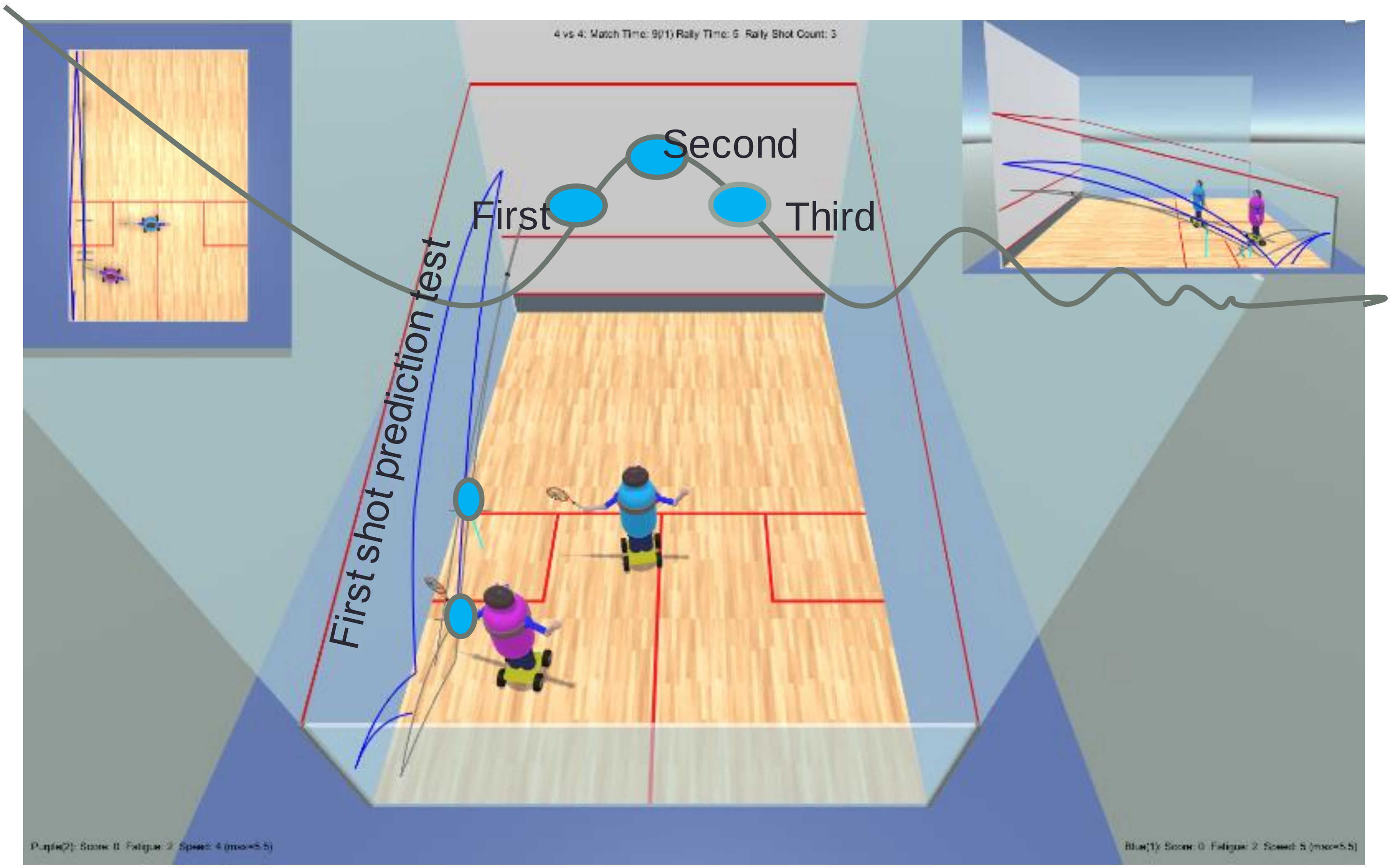


Predictive Thinking



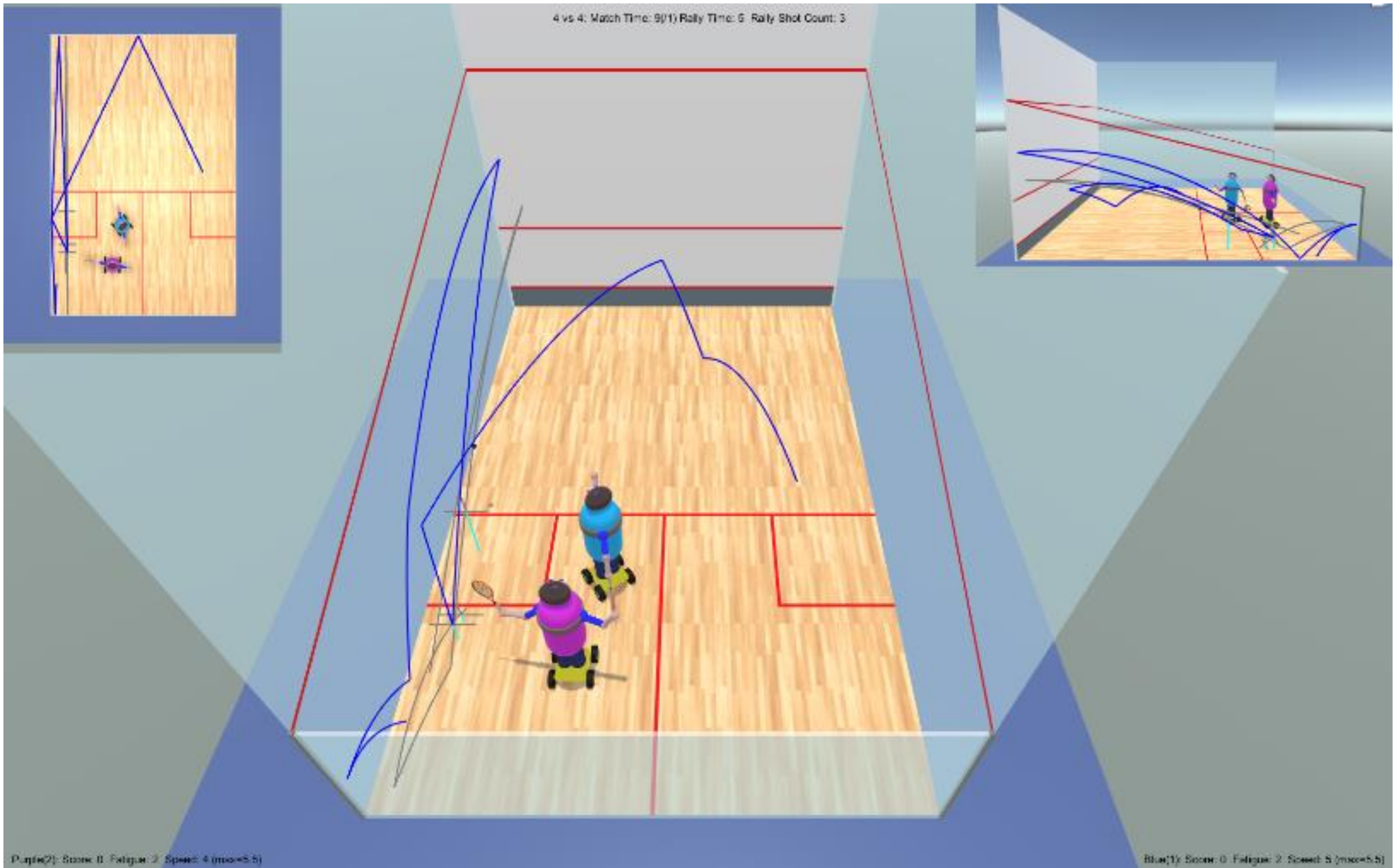
Predicting

9.050 PLAYER-ONE PROCEDURAL PRODUCTION-FIRED TEST-SHOT-Z22-Z23-STHI
Testing shot 51 0
better predicted value 2 for 51



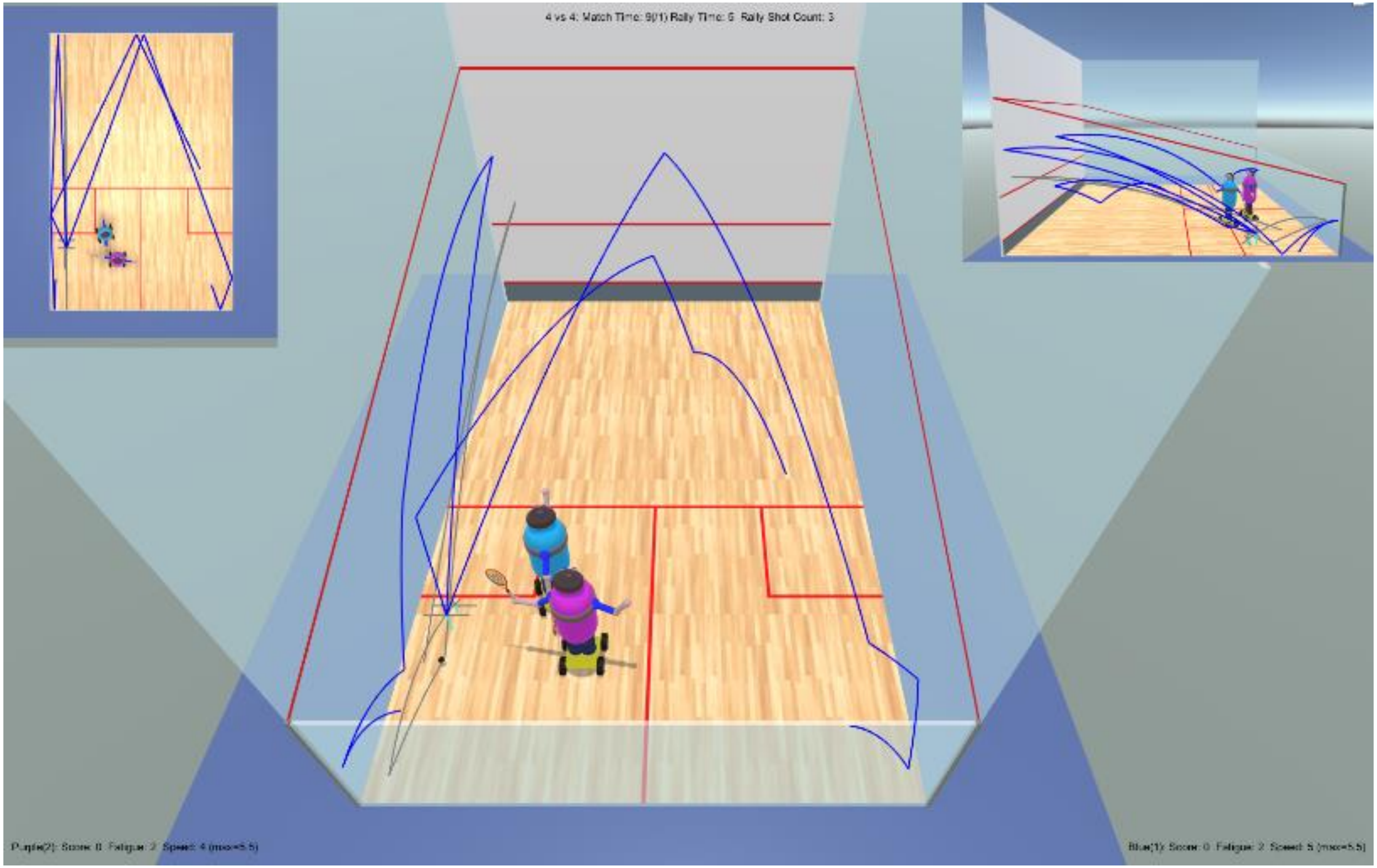
Predicting

9.250 PLAYER-ONE PROCEDURAL PRODUCTION-FIRED TEST-SHOT-Z22-Z23-BODF
Testing shot 23 1
predicted value 1 for 23



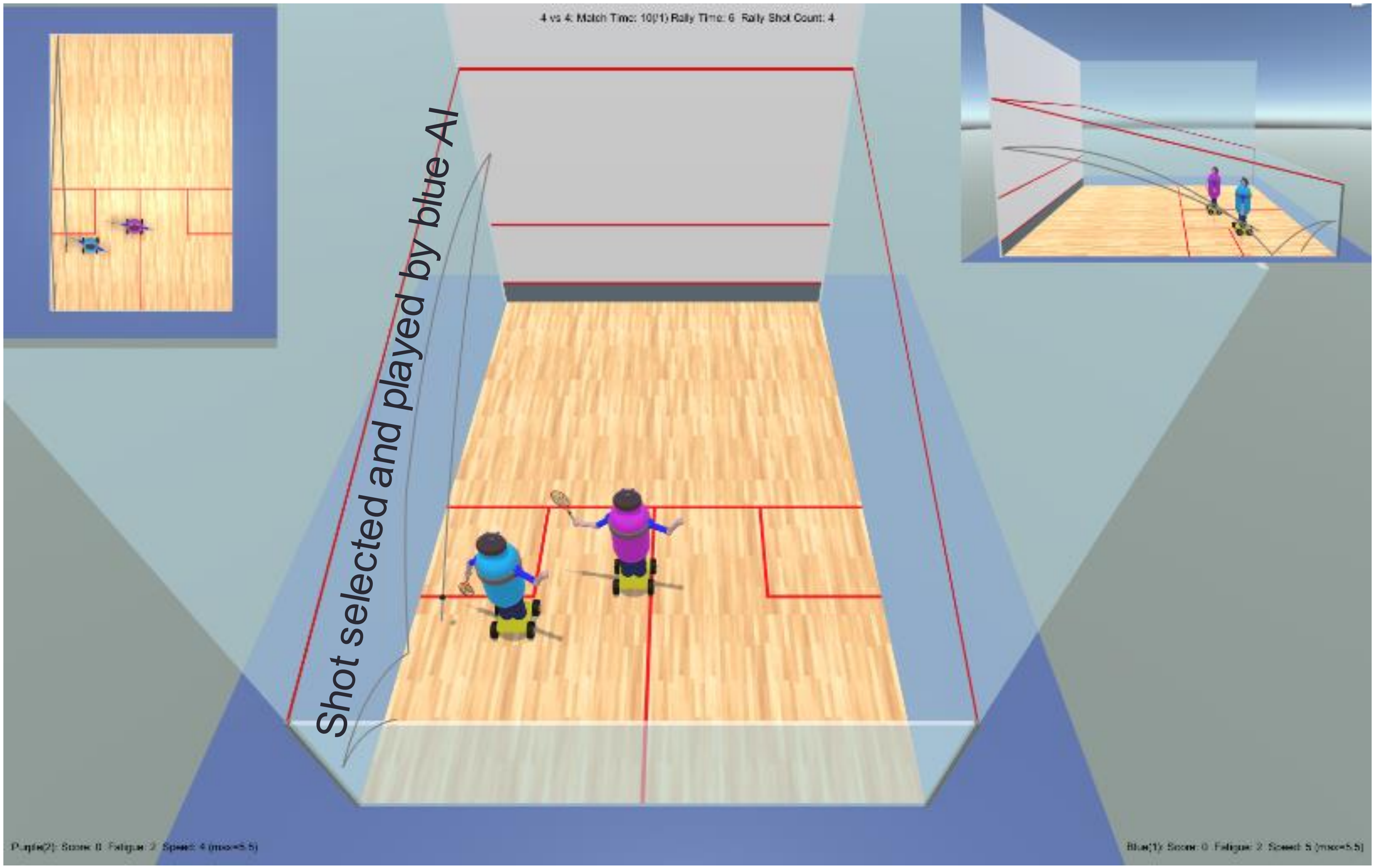
Predicting

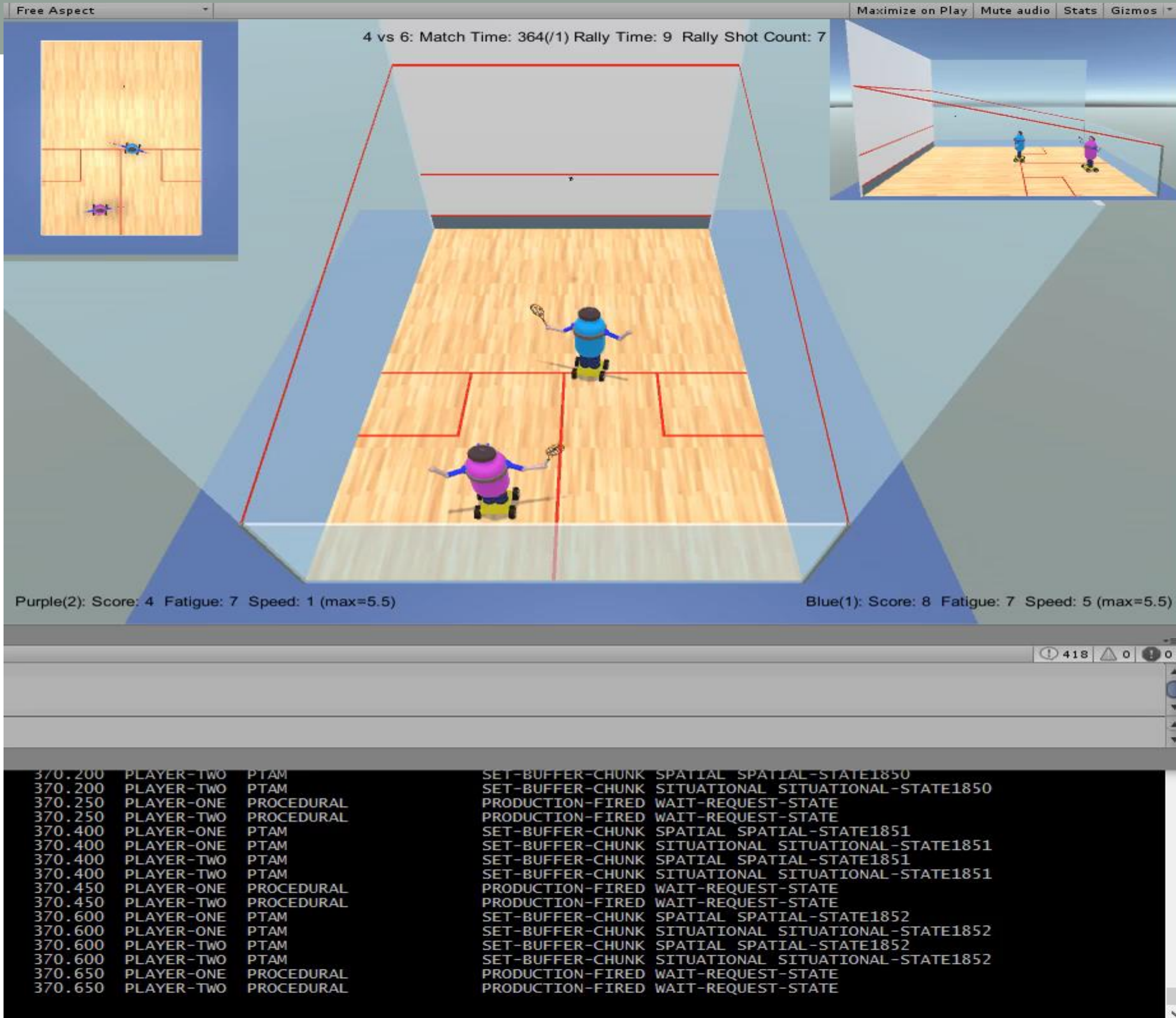
9.450 PLAYER-ONE PROCEDURAL PRODUCTION-FIRED TEST-SHOT-Z22-Z23-CRHI
Testing shot 52 2
predicted value 1 for 52



Predicting

9.850 PLAYER-ONE PROCEDURAL PRODUCTION-FIRED FINAL-SHOT-SELECTION




Free Aspect
Maximize on Play
Mute audio
Stats
Gizmos
4 vs 6: Match Time: 364(/1) Rally Time: 9 Rally Shot Count: 7
Purple(2): Score: 4 Fatigue: 7 Speed: 1 (max=5.5)
Blue(1): Score: 8 Fatigue: 7 Speed: 5 (max=5.5)
418
0
0
370.200 PLAYER-TWO PTAM SET-BUFFER-CHUNK SPATIAL SPATIAL-STATE1850
370.200 PLAYER-TWO PTAM SET-BUFFER-CHUNK SITUATIONAL SITUATIONAL-STATE1850
370.250 PLAYER-ONE PROCEDURAL PRODUCTION-FIRED WAIT-REQUEST-STATE
370.250 PLAYER-TWO PROCEDURAL PRODUCTION-FIRED WAIT-REQUEST-STATE
370.400 PLAYER-ONE PTAM SET-BUFFER-CHUNK SPATIAL SPATIAL-STATE1851
370.400 PLAYER-ONE PTAM SET-BUFFER-CHUNK SITUATIONAL SITUATIONAL-STATE1851
370.400 PLAYER-TWO PTAM SET-BUFFER-CHUNK SPATIAL SPATIAL-STATE1851
370.400 PLAYER-TWO PTAM SET-BUFFER-CHUNK SITUATIONAL SITUATIONAL-STATE1851
370.450 PLAYER-ONE PROCEDURAL PRODUCTION-FIRED WAIT-REQUEST-STATE
370.450 PLAYER-TWO PROCEDURAL PRODUCTION-FIRED WAIT-REQUEST-STATE
370.600 PLAYER-ONE PTAM SET-BUFFER-CHUNK SPATIAL SPATIAL-STATE1852
370.600 PLAYER-ONE PTAM SET-BUFFER-CHUNK SITUATIONAL SITUATIONAL-STATE1852
370.600 PLAYER-TWO PTAM SET-BUFFER-CHUNK SPATIAL SPATIAL-STATE1852
370.600 PLAYER-TWO PTAM SET-BUFFER-CHUNK SITUATIONAL SITUATIONAL-STATE1852
370.650 PLAYER-ONE PROCEDURAL PRODUCTION-FIRED WAIT-REQUEST-STATE
370.650 PLAYER-TWO PROCEDURAL PRODUCTION-FIRED WAIT-REQUEST-STATE

2. COGNITIVE AND NON-COGNITIVE ARCHITECTURES

more physical pressure, but if they reach it with a bit of time to spare it opens up a lot of attacking shots.

Squash is also a game of angles, much like a real-time game of snooker. Judging and playing the angles is an important part of the game.

Using squash as the test scenario provides a known rule set for the game and existing tactical knowledge for implementing the AI models.

Two predictive elements were added to the existing ACT-R architecture. The predictive module always provided a prediction of the ball's flight path for the purpose of intercepting and hitting the ball. A further predictive element was added that allowed the AI model to evaluate its own possible actions with a simulation to determine the likely outcome of those actions. Essentially, the model was able to ask very simple "what if?" questions about how its own actions might play out in the future. Performance change due to the ability to simulate and predict actions was the metric for answering the research question.

The cognitive models implemented included three different mechanisms for choosing shots to play during a game of squash: 1) a pure random shot selection to act as a base control model; 2) a model that used rules to implement a shot selection heuristic; and 3) a model that used simulation to predict shot outcomes before selecting a shot type.

The models were evaluated by playing them against one another. Data gathered from the squash play/simulation sessions recorded detailed information about shot selection, allowing analysis of the behaviour of the models and the effectiveness of their respective shot selection methods.

Section II, of this paper, gives some background to cognitive and non-cognitive architectures. In Section III a description of the research undertaken and methodology used is given. Section IV describes the AI modelling and how prediction was incorporated. Section V discusses the results obtained.

II. Cognitive and Non-Cognitive Architectures

Cognitive architectures are based on theories of how the human mind reasons to solve problems. These are AI systems based on human cognitive processes that work through problems in a systematic way [3]. They are based on the Computational Theory of Mind [13], that proposes that the mind works like a computer running a program, using logic and symbolic information, to work through, and solve, problems.

The cognitivist approach follows a rule-based manipulation of symbols, and uses patterns of symbols, as designed by humans, to represent the world [14]. A key characteristic is that the mapping of perceived objects to their associated symbols is either defined by humans, or learned in a way that can be viewed and interpreted by humans. Decisions about which actions to perform are derived by processing of the internal symbolic representations of the

episodic and semantic memory, reinforcement learning, and continuous model learning, it also incorporates a simultaneous localisation and mapping (SLAM) module. SOAR includes procedural memory encoded as production rules, and semantic memory implemented as declarative associations. It uses both symbolic and non-symbolic representations. A number of architectures similar to SOAR and ACT-R are reviewed in [16]. Lochner et al. take an alternative approach to cognitive architecture for robotics, proposing a content-based approach that overcomes the symbol grounding problem by matching perception and sensor data to extensive cloud-based and annotated repositories of images, video, 3D models, etc. [17].

Most operational robots do not use cognitive architectures. Instead, traditional robotic research and control has focused on software solutions that solve problems having well formulated solutions; this can be referred to as the *algorithmic approach* [1]. These systems are particularly suited to well-defined tasks and domains, and form a foundation for robotic capabilities. However, there is a need for higher level cognitive abilities to deal with less well defined problem solving and uncertain situations where the scope for variability is not sufficiently understood or is too complex, for the development of algorithmic solutions. It is in these situations that cognitive architectures might provide an effective solution.

The subsumption architecture is another alternative to cognitive architectures for robot control. The subsumption architecture approaches intelligence from a different perspective. Rather than rules that lay out a series of steps to accomplish a task, it uses a very sparse rule set that responds to sensor values to generate control outputs [18][19][20]. Brooks describes subsumption as a layered finite state machine where low-level functions, like "avoid obstacles" are subsumed into higher-level functions, like "wander" and "explore". Each successive layer gives increasing levels of competences. Lower levels pre-empt the higher levels, such that a robot can explore, but will avoid obstacles when necessary.

Key aspects of subsumption are: that it contains no high level declarative representations of knowledge; no declarative symbolic processing; no expert systems or rule matching; and it does not contain a problem-solving or learning module [2]. It responds to the world by reacting directly to sensor inputs, in order to generate corresponding control outputs. So in a canonical subsumption architecture, there is no inherent mechanism for problem-solving in an algorithmic way.

Subsumption can be very powerful. It is based on the concept that the environment stands for itself, i.e., the architecture reacts directly to environmental features, without a mediating representation. It is a functional architecture without being, or using, a declarative model of the external world. However, without additional features, like memory and

difficult to implement, and could be easily incorporated into other cognitive architectures.

Society of Mind proposes a theory that intelligence arises from the interactions of large numbers of simple functions [21][22]. This is not an actual architecture, but rather a theory [23] that argues against the idea that a single unified architecture or solution can account for intelligent behaviour.

A robotic AI can be created completely within a single architecture, using rules that control every aspect of the decision making process, but those architectures are not always ideal for every style of decision-making. Society of Mind theory argues for a modular approach to implementing an intelligence. Implementing simulation as an extension to a cognitive architecture, but using an external 3D engine to model the environment, follows this concept. The simulation is a separate, specialised function for solving problems in dynamic physical situations.

ACT-R is a hybrid cognitive architecture consisting of both symbolic and sub-symbolic components [24][25]. It is a goal-orientated architecture. The symbolic data consists of facts and production rules. The sub-symbolic data is metadata about facts and production rules that control which facts are recalled and which production rules are chosen to fire when multiple facts and rules are available.

ACT-R consists of a number of modules that interact through a production system that selects rules to execute, (Figure 2). Each module has a buffer, which can hold a chunk of data (a key/value pair structure) representing the current state of that module.

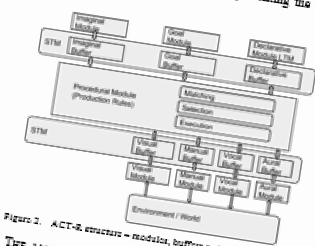


Figure 2. ACT-R structure – modules, buffers and production system.

The matching system looks for patterns in the buffers that it can use to select a production rule to potentially fire from amongst those available. Each production rule includes a pattern that gives the conditions under which it can fire. Production rules can make requests of the modules, so they can change their own internal state.

ACT-R to constitute the Predictive-ACT-R (PACT-R) architecture.

A. Research Design

The research consisted of developing and implementing a virtual environment for testing; developing a cognitive module that implemented the simulation-based cognition system; and developing AI models to test the system.

An ACT-R cognitive module was developed that mapped a symbolic representation of a simulated environment into the ACT-R framework. This module gave the required PACT-R functionality for interpreting and acting within the environment, as well as providing simple predictive capabilities using simulation.

The use of prediction and simulation in ACT-R was evaluated by comparing the performance of several models that each implemented different levels of prediction. The aim was to compare not only their performance, but also how easy/simple it was to model and use a predictive AI.

B. Implementation

The research consisted of three components. The first was the design and implementation of a cognitive module within ACT-R. This module gave models access to predictions about physical events, as well as a mechanism to take actions.

The second element was a simulation of the game of squash implemented in the Unity™ game engine. Parts of the PACT-R module were also implemented with Unity™, and communicated with the prediction module in PACT-R. The Unity™ components of ACT-R were the physics simulation and prediction engine.

The final element was modelling squash-playing AIs. Three evaluation models were developed for testing and cross-comparison.

C. Using Simulation and Prediction within a Cognitive Architecture

The research investigated the use of a physics engine to provide prediction for a cognitive architecture. The concept requires a physics engine that can model and simulate the environment of a robot controlled by a cognitive AI. The simulation provides a symbolic representation of the environment to a cognitive architecture. This gives the cognitive model (the production rules) the information it needs to understand and act within its environment.

One way of using this information is to explicitly encode rules that check for certain conditions, for example, whether an object is in a certain position, or is moving in a particular direction; or for the relationships between objects in the environment, for example, whether an object is to the left of another object [17][26]. From this, the rules can encode appropriate actions for the robot to take.

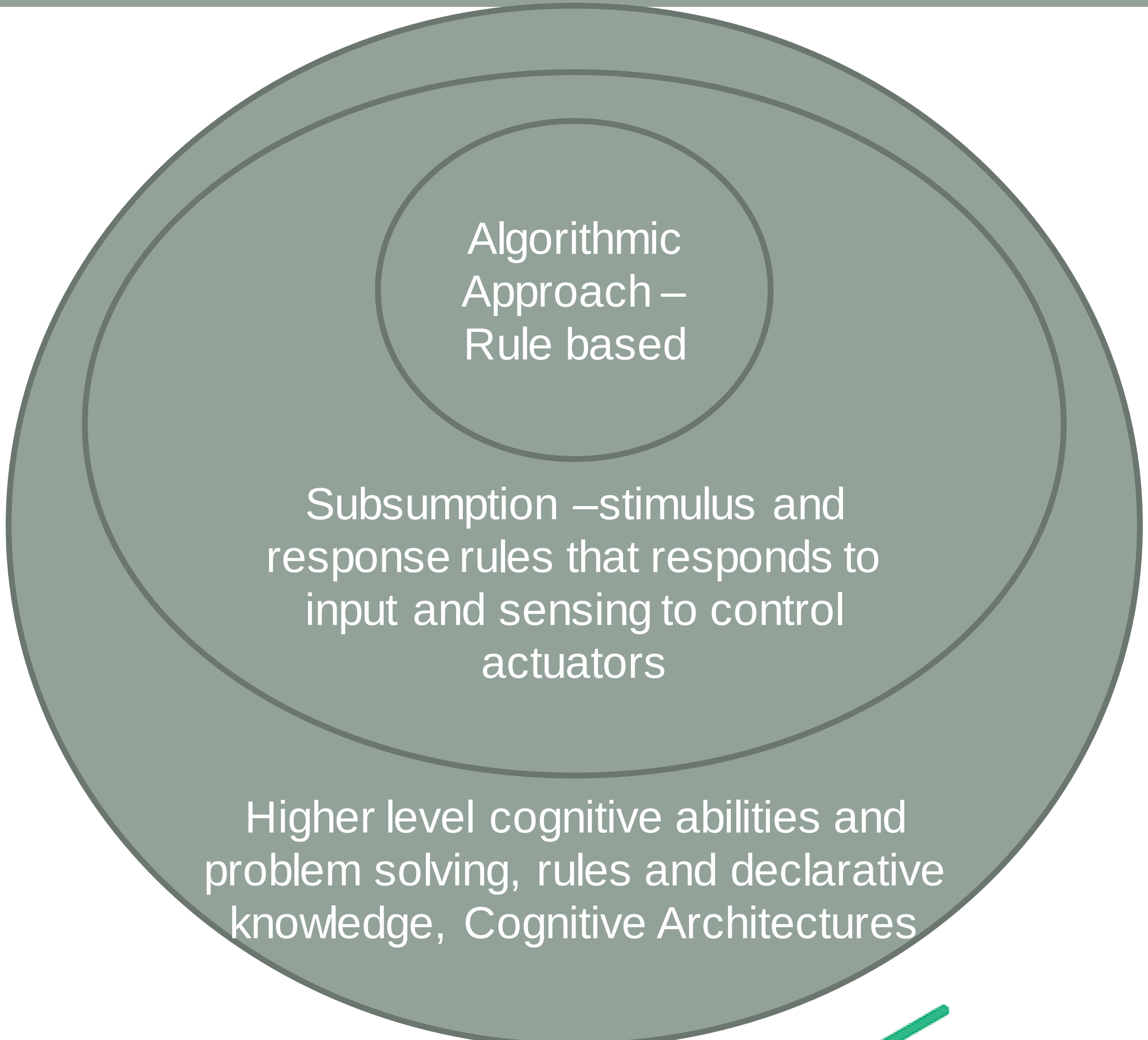
This research explored an alternative approach. Rather than using explicit rules to interpret and decide actions, a

UNIVERSITY of TASMANIA

DATA 61

CSIRO

2. COGNITIVE AND NON-COGNITIVE ARCHITECTURES

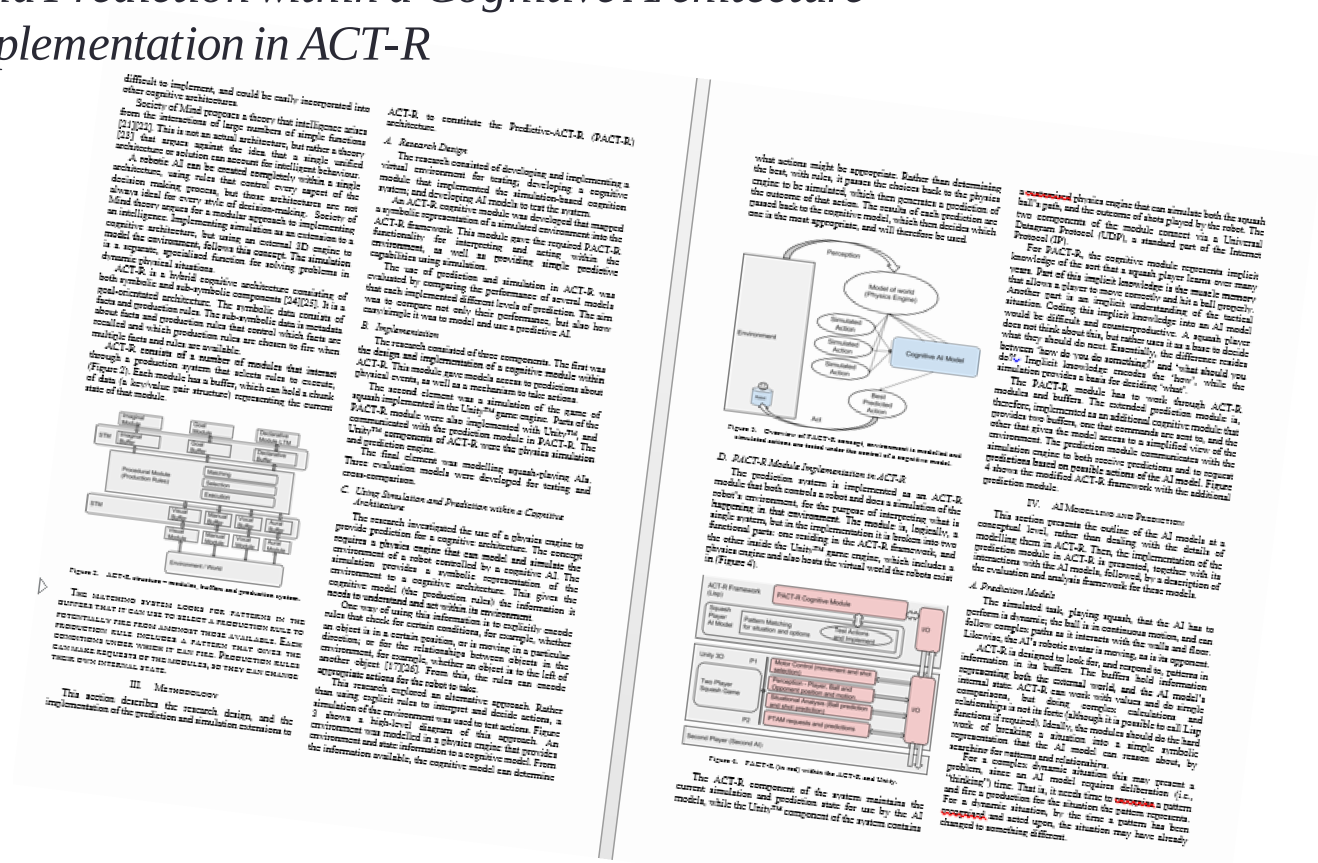


A. Research Design

B. Implementation

C. Using Simulation and Prediction within a Cognitive Architecture

D. PACT-R Module Implementation in ACT-R



UNIVERSITY of TASMANIA

DATA 61

CSIRO

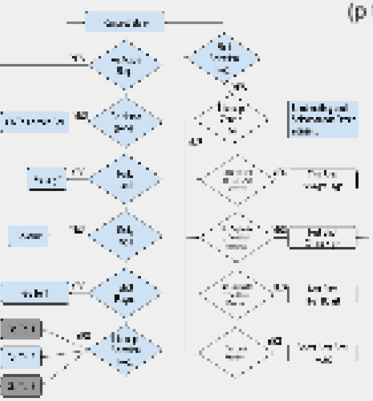
Research Design

- Experimental software development

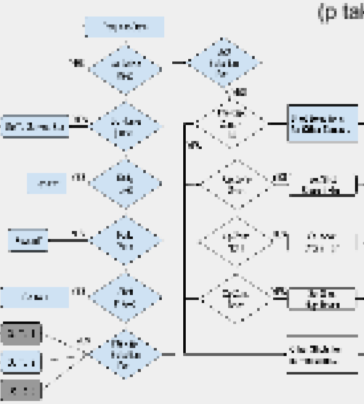
- Implement a new cognitive module for ACT-R
- Module gives an interpretation of a simulated world.
- Implement multiple ACT-R AI models and have them compete head to head
- Implement a suitable simulated environment and scenario

- Base line AI model and enhanced heuristic models for comparative testing
- Model with speculative prediction that tests prospective actions for positive or negative outcomes
- Statistical analysis of comparative results.

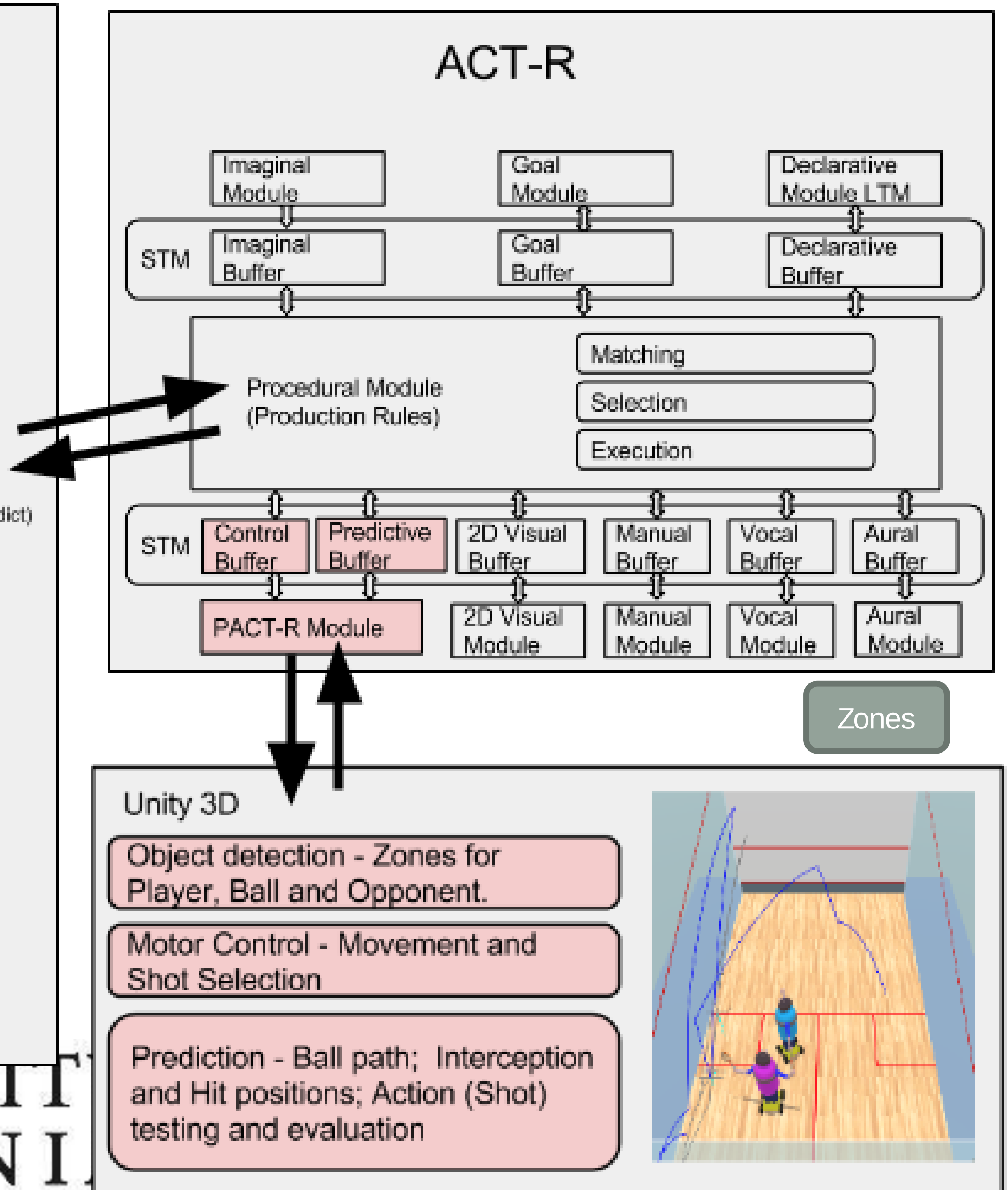
Rules

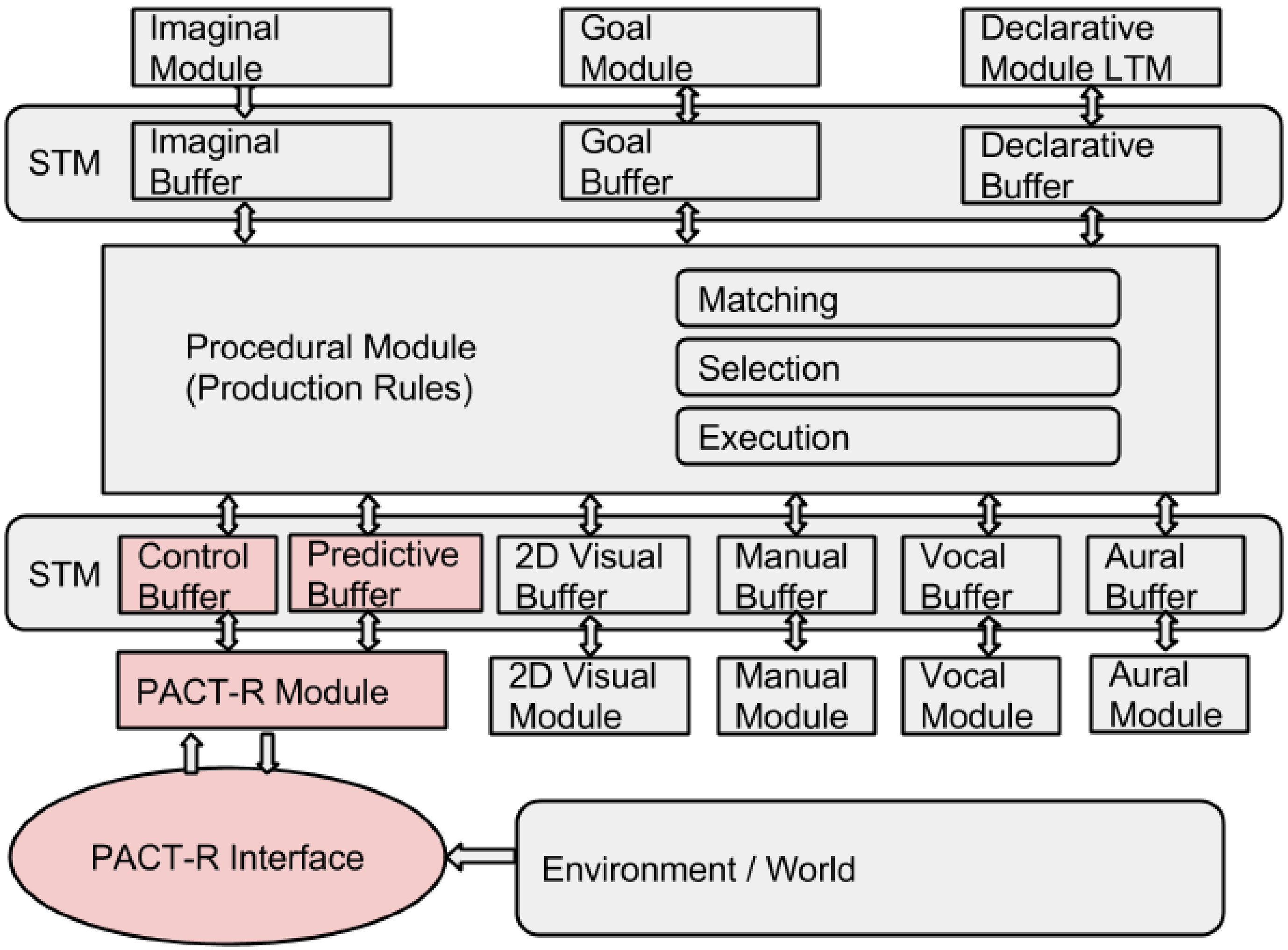


```
(p take-shot-z22-z23-StHi
  =goal>
   ISA playing-mode
   state    2              ; in play mode
  ?command>
   state free
  =predictive>             ; PACT-R module
   ISA predictive-state    ; correct chunk
   special 5               ; in prediction mode
   < prediction-count 4 ; more testing allowed
   - registered-shot 51 ; not already tested
   > intercept-zone-width 1 ; court pos wide
   intercept-zone-depth 2   ; and mid depth
 ==>
  +command>
   ISA command-packet
   req-cmd              5      ; Test Shot (predict)
   :req-param  51      ; Long High Straight
```



```
(p take-shot-z22-z23-StHi-OpSh
  =goal>
  ISA playing-mode
  state  2     ; play mode
  ?command>
  state free
  =predictive>   ; PACT-R module
  ISA predictive-state ; correct chunk type
  special 5              ; shot selection mode
  > intercept-zone-width 1 ; position wide
   intercept-zone-depth 2 ; position middle
   > op-zone-depth 2  ; op at front of court
 ==>
  +command>
   ISA command-packet
   req-cmd            4       ; Set Shot to play
   :req-param  51      ; Long High Straight
```



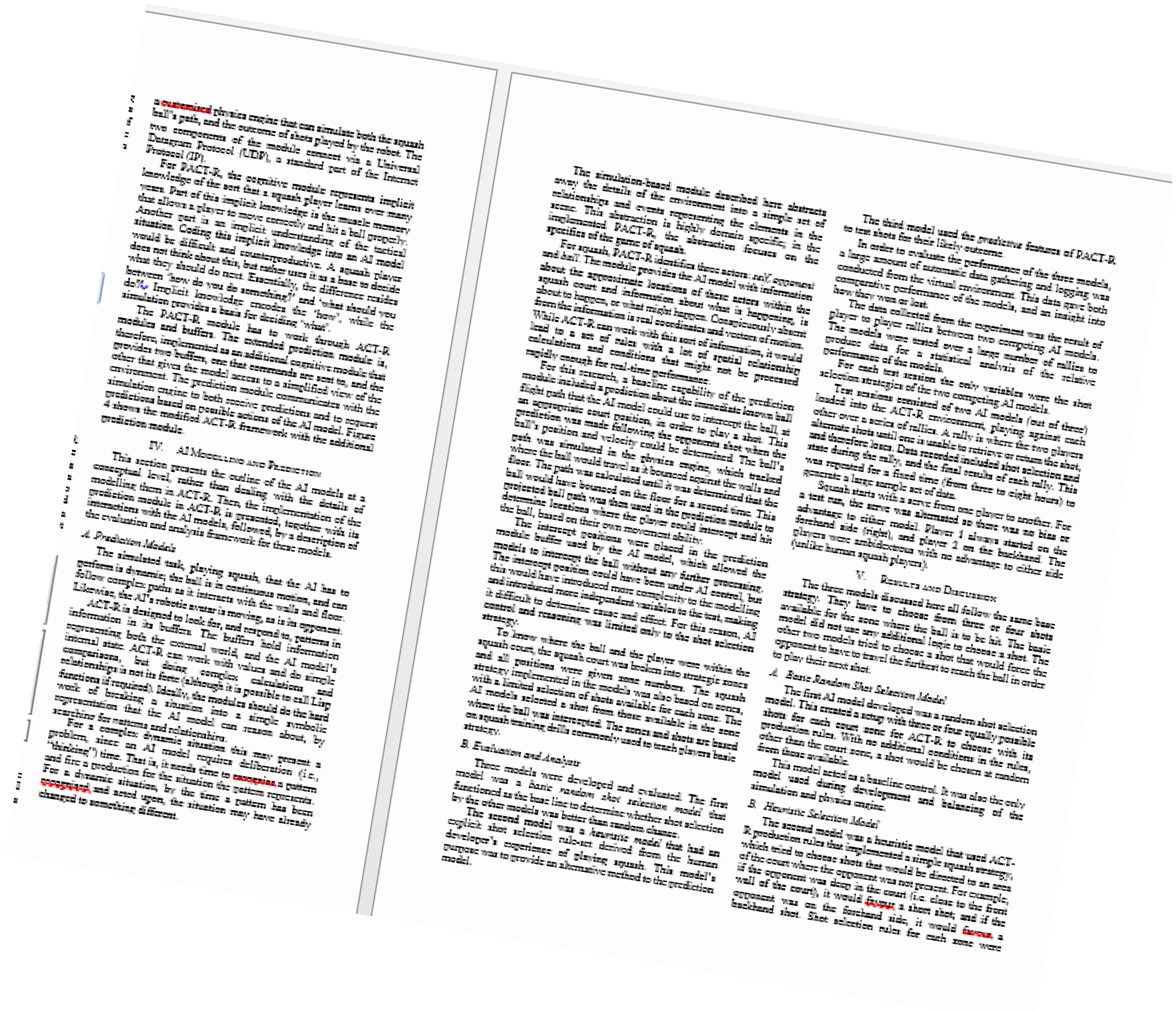


UNIVERSITY of TASMANIA

DATA 61 CSIRO

A. Prediction Models

B. Evaluation and Analysis



UNIVERSITY *of* TASMANIA

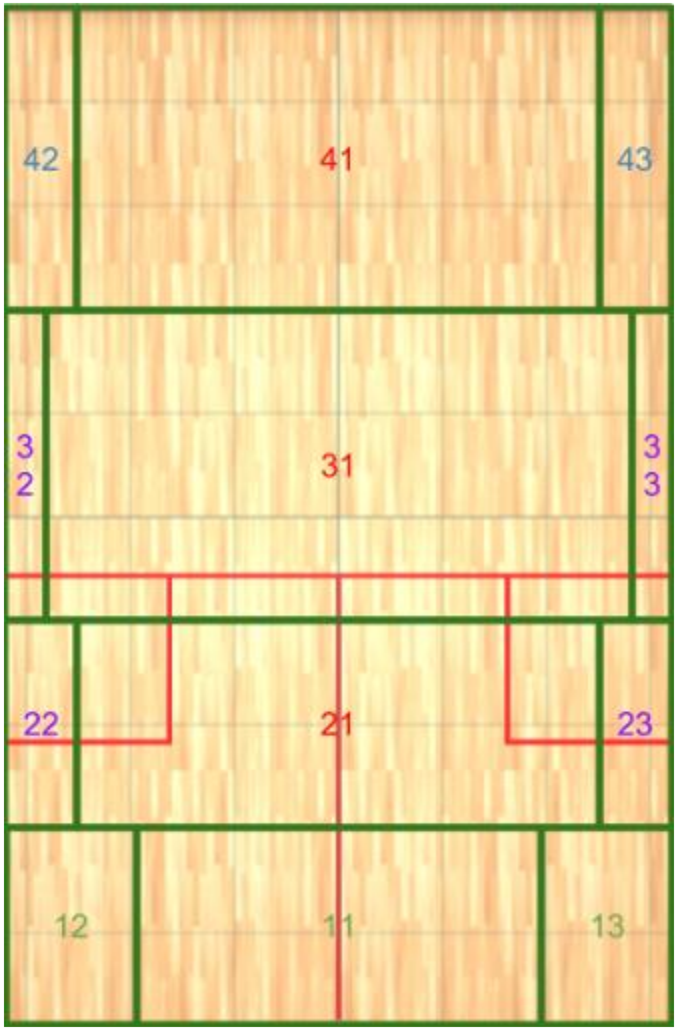
DATA 61 CSIRO

Three models compared:

1. **Random** Squash Shot Selection Model
2. **Heuristic** Squash Shot Selection Model
3. **Predictive** Squash Shot Selection Model

A. Basic Random Shot Selection Model
B. Heuristic Selection Model
C. Predictive Selection Model
D. Performance

The first AI model developed was a random shot selection model. This created a setup with three or four equally possible shots for each court zone for ACT-R to choose with its production rules. With no additional conditions in the rules, other than **the court zone**, a shot would be chosen at random from those available. This model acted as a baseline control. It was also the only model used during development and balancing of the simulation and physics engine.

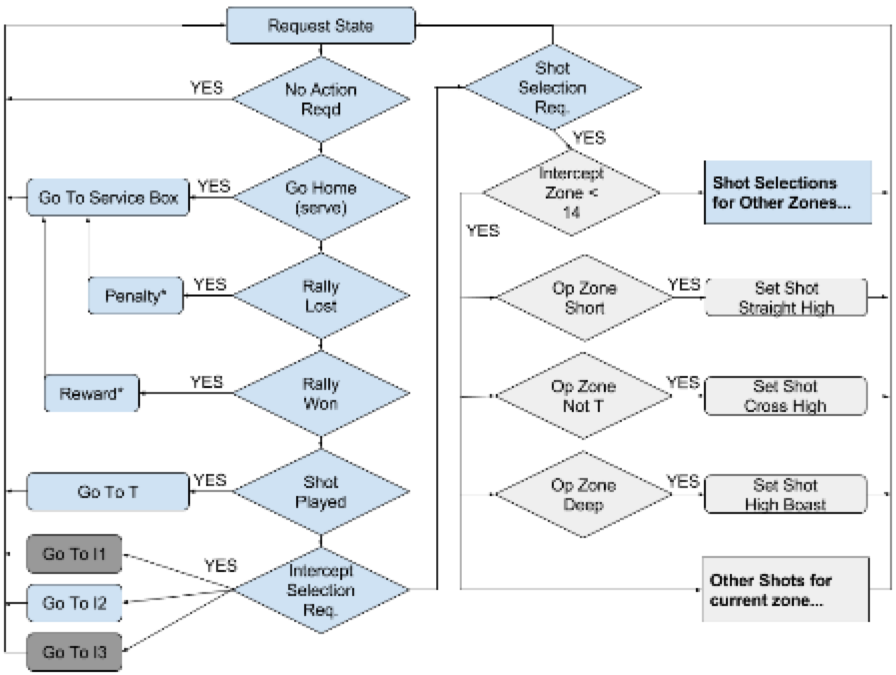


UNIVERSITY of TASMANIA

DATA 61

CSIRO

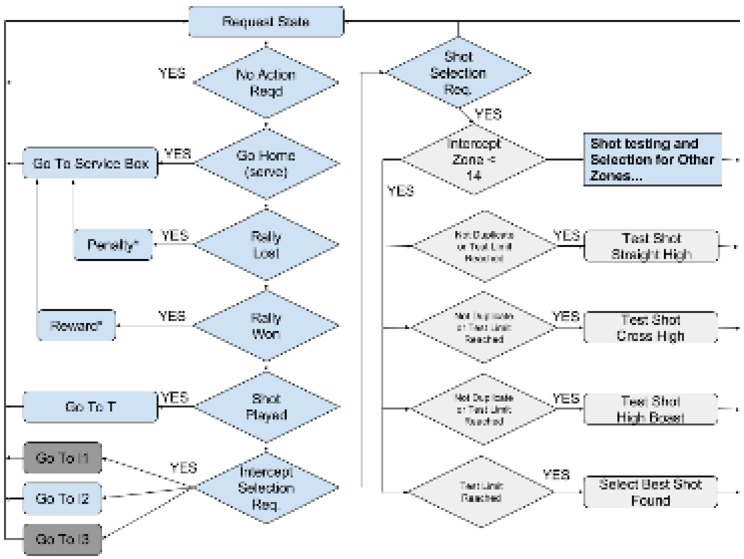
The second model was a heuristic model that used ACT-R **production rules that implemented a simple squash strategy**, which tried to choose **shots that would be directed to an area of the court where the opponent was not present**. For example, if the opponent was deep in the court (i.e. close to the front wall of the court), it would favour a short shot; and if the opponent was on the forehand side, it would favour a backhand shot. Shot selection rules for each zone were implemented using this simple strategy. In real squash, this approach is a good starting point for any human player.



Heuristic Production Rules in LISP

```
(p take-shot-z22-z23-StHi-OpSh
        =goal>
                ISA playing-mode
                 state    2                        ; model is playing?
        ?command>
                state free                         ; module free to accept command?
        =predictive>
                ISA spatial-state
                special 5                          ; shot selection requested?
              > intercept-zone-width 1             ; ball is wide?
                intercept-zone-depth 2             ; ball is middle of court?
              > op-zone-depth 2                    ; opponent is short?
    ==>
        +command>
                ISA command-packet
                req-cmd      4                     ; Set Shot
                :req-param   51                    ; Long High Straight
```

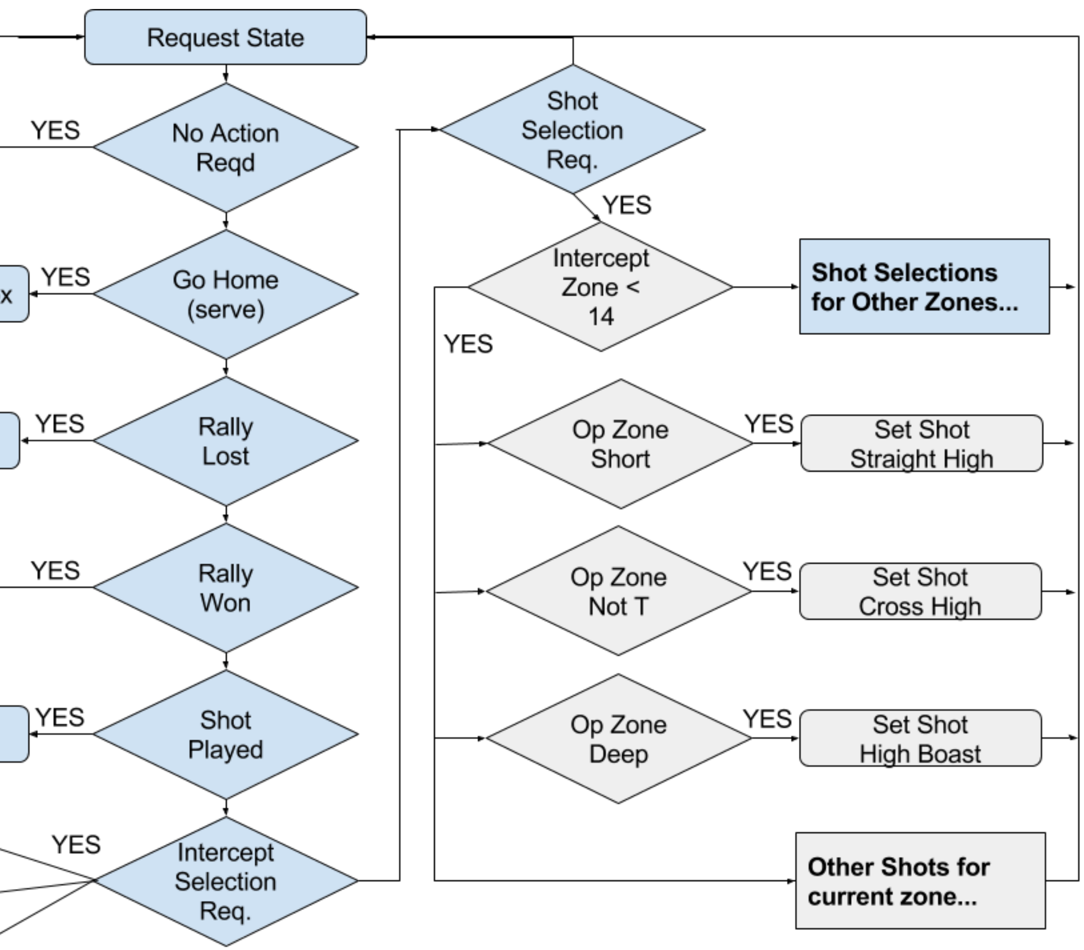
The predictive model went a step further in predicting the outcome of shots the AI model might take. This was done by allowing **the AI model to choose a possible shot before passing that information to the prediction module for simulating and predicting its consequences.** The module would simulate how the shot would play out to predict where the opponent would be when the shot was played, and how much difficulty they would have in then retrieving it and playing a counter shot. The prediction was based on the same strategy as the heuristic model, **trying to find a shot that was as far from the opponent as possible.**



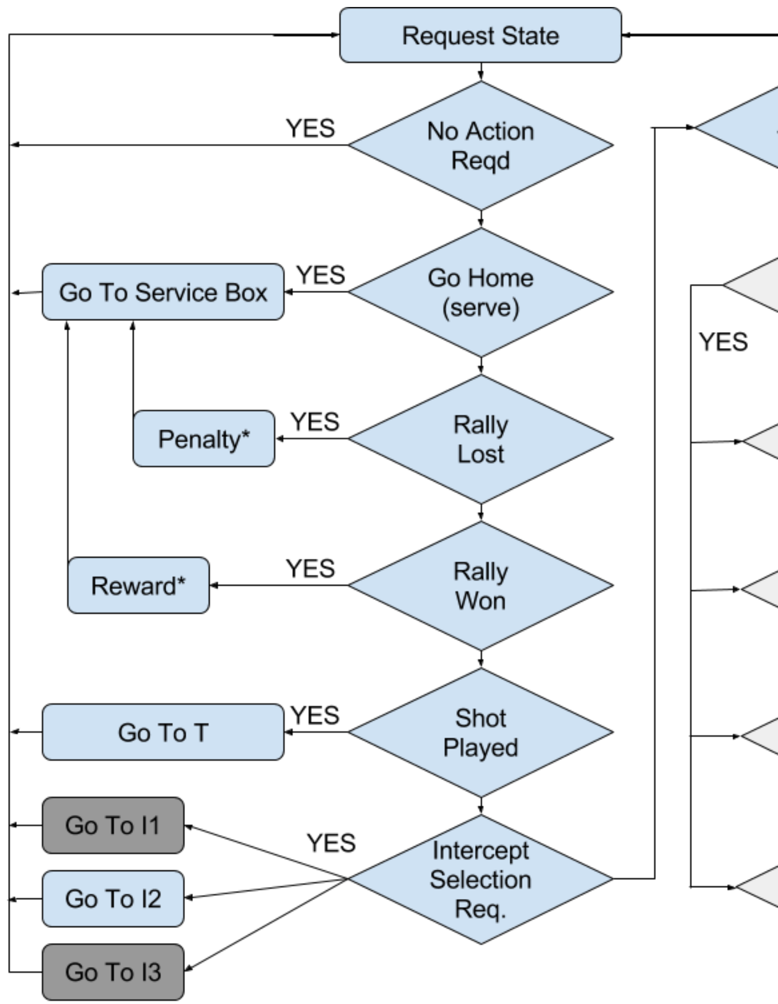
Predictive Production Rule

```
(p test-shot-z22-z23-StHi
        =goal>
                ISA playing-mode
                 state    2                        ; model is playing?
        ?command>
                state free                         ; module free to accept command?
        =predictive>
                ISA spatial-state
                special 5
              < prediction-count 4                 ; can test more shots?
               - registered-shot 51                ; this shot not just tested?
              > intercept-zone-width 1             ; ball is wide?
                intercept-zone-depth 2             ; ball is middle of court?
   ==>
        +command>
                ISA command-packet
                req-cmd     5                      ; Test Shot
                :req-param  51                     ; Long High Straight
```

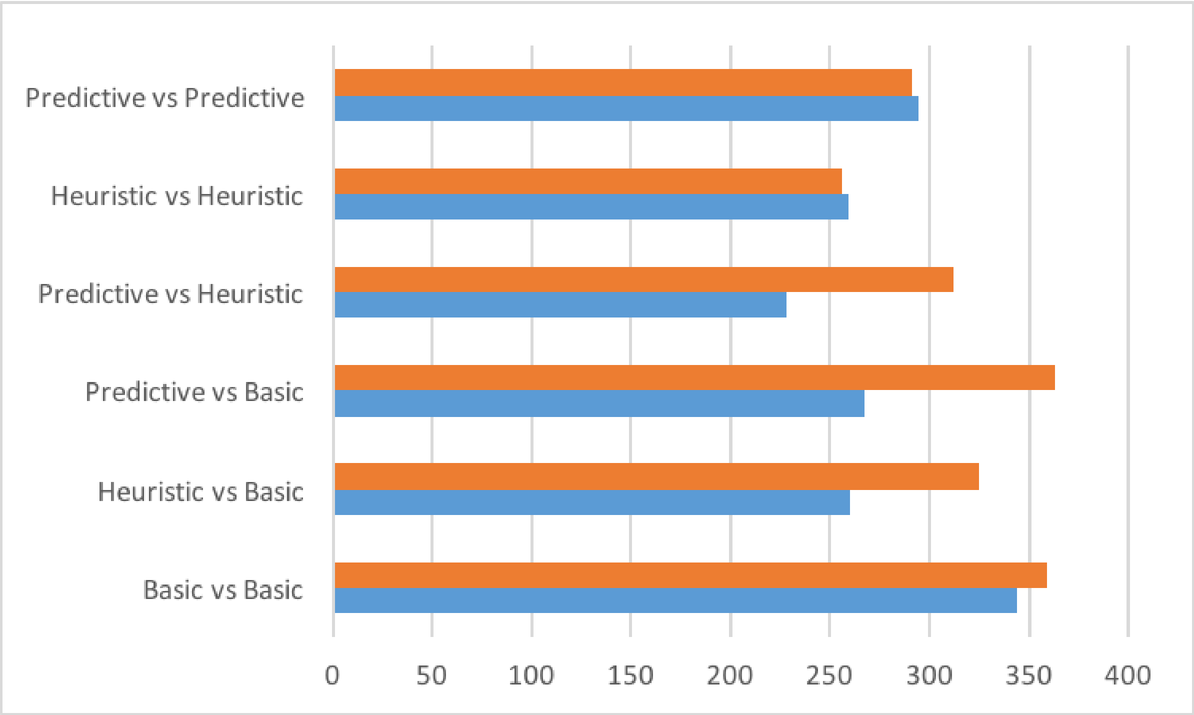
Heuristic



Predictive



Head to head **scores** for all models over six hour duration games.



The basic model: 25 rules
The predictive model: 26 rules.
The heuristic model: 45 rules to implement a simple shot selection strategy.

The results show that, within the limitations of the experiment, a predictive model – with an ability to use simulation to test its own actions to determine and evaluate their possible outcome – held a clear advantage over a model that used heuristics to test relationships between objects in a simulated scenario.

7. FUTURE WORK

Ongoing research is to use PACT-R in physical robotics. PACT-R is intended for robotics and embodied AI. Taking this system into the real world presents the considerable challenge of perceiving and simulating at least a small part of the real world.

Also to use PACT-R in combination with machine learning.

THANKS!

davidp12@utas.edu.au