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.

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

Först involverar det Case-based Reasoning, en teknik för att utveckla AI-agenter. Med hjälp av den här tekniken ska en AI-agent skapas med varierbar skicklighetsnivå. Och det AI-agenten ska göra, är att 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. Det är inte en teknik specifikt för spel-AI, men det är det som mitt arbete är inriktat på.

Ett exempel vore en AI-agent för en soldat i ett FPS spel, som ska utföra handlingar baserat på vad den ser och vad den har sett. I bilden till vänster syns den lila soldaten som mött tre orangea fyrkantiga enheter. Den undrar vad den ska göra, och beslutar sig att skjuta, vilket den gjorde när den stötte på tre orangea trianglar. Det är viktigt för AI-agenten att känna igen liknande situationer på ett bra sätt, trianglarna är kanske fiender medan kvadraterna är allierade, så det kanske inte är bra att skjuta kvadraterna.

Ett minne det vill säga en situation tillsammans med sin handling kallas för ett fall. Och gruppen fall som AI-agenten kommer ihåg kallas för en fallbas.

AI-agenten kan inte utföra sin uppgift utan några ursprungliga minnen. Så vanligtvis lär man upp AI-agenter genom att låta en mänsklig expert utföra uppgiften som AI-agenten ska göra. AI-agenten använder sedan de insamlade data som fallbas när den försöker utföra uppgiften själv.

Case-based reasoning har använts för att utveckla spel-AI i ett antal tidigare forskningsarbeten. T.ex har Aha, Molineaux och Ponson undersökt hur stora mängder fall kan hanteras för att spara minne och minska söktider. Och Rekaber, Shadfer och Osareh har undersökt hur AI-agenter kan lära sig samarbeta i lagsporter som fotboll.

I tidigare forskningsarbeten har traditionellt en expert används per fallbas och beteende, för att det kan vara svårt för en person att uttrycka olika beteenden.

Jag tänker undersöka om det går att skapa flera vettiga beteenden från ett färre antal experter genom att blanda deras beteenden. I mitt fall är varje beteende en skicklighetsnivå. I bilden visas hur expertdata från tre personer används som beteenden för fem skicklighetsnivåer för en AI-agent.

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 den stora anledningen till varför jag gör arbetet är att undersöka om det är möjligt att spara dessa resurser.

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

Det trevliga med schack jämfört med många andra spel är att det finns relativt få lägen att göra handlingar i, och lägena kräver inte mycket plats att lagra på en dator. 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 frames per sekund och grenar ut sig för varje frame, som måste representera en hel värld per läge. För att använda case-based reasoning för dessa spel måste antal fall och storleken på fallen ofta minskas, vilket inte behöver göras för schack, åtminstone inte lika mycket.

Utöver det har schack ett väldigt stort ekosystem. Det finns mycket forskning om schack, standardiserade kommunikationsprotokoll mellan schackmotorer, och väldigt många dokumenterade matcher. Därför behöver inte inga experter spela spelet under övervakning, eftersom det redan finns dokumenterade matcher som expertdata.

I arbetet ska jag skapa en schackspelande AI-agent, alltså en så kallad schackmotor. Schackmotorn ska kunna spela på olika skicklighetsnivåer, där varje nivå överensstämmer med en fallbas.

Jag ska använda spelade matcher från en databas, och skapa fallbaser för olika skicklighetsnivåer baserade på matcher från spelare vars rankning matchar skickilighetsnivån. Så för högre skicklighetsnivåer ska matcher från högre rankade spelare användas. Vissa svårighetsgrader ska även använda matcher från flera olika spelare med både högre och lägre rankning.

AI-agenten ska spela mot sig själv på olika skicklighetsnivåer, så först ska AI-agenten spela skicklighetsnivå 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 alla svårighetsgraderna, inklusive dem som var blandningar av experter. Det skulle alltså inte krävts separata experter för de skicklighetsnivåerna, för att blandningarna var bra nog.

Detta skulle då visa att det är rimligt att skapa case-based reasoning baserade AI-agenter med flera skicklighetsnivåer baserat på färre experter.

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