Hej, jag är här för att presentera mitt arbete av en flexibel schack-AI-agent, baserad på Case-based reasoning.

Jag heter Johannes Qvarford och går tredje året på programmet datorspelsutveckling – programmering. Och det är jag som tänkt utföra det här arbetet.

Mitt arbete består av ett antal delar som jag kommer presentera i ordning.

Först involverar det Case-based Reasoning, en teknik för att utveckla Artificiell Intelligens det vill säga AI.

Med hjälp av den här tekniken ska en AI-agent skapas med varierbar skicklighetsnivå. Och så ska AI-agenten spela schack.

Jag kommer förklara syftet djupare framöver, och sedan berätta vilken metod jag ska använda.

Efter det kommer jag svara på de frågor ni har.

Okey, Case-based reasoning är en teknik inom AI för att lösa problem baserat på tidigare, liknande problem. Till exempel, om den lila gubben i bilden till vänster möter tre röda kvadrater, så kanske den bör göra som den gjorde när den mötte tre röda trianglar. I det här fallet var det att gå bakåt. Ett minne, det vill säga en situation och vad som gjordes i situationen kallas för ett fall. Och alla fall som AI-agenten kan referera till kallas tillsammans för en fallbas.

Vanligtvis lär man upp AI-agenter genom att låta en expert utföra uppgiften som AI-agenten ska göra. AI-agenten använder sedan de insamlade data som fallbas när den försöker utföra uppgiften själv.

Traditionellt sätt har en expert används per beteende i tidigare forskningsarbeten.

Jag tänker dock testa om det går att skapa flera vettiga beteenden från ett färre antal experter genom att blanda deras beteenden. I bilden visas hur expertdata från tre personer används som beteenden för fem skicklighetsnivåer för AI-agenten.

Om det visar sig fungera bra, så kan det innebära att det kan krävas mycket mindre resurser för spelutvecklare att utveckla AI-agenter med varierbar skicklighetsnivå. I exemplet så krävs bara tre experter istället för fem, vilket är en stor minskning, och det är den stora anledningen till varför jag gör arbetet.

Okey, så varför har jag valt att skapa en schack AI-agent? Jag har valt schack mest som ett exempel för att det har ett antal trevliga egenskaper. Tanken är att det även ska gå att implementera för andra spel med varierbar svårighetsgrad.

Det trevliga med schack jämfört med många datorspel är att det finns relativt få lägen att göra handlingar i, och lägena kräver inte mycket plats att lagra. Visserligen grenar antal möjliga lägen ut sig för varje drag, men schackmatcher tar ofta inte längre än 100 drag. Jämför detta med ett realtidsstrategispel som körs i 30 eller 60 frames per sekund, som måste representera en hel värld per läge.

Utöver det har schack ett väldigt stort ekosystem. Det finns mycket forskning om schack, standardiserade kommunikationsprotokoll mellan schackmotorer, och väldigt många dokumenterade matcher att använda som expertdata.

I arbetet ska jag skapa en schackspelande AI-agent, alltså en så kallad schackmotor.

Jag ska använda spelade matcher från den universella schackfederationen databas, och skapa fallbaser för olika svårighetsgrader baserade på matcher från en spelare vars rankning matchar svårighetsgraden. Så för högre svårighetsgrader ska matcher från högre rankade spelare användas. Om det inte finns en spelare som svårighetsgraden matchar, ska fallbasen vara en blandning av den högre och lägre svårighetsgradens fallbas.

Till sist ska AI-agenten spela mot sig själv på olika svårighetsgrader, så först ska AI-agenten spela svårighetsgrad 1 mot 2, 3, 4, och sedan 2 mot 1,3,4 och så vidare. Sedan graderas svårighetsgraderna baserat på hur bra de spelade. Om de stämmer överens, det vill säga att högre svårighetsgrader rankades högre, så visar det att AI-agenten betedde sig som förväntat på de svårighetsgraderna som var blandningar av experter. Och det skulle då visa att det är rimligt att skapa Case-based reasoning baserade AI-agenter med varierbar svårighetsgrad med färre experter än svårighetsgrader.

Då var min presentation klar. Då får jag tacka för mig, och inväntar era frågor.