

Part A

Trond Zachariassen

Januar 2024

Deklarasjon

Info om algoritmene er hentet fra AIMA læreboken sine definisjoner og terminologi.[1]

1 Spill - Don't Panic Episode 1

1.1 Problemet

Målet i Don't panic episode 1 på codeingame er å guide klonene til trygghet ved hjelp av enkle kommandoer og portaler slik at klonene finner utgangen. Klonene beveger seg i retningen de er rettet mot og kan ved hjelp av kommandoen **BLOCK** endre retning. Når klonen har byttet retning og ser enten mot en portal eller utgangen sendes kommandoen **WAIT**.

1.2 PEAS

- Ytelsesmål: Vellykket redning av minst én klon.
- Miljø: Flere rektangulære plan, portaler, kloner og én utgang. Miljøet er dynamisk med bevegende kloner og portaler som endrer posisjon fra plan til plan.
- Aktuatorer: Handlinger tatt for å enten blokkere eller vente, som påvirker retningen klonene går.
- Sensorer: Gir informasjon om gjeldende tilstand, plassering av portal og posisjonen og retningen til den ledende klonen.

1.3 Kategorisering av miljø

- Fully Observable: Alt av informasjon om miljøet er synlig, så miljøet er fully observable.
- Single-agent: Én agent som styrer valget til fremste klon ut i fra informasjonen hentet fra miljøet.
- Cooperative: Samarbeid om et felles mål om å få klonene til trygghet.
- Deterministic: De samme handlingene fører til det samme resultatet. Endrer ikke på seg underveis.
- Episodic: Hver runde er uavhengig av tidligere og fremtidige runder.
- Dynamic: Kloner og portaler som flytter på seg og har forskjellige posisjoner på hvert plan.

- Discrete: Problemet er diskret med begrensede plan og posisjoner.
- Known: Hele miljøet er kjent for agenten.

2 Problemformuleringer

- Tilstandene representerer klonens retning og tilstand, samt portalens posisjon.
- Den opprinnelige tilstanden bestemmes av inngangsparametrene: antall etasjer og klonens posisjon.
- Det er to handlinger som kan utføres: blokker eller vent.

3 Løsningsmetodikk

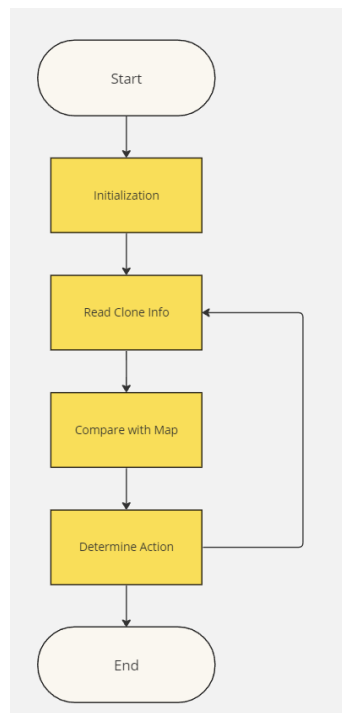
3.1 Algoritmebeskrivelse

- Algoritme type: Regelbasert (reaktiv/refleksiv)
- Naturinspirert: Enkel beslutningstaking basert på forhåndsdefinerte forhold.

3.2 Agent typer

- Målbasert agent: Agenten har som mål å oppnå målet gitt av spillet som er å redde klonene.
- Lærings agent: Agenten lærer av omgivelsene og tilpasser seg basert på tilstand og utfall.
- Problemløsende agent: Agenten løser problemet ved å bestemme når den skal blokkere eller vente effektivt.

4 Flytskjema



Figur 1: Detaljert flytskjema som viser alle beslutningspunkter og mulige veier gjennom programmet. Innenfor Initialization leses inputs og sjekkes kartet, fulgt av en løkke som håndterer klonenes bevegelser basert på deres posisjon og retning.

5 Kode:

```
import sys, math

nbFloors, width, nbRounds, exitFloor, exitPos, nbTotalClones,
    ↪ nbAdditionalElevators, nbElevators = [int(i) for i in input().split()]

map = [exitPos] * nbFloors

for i in range(nbElevators):
    elevatorFloor, elevatorPos = [int(i) for i in input().split()]
    map[elevatorFloor] = elevatorPos

while True:
    cloneFloor, clonePos, direction = input().split()
    cloneFloor = int(cloneFloor)
    clonePos = int(clonePos)

    if clonePos < map[cloneFloor] and direction == "LEFT" or \
        clonePos > map[cloneFloor] and direction == "RIGHT":
        print ("BLOCK")
    else:
        print ("WAIT")
```

Referanser

[1] Stuart J. Russell and Peter Norvig. Artificial Intelligence A Modern Approach Fourth Edidtion. PEARSON, 2021.