

Lab 1A Rapport

AI MINER WITH FRIENDS.

Benjamin Vesterlund | SOOO6D | 05/02-19

Problemspecifikation

Given uppgift var att skapa en karaktär vars inre tillstånd representeras med en FSM, denna karaktärs miljö ska visuellt visas antingen grafiskt eller i textform. En "realistisk" tid ska även implementeras.

Karaktären ska ha följande liv.

- Samlapengar (två jobb)
- Köpa grejer (mat, spade)
- Sova
- Äta
- Dricka
- Umgås med kompisar

I uppgiften ingår även en balans mellan dessa saker.'

Meddelanden mellan karaktärer ska även implementeras och vissas i form av SMS. där dom ska bestämma möten och avboka möten om det skulle behövas (om karaktären blir för hungrig eller törstig)

Minst 4st karaktärer ska finnas i världen.

Användarhandledning

Projektet finns på <https://git.gscept.com/benves-7/door12d.git>

I Git projektet så finns det två kurser, både GameEngine och AI, under AI mappen finns två mappar, mappen "Lab 1A v2" är aktuell mapp, eftersom den första inte gick att köra på extern dator (hemma).

Startup är "Lab_1A.py", då skapas en lista på miners med 4st entitets. Vill man lägga till fler gör man det där.

För att lägga tills positioner så läggs det till i "Positions.py" i form av en nyckel i ett bibliotek med en tuple som värde.

Systembeskrivning

Har en bas klass BaseGameEntity som ärvs av klassen Miner.

Miner innehåller variabler för alla behov miners har så som hunger och törst. Även funktioner för att byta State (**ChangeState**), kolla alla behov (**CheckNeeds**), och för att förflytta karaktären mot en plats (**GoTowards**).

En bas klass State innehåller tomma virtuella funktioner som ärvs ner till alla States en klass kan ha. Innehåller även hanteringen av meddelanden som skickas mellan alla miners.

Alla State klasser kollar om karaktären är framme på en viss koordinat med miners funktion GoTowards och sedan uträttar den dem fördelar och negativa saker dem ges av att vara på platsen.

Lösningens begränsningar

Begränsningar är många, meddelande hanteringen är bristande i den form programmet har just nu och gör att karaktärer kan hamna i en loop av att springa mot ett möte och avbryta ett möte på grund av fel mötesplats. Lösning är att alltid mötas på "Diner" och att hunger och törst löser sig av att vara där med någon.

Diskussion

Största problemet jag har stött på är språket python använder sig av, annars har det varit svårast att få meddelanden att fungera mellan karaktärerna, eftersom jag har gjort meddelande systemet lite dumt så kan dem hamna i en loop av att meddela varandra att dem vill träffas.

Genom att ändra miners variabel `m_Location` i ett States Enter funktion gjorde det enkelt att få karaktären att alltid gå mot platsen man vill.

Laborationen har varit bra då den har gett en bra bild av värdet på att planera först och hur enkelt en del saker blir med FSM!

Grafiska lösningen är hittad på Google ("Grafics.py") struktur på messagedispatcher och entitymanagern är inspirerad av Filip Renman early draft.