

Proiect ALJV

Roșu Mihai-Cosmin

1. Descriere joc:

Jocul implementat este de tip 2D top-down în care jucătorul trebuie să învingă un singur inamic. Pe hartă, care este sub forma unui labirint, se pot găsi 3 tipuri de colectibile: heal, power-up de viteză și arme.



Fiecare heal oferă 25 HP, fiecare power-up de viteză dublează viteza pentru 3 secunde, iar fiecare armă are 20 damage și 5 durabilitate. Atât jucătorul, cât și inamicul încep cu 100 HP.

2. Inteligență artificială

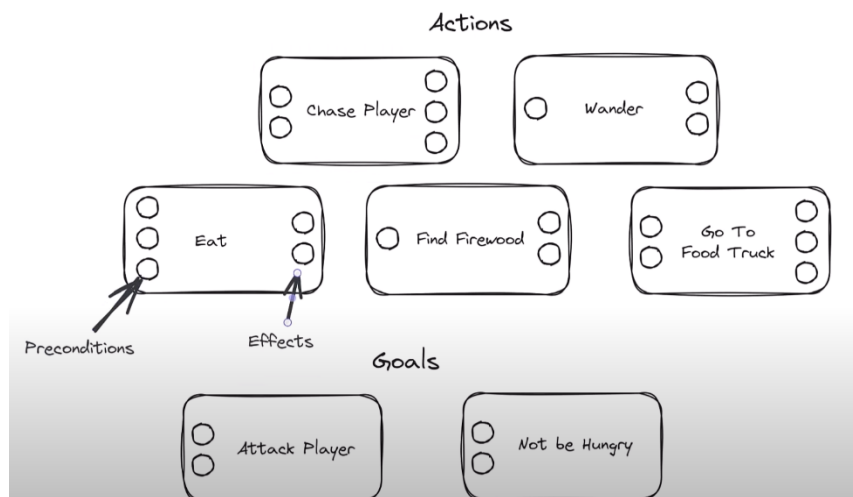
Inamicul din joc folosește Goal Oriented Action Planning (GOAP) pentru a determina acțiunile pe care le face.

GOAP este un sistem folosit în jocuri video pentru luarea deciziilor în ceea ce privește comportamentul caracterelor. El este format din 3 părți:

- Beliefs (Cunoștințe): Reprezintă informația pe care o are agentul;
- Actions (Acțiuni): Reprezintă acțiunile agentului;
- Goals (Scopuri): Reprezintă scopul la care vrea să ajungă agentul.

Fiecare scop are un număr de precondiții necesare pentru a fi îndeplinit. De asemenea, fiecare acțiune are un număr de precondiții necesare pentru a putea avea loc, dar și un număr de efecte care devin adevărate după ce acțiunea are loc.

GOAP folosește cunoștințele agentului și scopul la care vrea să ajungă pentru a construi un șir de acțiuni care duc la îndeplinirea scopului ales.



3. Implementare și rezultate

Pentru implementare a fost folosit următorul videoclip: [Better AI in Unity - GOAP \(Goal Oriented Action Planning\)](#) .

Agentul are definite multiple cunoștințe, de la informații legate de nivelul său de HP, locațiile power-up-urilor, informații despre jucător și informații sau presupuneri legate de un eventual conflict cu jucătorul.

```
factory.AddBelief("Nothing", () => false);

factory.AddBelief("AgentIdle", () => !pathfinder.hasPath);
factory.AddBelief("AgentMoving", () => pathfinder.hasPath);
factory.AddBelief("AgentHealthLow", () => enemyController.healthBar.currentHealth <= 20);
factory.AddBelief("AgentIsHealthy", () => enemyController.healthBar.currentHealth > 20);
factory.AddBelief("AgentIsFullHealth", () => enemyController.healthBar.currentHealth == enemyController.healthBar.maxHealth);
factory.AddBelief("AgentKnowsHeals", () => heals.Count > 0);
factory.AddBelief("AgentKnowsSpeeds", () => speedUps.Count > 0);
factory.AddBelief("AgentKnowsSwords", () => swords.Count > 0);
factory.AddBelief("AgentFastSpeed", () => enemyController.characterController.speedUp > 1f);
factory.AddBelief("AgentHasNoWeapon", () => !GameManager.GetInstance().enemyHasSword);
factory.AddBelief("AgentHasWeapon", () => GameManager.GetInstance().enemyHasSword);
factory.AddBelief("AgentCanKillPlayer", () => GameManager.GetInstance().enemyHasSword && (enemyController.sword.durability + enemyController.sword.damage) >= playerHP);
factory.AddBelief("AgentCannotKillPlayer", () => GameManager.GetInstance().enemyHasSword && (enemyController.sword.durability + enemyController.sword.damage) < playerHP);
factory.AddBelief("AgentCanKillPlayerOneHit", () => GameManager.GetInstance().enemyHasSword && enemyController.sword.damage >= playerHP);

factory.AddPlayerSensorBelief("PlayerInChaseRange", chaseSensor);
factory.AddPlayerSensorBelief("PlayerInAttackRange", attackSensor);

factory.AddBelief("AttackingPlayer", () => false);
```

De asemenea, agentul are definite multiple acțiuni care să îl conducă spre scopurile pe care le poate lua. Aceste acțiuni includ mișcarea către un anumit loc, urmărirea jucătorului și atacarea jucătorului.

Pentru partea de scopuri, agentul are definite 6 scopuri: Căutare, Colectare Power-up de viteză, Obținere HP complet, Obținere a unei arme noi, Vindecare și Atacare jucător.

Pentru verificarea rezultatelor, se pot vizualiza scopurile alese, și șirul de acțiuni pe care îl ia pentru a îndeplini scopul ales.

```
[22:15:46] Calculating any potential new plan
UnityEngine.Debug:Log (object)
[22:15:46] Goal: Wander with 1 actions in plan
UnityEngine.Debug:Log (object)
[22:15:46] Popped action: Wander Around
UnityEngine.Debug:Log (object)
[22:15:49] Wander Around complete
UnityEngine.Debug:Log (object)
[22:15:49] Plan complete
UnityEngine.Debug:Log (object)
```

```
[22:15:49] Calculating any potential new plan
UnityEngine.Debug:Log (object)
[22:15:49] Goal: GetNewWeapon with 1 actions in plan
UnityEngine.Debug:Log (object)
[22:15:49] Popped action: MoveToWeapon
UnityEngine.Debug:Log (object)
[22:15:51] MoveToWeapon complete
UnityEngine.Debug:Log (object)
[22:15:51] Plan complete
UnityEngine.Debug:Log (object)
```

```
[22:16:14] Calculating any potential new plan
UnityEngine.Debug:Log (object)
[22:16:14] Goal: HealUp with 1 actions in plan
UnityEngine.Debug:Log (object)
[22:16:14] Popped action: MoveToHeal
UnityEngine.Debug:Log (object)
[22:16:15] MoveToHeal complete
UnityEngine.Debug:Log (object)
[22:16:15] Plan complete
UnityEngine.Debug:Log (object)
```

```
[22:16:52] Calculating any potential new plan
UnityEngine.Debug:Log (object)
[22:16:52] Goal: AttackPlayer with 3 actions in plan
UnityEngine.Debug:Log (object)
[22:16:52] Popped action: MoveToHeal
UnityEngine.Debug:Log (object)
[22:16:54] MoveToHeal complete
UnityEngine.Debug:Log (object)
```

! [22:16:08] Calculating any potential new plan
UnityEngine.Debug:Log (object)

! [22:16:08] Goal: AttackPlayer with 2 actions in plan
UnityEngine.Debug:Log (object)

! [22:16:08] Popped action: ChasePlayerWhenHealthy
UnityEngine.Debug:Log (object)

! [22:16:09] Target changed, clearing current action and goal
UnityEngine.Debug:Log (object)

! [22:16:09] Calculating any potential new plan
UnityEngine.Debug:Log (object)

! [22:16:09] Goal: AttackPlayer with 1 actions in plan
UnityEngine.Debug:Log (object)

! [22:16:09] Popped action: AttackPlayer
UnityEngine.Debug:Log (object)

! [22:16:10] AttackPlayer complete
UnityEngine.Debug:Log (object)

! [22:16:27] Calculating any potential new plan
UnityEngine.Debug:Log (object)

! [22:16:27] Goal: AttackPlayer with 4 actions in plan
UnityEngine.Debug:Log (object)

! [22:16:27] Popped action: DiscardWeapon
UnityEngine.Debug:Log (object)

! [22:16:27] DiscardWeapon complete
UnityEngine.Debug:Log (object)

! [22:16:27] Calculating any potential new plan
UnityEngine.Debug:Log (object)

! [22:16:27] Current goal exists, checking goals with higher priority
UnityEngine.Debug:Log (object)

! [22:16:27] No plan found
UnityEngine.Debug:LogWarning (object)

! [22:16:27] Goal: AttackPlayer with 3 actions in plan
UnityEngine.Debug:Log (object)

! [22:16:27] Popped action: MoveToWeapon
UnityEngine.Debug:Log (object)

! [22:16:30] Target changed, clearing current action and goal
UnityEngine.Debug:Log (object)

! [22:16:30] Calculating any potential new plan
UnityEngine.Debug:Log (object)

! [22:16:30] Goal: AttackPlayer with 2 actions in plan
UnityEngine.Debug:Log (object)

! [22:16:30] Popped action: ChasePlayerWhenHealthy
UnityEngine.Debug:Log (object)