

# Developing Smarter Artificial Intelligence for Companion Characters in Games

PHD Student: John Drake    Advisor: Professor Maxim Likhachev

Student Research Assistants: Rikky Roy Koganti, Pasan Pow Julsaksrisakul

## Introduction

This research project is focused on investigating how to make virtual companions behave more intelligently around human players in a game. Many games have such virtual companions, and they are usually built on developer-tailored state machines or behavior trees. However, they don't provide the opportunity for the player to teach the companion character smarter behaviors.

We are currently building on "Skyrim," an adventure video game which has non-player game characters that can be recruited as companions to help you fight. These companions are kind of dumb by default and do not understand any kind of complex tactics. We are looking into how to apply search-based planning algorithms to this problem.



## Objectives

- Create a system which allows a player to **teach** its companions complex game strategies including what to do in the world and how to do it
- Applying the **Experience-Graphs algorithm** to an AI planner
- Designing and implementing a **model of the game world** in Skyrim for the planner to use

## Technical Set-up

- Core algorithmic code is developed in C++ , with in-game Papyrus code to interface with the game engine.
- Uses the Skyrim Script Extender (SKSE) utility to inject our new functionality into the game world
- Using the Skyrim Creation Kit game editing tool to modify 3D game world for testing purposes



## Process

Pasan and Roy worked on implementing a model of the game world for the planner to use. This involved designing bear, wolf and player character models that simulate the behavior of their counterparts in the Skyrim game. John worked on the overall framework of the project and setting up a visualization system to test the game model code to see how the behavior worked. He set up a test map in Skyrim using Creation Kit.



After that, Pasan worked on getting the navmesh data from the test map, while Roy implemented Papyrus code in order to get the data of all NPCs and the player character in the map. This is to help visualize how the enemy characters navigate the world in our world model. Pasan implemented an **A-star search algorithm** on the navmesh with the data that Roy gathered from the test map.



This data is also sent to the AI planner for its use. John's main focus at this point was to implement the planning algorithm itself, which makes the choices about how to explore the **potential futures** and what inputs the world model should have.

## Progress so far

- Working models of player character, bear and the wolf
- Implemented visualization code allowing us to see how the models are behaving
- Built up framework to connect game world to C++ research code
- Tested what happens when reusing demonstration data directly without any planning. Helped to find particular problems and weaknesses for planning algorithm to address. *Planner code currently being written*