# Dynamic Game Difficulty Scaling Using Adaptive Behavior-Based AI

Chin Hiong Tan, Kay Chen Tan, and Arthur Tay

## Presented by Nick Brusso

11/28/2012

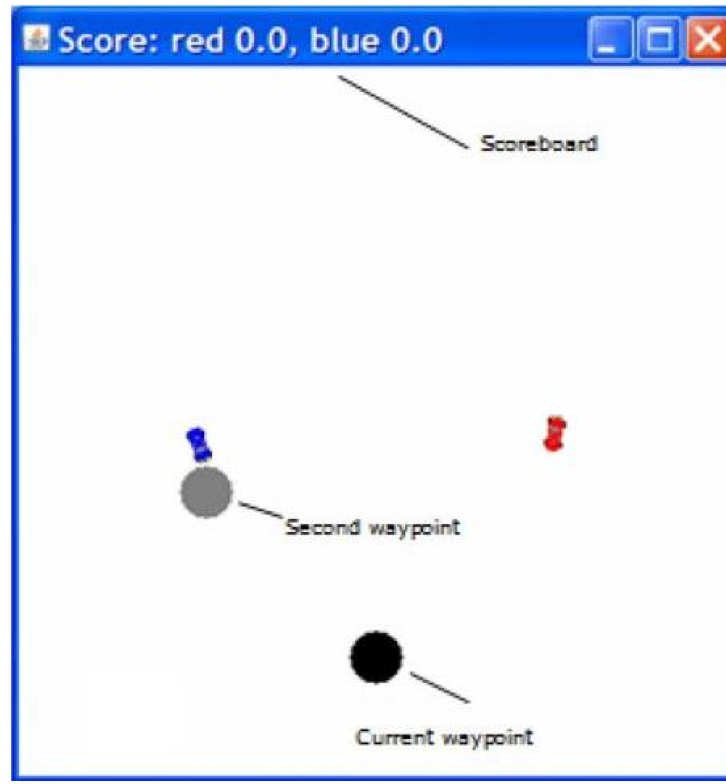
In IEEE Transactions on Computational Intelligence and AI in Games, Volume 3 Issue 4 (2011), pp. 289-301

# Objective

- Create an entertaining game AI
  - Satisfying to play against for a wide audience
  - High replay value
- Adapt game AI as the player plays
  - Dynamically scale difficulty in real-time
  - Two implementations are presented; I focus on the simpler one.

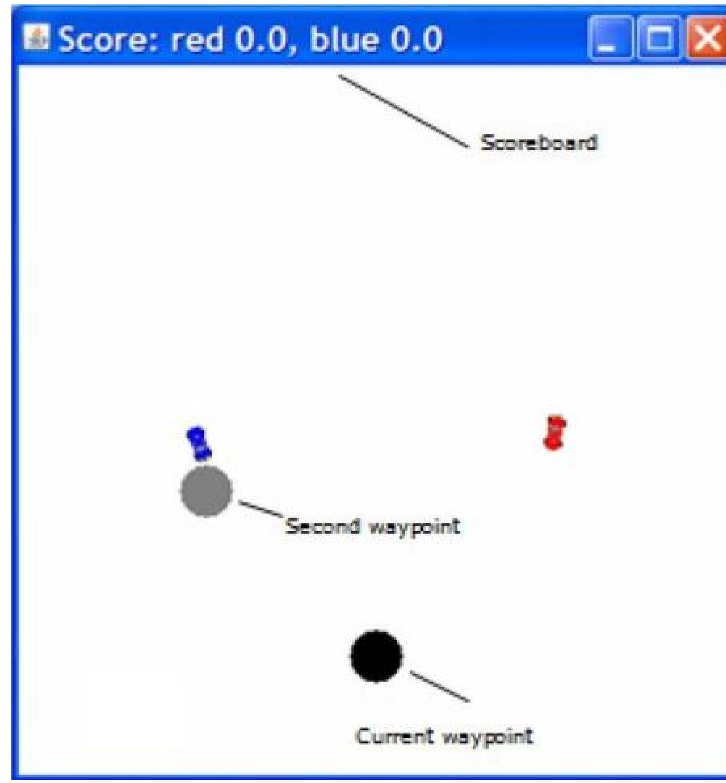
# Game Environment

- Simulated car racing
  - Current is worth 1 point
  - Second is worth 0 points
  - After Current is passed, Second → Current, NewWaypoint → Second
- Objective: gain the most points in a set time.
- Cars can move outside window boundaries
  - Advantageous for the AI



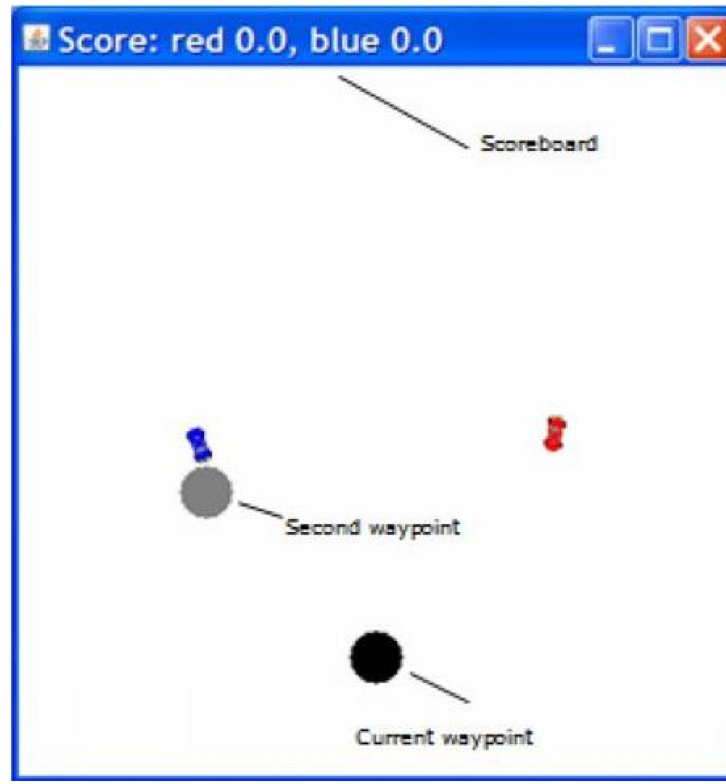
# Game Environment

- Control actions (on/off):
  - Accelerate, Decelerate, Left Turn, Right Turn, Neutral
  - A player would use the arrow keys
- Car physics:
  - Collisions between cars
  - Side skidding



# AI Behavior Components

- Driving Behavior
  - Speed Regulator
  - Reversing
  - Direction Switching Compensation
  - Tight Angle Turning
- Tactical Behavior
  - Waypoint Prediction
  - Time Wasting
  - Blocking



# Adaptive Controllers

- Satisfying gameplay experience
  - Over *n* games, |*Wins - Losses*| and *Draws* minimized
  - |*p1Score - p2Score*| minimized and *max*(*p1Score, p2Score*) maximized
- Artificial Stupidity
  - Force the AI to make deliberate mistakes
    - Selectively activate/deactivate behavior components.
    - Requires that the AI is overdesigned (small window for the player in this case)

# Adaptive Uni-Chromosome Controller (AUC)

- Stores one chromosome which encodes seven real numbers (probabilities of activating each behavior)
  - Expected behavior set encoded by the chromosome represents a "winning" strategy
- Chromosome is initialized to random values when the game begins.
- Chromosome is updated whenever a waypoint is passed, and a new behavior set is selected using probabilities.
  - If we lost the previous waypoint, probabilities are used as-is
  - If we won, probabilities are complemented before selection

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| 0.8 | 0.6 | 0.1 | 0.3 | 0.9 | 0.5 | 0.2 |

# Adaptive Uni-Chromosome Controller (AUC)

- AUC Update Algorithm
  - *win_i* : probability that behavior *i* is activated in the next phase.
  - *myDist*, *otherDist* : distances from each car to the waypoint.
  - *sgn(behavior_i)* : 1 if activated, -1 if not activated
  - *l* and *m*: learning and mutation rates (*l* = 0.1, *m* is unused)

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| 0.8 | 0.6 | 0.1 | 0.3 | 0.9 | 0.5 | 0.2 |

*1) If AUC win*
*for each behavior component* $(i = 1 \text{ to } 7)$
*if* $(rand() < myDist/(myDist + otherDist))$
$win_i = (win_i + sgn(behavior_i) \times l) \times m;$

*2) If AUC lose*
*for each behavior component* $(i = 1 \text{ to } 7)$
*if* $(rand() < otherDist/(myDist + otherDist))$
$win_i = (win_i - sgn(behavior_i) \times l) \times m;$

# Testing

The full controller (all behaviors enabled) and AUC were both tested against five static controllers:

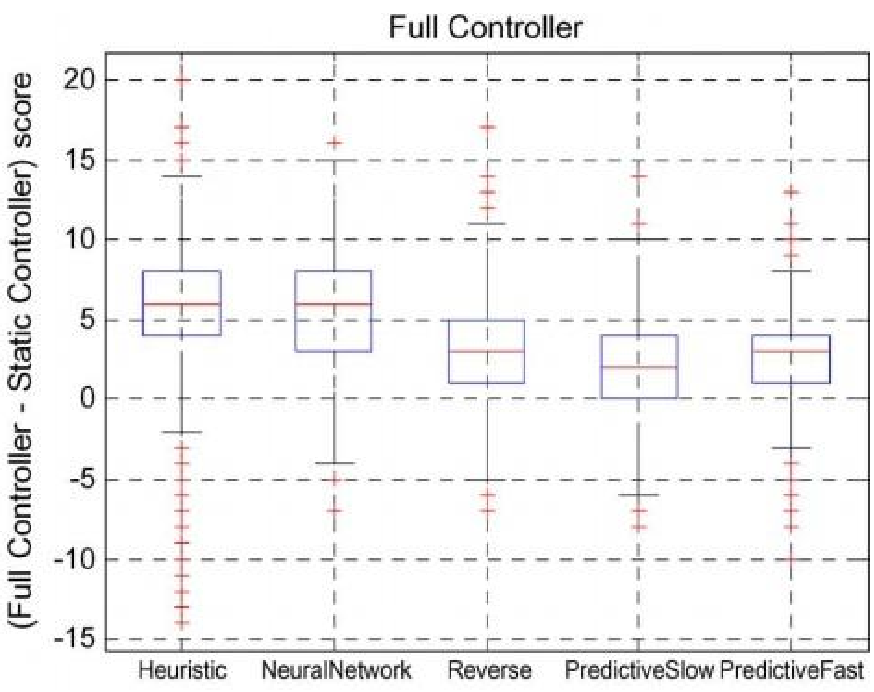
- Heuristic Controller (HC)
  - Uses simple rules to collect as many waypoints as possible; ignores opponent
- Neural Network Controller (NNC)
  - 9 Inputs: own orientation, opponent orientation, own speed, angle to *current*, angle to *second*, distance to *current*, distance to *second*, angle to opponent, distance to opponent
  - 2 Outputs: steering control, driving control
- Reverse Enabled Controller (RC)
  - Behavior controller with only reversing and direction switching behaviors active (subset of full controller)
  - Constant speed used instead of speed regulator behavior

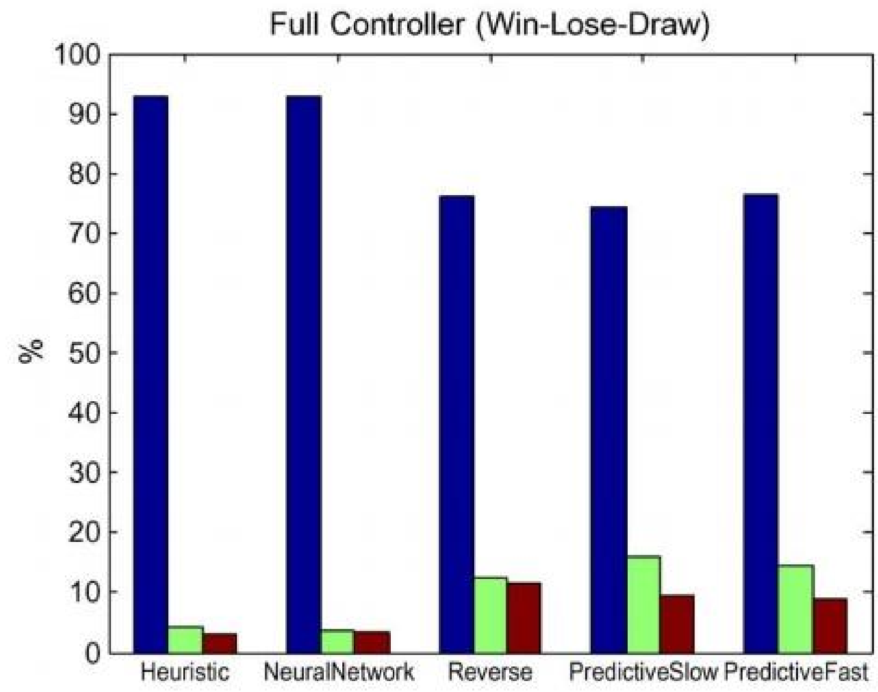
# Testing

The full controller (all behaviors enabled) and AUC were both tested against five static controllers:

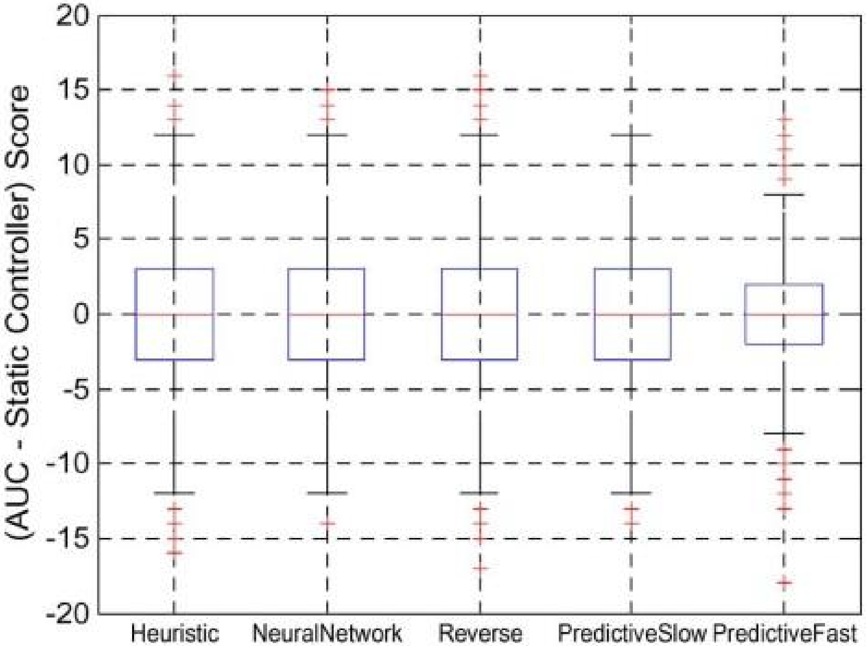
- Predictive Slow Controller (PSC)
  - Same as the Heuristic Controller with the Waypoint Prediction behavior activated
  - Slow constant speed used (5px per time step); This prevents skidding and overshooting the waypoint
- Predictive Fast Controller (PFC)
  - Same as PSC, with a speed of 8px per time step.
  - Reaches the waypoint faster, but might overshoot

# Results (Full Controller)

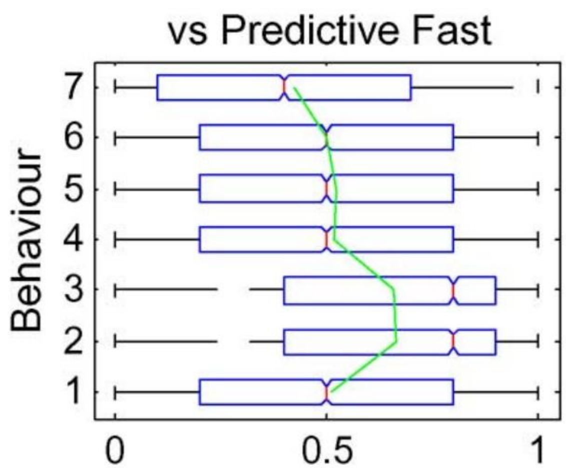


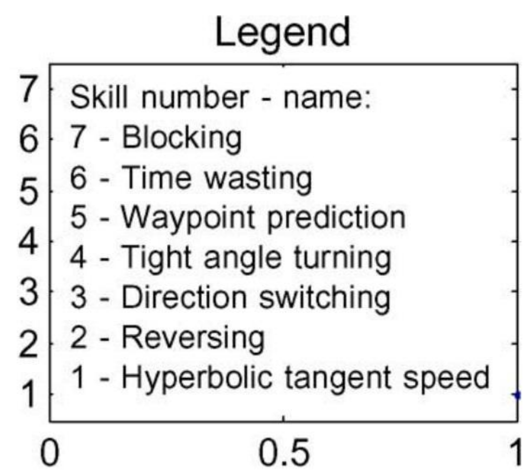
# Results (AUC)



20
15
10
5
0
-5
-10
-15
-20
(AUC - Static Controller) Score
Heuristic
NeuralNetwork
Reverse
PredictiveSlow
PredictiveFast

vs Predictive Fast
Behaviour
7
6
5
4
3
2
1
0
0.5
1

Legend
Skill number - name:
7 - Blocking
6 - Time wasting
5 - Waypoint prediction
4 - Tight angle turning
3 - Direction switching
2 - Reversing
1 - Hyperbolic tangent speed
0
0.5
1

# Conclusions

- AUC performed well in creating an entertaining experience.
  - Achieved a score difference of <= 4 for at least 70.22% of games played.
  - Wins/losses were well-distributed
- Deals well with a variety of opponents