matthew **bedder**

Ph.D. student in the Intelligent Games and Games Intelligence CDT

about

Matthew Bedder
YCCSA (RCH/234)
Ron Cooke Hub
University of York
YO10 5GE

mb708@york.ac.uk
www.bedder.co.uk

programming

C, C++
Matlab
Octave
Java
Python

technologies

Git
SVN
Unity
LaTeX
AVS Express

interests

Artificial intelligence for games; novel artificial intelligence applications; computer vision

education

since 2014 **Ph.D. in Computer Science** — University of York, IGGI CDT
Hierarchical Monte Carlo Tree Search
Looking into the automated usage of abstractions to guide MCTS
Modules taken in games design, games AI, and games analytics

2009 - 2014 **MEng Computer Science with Artificial Intelligence** — University of York
Project titled *Plan-Based Monte Carlo Tree Search*
Awarded first-class degree with honours

experience

10/13 - Now **University of York** — Casual research contract
Continuation of research on Parkinson's Disease
Work culminated in a paper in IET Systems Biology (to appear)

07/13 - 09/13 **YCCSA** — Summer School research position
Automated motion analysis for Parkinson's Disease
Collaboration with the departments of Biology and Electronics

07/11 - 07/12 **BAE Systems Advanced Technology Centre** — Industry Placement
Numerous large software projects in the area of computer vision
Nominated for Chairman's Award due to research performed

02/10 - 09/14 **University of York** — Student Ambassador
Running outreach and admissions events

events

2015 **Game Republic Student Showcase** — gamerepublic.net
Presented the Unity game *Vikings* to local industry figures

2015 **Pint of Science** — pintofscience.co.uk
Organised the *Tech Me Out* stream of talks for Pint of Science in York

2013 **C2D2 Poster Presentation** — york.ac.uk/c2d2
Presented a poster of my research at the Centre for Chronic Diseases and Disorders (C2D2) 2013 conference

references

Dr Daniel Kudenko (Academic supervisor)
daniel.kudenko@york.ac.uk
Department of Computer Science
University of York YO10 5HG

Dr Stephen Smith (YCCSA research project lead)
stephen.smith@york.ac.uk
Department of Electronics
University of York YO10 5DD

research

Monte Carlo Tree Search (MCTS) has been an area of much interest for AI researchers since its discovery in 2006. With it performing tremendously over certain games, MCTS has been extensively studied, although to date much of this research has been focussed on card and board games, with the board game *Go* perhaps seeing the most focus. Only recently have there been attempts in using Monte Carlo Tree Search in commercial video games, with the video games industry seeming less enamoured in the technique than academia.

Why is this the case? Well, although the core concept behind MCTS may be simple, the intricacies involved in optimising the technique to play games well can be very difficult and confusing. A multitude of different methods have been attempted to make MCTS agents play more intelligently, make decisions faster, and use fewer resources, but there is little consensus on which modifications are best over which domains, and even if the modifications are applicable beyond the specific games used in the specific research. Although researchers may be delighted to find out that MCTS outperforms all other agents for certain board games, how is this meant to be relevant to video games which often contain tougher challenges regarding the complexity of the interactions, and the time allowed be agents for selecting actions?

In the research I am undertaking for my Ph.D. I am looking into methods of using game abstractions to guide MCTS searching. In this I hope that I will be able to reduce the amount of effort required for AI programmers to implement MCTS into existing games (as the generation of useful abstractions can be somewhat simpler than the generation of useful heuristics), and I hope to reduce the amount of computation time required by MCTS agents to perform intelligent actions.

Early results from my research over a simple turn-based Capture-the-Flag game suggests that my approach could result in my modified MCTS agents outperforming existing agents whilst taking less time to make decisions, and I hope that I will soon be able to confirm these results over more complex games. I am also planning to look into automated or semi-automated abstraction generation, as well as investigating the impact of using “good” or “bad” abstractions over the performance of the agent.