

GameAdviser

Annunziata Valentina, 0512108693

Capone Christian, 0512108534

Anno accademico 2022/2023

Contents

1	Introduzione	2
2	Specifica dell'Ambiente	2
3	Soluzione	2
4	Algoritmi adottati	3
4.1	Ricerca Lineare	3
4.1.1	Descrizione Algoritmo	3
4.1.2	Pseudocodice	3
4.1.3	Numero Confronti	3
4.1.4	Considerazioni	3
4.2	Ricerca a Costo Uniforme	4
4.2.1	Descrizione Algoritmo	4
4.2.2	Pseudocodice	4
4.2.3	Numero Confronti	4
4.2.4	Considerazioni	4
5	Implementazione	5
5.1	Ricerca Lineare	5
5.2	Ricerca a Costo Uniforme	5
5.3	Decrescente	6
5.4	FillDatabase	6
6	Risultati	7
6.1	Test tipologia = "Horror"	7
6.1.1	Valutazione minima = 3 , size = 1010	7
6.1.2	Valutazione minima = 4 , size = 1010	7
6.1.3	Valutazione minima = 2 , size = 1010	8
6.1.4	Valutazione minima = 3 , size = 510	8
6.1.5	Valutazione minima = 4 , size = 510	9
6.1.6	Valutazione minima = 2 , size = 510	9
6.2	Test tipologia = "Fantascienza"	10
6.2.1	Valutazione minima = 2 , size = 510	10
6.2.2	Valutazione minima = 3 , size = 510	10
6.2.3	Valutazione minima = 4 , size = 510	11
6.2.4	Valutazione minima = 4 , size = 1010	11
6.2.5	Valutazione minima = 3 , size = 1010	12
6.2.6	Valutazione minima = 2 , size = 1010	12
6.3	Test tipologia = "Open World"	13
6.3.1	Valutazione minima = 3 , size = 2010	13
7	Conclusioni	13

1 Introduzione

GameToday è una piattaforma online il cui nucleo è la recensione videoludica, tramite la quale novizi del settore o appassionati di videogiochi possono informarsi creando una community stabile e peculiare. All'interno di tale sistema è stato implementato GameAdviser. Questo tool che sfrutta l'intelligenza artificiale è in grado di consigliare ad un utente quali sono i videogiochi presenti in piattaforma che più sono conformi alle sue preferenze, quest'ultime saranno fornite in fase di registrazione dall'utente stesso mediante un campo della form.

2 Specifica dell'Ambiente

La specifica di un agente è costituita da:

TIPO DI AGENTE	Sistema di raccomandazione di videogiochi
MISURA DI PRESTAZIONE (PERFORMANCE)	L'accuratezza dei videogiochi consigliati, ovvero quanto la tipologia di un videogioco e le preferenze di un utente sono simili
AMBIENTE	Catalogo dei videogiochi, insieme degli utenti
ATTUATORI	Visualizzazione dei videogiochi più consoni ad un determinato utente
SENSORI	Pagina "Videogiochi consigliati" nell'area personale dell'utente

Nello specifico l'ambiente preso in esame è:

- Completamente osservabile: in quanto l'agente conosce tutte le caratteristiche dei videogiochi e degli utenti;
- Singolo Agente: vi è la presenza di un unico agente;
- Deterministico: lo stato successivo dell' ambiente è completamente determinato dallo stato corrente e dall'azione eseguita dall'agente (la compilazione della form);
- Statico: l'agente elabora sull'insieme di tipologie di videogioco già presenti, se viene aggiunta una nuova all'insieme, l'agente dovrà effettuare l'elaborazione da zero;
- Discreto: le percezioni e le azioni hanno valori finiti/discreti (es. limitate tipologie di videogioco).

3 Soluzione

Formulazione incrementale del problema:

Sia N l'insieme di tutti i videogiochi; N non è ordinato in base alle valutazioni degli utenti;

Sia V l'insieme dei videogiochi consigliati;

Sia v il videogioco considerato in un dato istante;

Sia $v(gm)$ il genere e la media delle valutazioni del videogioco v ;

- Stati: ogni $v(gm)$ è uno stato;
- Stato iniziale: V è vuoto;
- Azioni: aggiungere un v ad V se $v(gm)$ rispetta il genere dell'utente e la valutazione minima;
- Modello di transizione; restituisce V con $|V| + 1$ se viene trovato un v potenzialmente adatto alla soluzione;
- Test obiettivo: N è vuoto, oppure non ha più senso proseguire nella ricerca (ogni ulteriore iterazione dopo un certo punto della ricerca sarebbe inutile).

Lo scopo dell'intelligenza artificiale è di consigliare una lista di videogiochi per ogni utente in base alle loro preferenze. Tra le varie strategie, la più semplice che è stata considerata è la ricerca lineare, è stato analizzato inoltre l'algoritmo A*. Tuttavia quest'ultimo tiene conto dell'euristica $h(n)$ (costo dal nodo osservato a quello obiettivo), il quale non è attinente con le specifiche del nostro problema. Si è considerato il caso in cui $h(n)$ è uguale a 0, ovvero si sta trattando dell'algoritmo di ricerca a costo uniforme. Quindi il lavoro svolto si è concentrato su due algoritmi di ricerca non informata: Ricerca Lineare e Ricerca a Costo Uniforme.

4 Algoritmi adottati

4.1 Ricerca Lineare

La ricerca lineare (o 3ricerca sequenziale) è l'algoritmo di ricerca più semplice che cerca un elemento in un elenco in ordine sequenziale. Si basa sulla visita di un elenco, controllando dall'inizio alla fine le proprietà di tutti gli elementi trovati.

4.1.1 Descrizione Algoritmo

Si controlla inizialmente se il Catalogo non sia vuoto: se lo è, viene restituita Lista. L'algoritmo utilizza una variabile valutazioneMin che è inizializzata a 3, tale valore è stato scelto in quanto si vuole restituire una lista contenente solo i videogiochi con una certa media di valutazioni. Si confrontano sequenzialmente queste ultime con valutazioneMin e il genere (preferito dell'utente) se corrisponde con quello dei videogiochi presi in considerazione; se la verifica da esito positivo allora si aggiunge il videogioco alla lista delle soluzioni. La ricerca continua finché non sono stati esaminati tutti gli elementi di Catalogo.

4.1.2 Pseudocodice

1. function Ricerca Lineare(Catalogo, genere)
2. if Catalogo is empty then return Lista
3. valutazioneMin in \leftarrow 3
4. $i \leftarrow 0$
5. loop:
6. if $i \geq \text{Catalogo.size}$ then end loop
7. if $\text{Catalogo}[i] \geq \text{valutazioneMin}$ AND $\text{Catalogo}[i].\text{equals}(\text{genere})$
8. Lista \leftarrow add(Catalogo[i])
9. $i \leftarrow i + 1$
10. goto loop
11. close
12. return Lista

4.1.3 Numero Confronti

Il genere inserito dall'utente viene confrontato con il genere di tutti gli n videogiochi di Catalogo; il numero massimo di confronti è pari alla dimensione della lista di input.

4.1.4 Considerazioni

Il numero di confronti, poiché dipende dalla cardinalità del Catalogo, potrebbe essere eccessivamente elevato e quindi il tempo necessario per eseguire tutti i confronti potrebbe essere notevole. E' stata considerata una soluzione più efficiente.

4.2 Ricerca a Costo Uniforme

La ricerca a costo uniforme è una strategia di ricerca non informata in cui l'algoritmo espande il nodo sulla frontiera con il costo di cammino più basso dal nodo iniziale (nodo radice). Per questo è possibile memorizzare la frontiera come coda a priorità ordinata secondo g e in cima alla coda ci saranno i nodi di costo minore.

4.2.1 Descrizione Algoritmo

In primis si controlla se Catalogo è vuoto, in tal caso viene restituito Lista. Si controlla la valutazione di tutti gli elementi di Catalogo per decidere in che modo ordinare la lista. La media delle valutazioni ha un valore massimo di 5, considerando che i videogiochi che faranno parte della soluzione hanno una valutazione alta, si è stabilito tale criterio di ordinamento: se la media è maggiore o uguale di 3, la lista viene ordinata in ordine decrescente. Si inizia a scorrere tutto il Catalogo, verificando che il videogioco rispetti il genere e il minimo della valutazione considerata affinché sia un buon candidato per l'insieme delle soluzioni, e quindi si aggiunge a Lista. Successivamente si verifica se è il caso di proseguire nella ricerca: poiché la lista è ordinata, è necessario fissare un "limite" oltre il quale l'esplorazione della lista risulterebbe inutile (tutti gli eventi oltre il limite fissato non saranno mai candidati per l'insieme Lista).

4.2.2 Pseudocodice

1. function RICERCA-COSTO-UNIFORME(Catalogo, genere)
2. if Catalogo is empty then return Lista
3. if Catalogo.valutazione ≥ 3 then sort in descending order
4. valutazioneMin in $\leftarrow 3$
5. $i \leftarrow 0$
6. flag \leftarrow false
7. loop
8. if $i \geq$ Catalogo.size OR flag = true then end loop
9. if Catalogo[i] \geq valutazioneMin AND Catalogo[i].equals(genere) then Lista \leftarrow add(Catalogo[i])
10. if Catalogo[i] < valutazioneMin then flag \leftarrow true
11. $i \leftarrow i + 1$
12. goto loop
13. close
14. return Lista

4.2.3 Numero Confronti

La dimensione della lista in input non influisce sul numero di confronti, in quanto si considerano soltanto i videogiochi con una media della valutazione uguale o superiore a 3. Inoltre la lista è ordinata in ordine decrescente (in base alle valutazioni). Si ha dunque un numero di confronti nettamente minore.

4.2.4 Considerazioni

L'ordinamento in base alla valutazione comporta un'ottimizzazione al precedente algoritmo: verranno considerati prima i videogiochi che hanno maggior probabilità di far parte dell'insieme delle soluzioni.

5 Implementazione

Per l'implementazione sono stati utilizzati:

La classe Utente che offre i metodi necessari per gestire le informazioni relative all'utente, in particolare la tipologia di videogioco che preferisce;

La classe Videogioco che offre i metodi necessari per gestire le informazioni relative al videogioco, tra cui il genere e la media delle valutazioni degli utenti;

Un ArrayList di oggetti Videogioco per realizzare la frontiera (l'insieme dei nodi da esplorare);

La classe Decrescente che implementa l'interfaccia Comparator sulla classe Videogioco, utilizzata per l'ordinamento della collezione di videogiochi;

Una classe FillDatabase che consente il riempimento della base di dati settando in maniera casuale gli attributi degli utenti, dei videogiochi e delle recensioni.

5.1 Ricerca Lineare

```
public ArrayList<Videogioco> ricerca(ArrayList<Videogioco> catalogo, String genere){
    ArrayList<Videogioco> lista= new ArrayList<>();
    if (catalogo.isEmpty())
        return lista;
    int valutazioneMin=3;
    int i=0;
    System.out.println("Ricerca Lineare:\n valutazione Min: "+valutazioneMin);
    while(i<catalogo.size()){
        if(catalogo.get(i).getMediaValutazioni()>=valutazioneMin && catalogo.get(i).getTipologia().equals(genere))
            lista.add(catalogo.get(i));
        i++;
    }
    System.out.println("Numero di Videogiochi trovati:"+lista.size());
    System.out.println("Numero di Iterazioni effettuate:"+i);
    return lista;
}
```

5.2 Ricerca a Costo Uniforme

```
public ArrayList<Videogioco> ricerca(ArrayList<Videogioco> catalogo, String genere){
    ArrayList<Videogioco> lista= new ArrayList<>();
    if (catalogo.isEmpty())
        return lista;
    int valutazioneMin = 3;
    int i=0;
    boolean flag=false;
    System.out.println("Ricerca a Costo Uniforme:\n valutazioneMin: "+valutazioneMin);
    while(i < catalogo.size() && !flag){
        if (catalogo.get(i).getMediaValutazioni()>=valutazioneMin && catalogo.get(i).getTipologia().equals(genere))
            lista.add(catalogo.get(i));
        if (catalogo.get(i).getMediaValutazioni()<valutazioneMin)
            flag=true;
        i++;
    }
    System.out.println("Numero di Videogiochi trovati:"+lista.size());
    System.out.println("Numero di Iterazioni effettuate:"+i);
    return lista;
}
```

5.3 Decrescente

```
public class Decrescente implements Comparator<Videogioco> {  
    @Override  
    public int compare(Videogioco e1, Videogioco e2) {  
        if (Double.compare(e1.getMediaValutazioni(), e2.getMediaValutazioni())==0)  
            return 0;  
        if (Double.compare(e1.getMediaValutazioni(), e2.getMediaValutazioni())==1)  
            return 1;  
        else  
            return -1;  
    }  
}
```

5.4 FillDatabase

```
public class FillDatabase {  
  
    private static Random random = new Random();  
    private static int size = 1000;  
    private static List<String> titoli = new ArrayList<>();  
  
    /**  
     * questo è il metodo principale, il quale richiama gli altri metodi  
     */  
  
    public static void main() {  
        generateUtente();  
        generateVideogioco();  
        findVideogioco();  
        generateRecensione();  
    }  
}
```

6 Risultati

Per quanto riguarda il test degli algoritmi, sono stati eseguite diverse implementazioni variando la dimensione della lista dei videogiochi, la media della valutazione dei videogiochi e la tipologia di videogioco preferito dall'utente.

6.1 Test tipologia = "Horror"

6.1.1 Valutazione minima = 3 , size = 1010

```
Sort time: 8008000

Ricerca a Costo Uniforme:
ValutazioneMin: 3
Tipologia: Horror
Size catalogo: 1010
Numero di Videogiochi trovati:87
Numero di Iterazioni effettuate:598
Execution time: 3112200

Ricerca Lineare:
Valutazione Min: 3
Tipologia: Horror
Size catalogo: 1010
Numero di Videogiochi trovati:87
Numero di Iterazioni effettuate:1010
Execution time: 1580000
```

6.1.2 Valutazione minima = 4 , size = 1010

```
Sort time: 9062800

Ricerca a Costo Uniforme:
ValutazioneMin: 4
Tipologia: Horror
Size catalogo: 1010
Numero di Videogiochi trovati:53
Numero di Iterazioni effettuate:426
Execution time: 5320800

Ricerca Lineare:
Valutazione Min: 4
Tipologia: Horror
Size catalogo: 1010
Numero di Videogiochi trovati:53
Numero di Iterazioni effettuate:1010
Execution time: 1418500
```

6.1.3 Valutazione minima = 2 , size = 1010

```
Sort time: 8622600

Ricerca a Costo Uniforme:
ValutazioneMin: 2
Tipologia: Horror
Size catalogo: 1010
Numero di Videogiochi trovati:118
Numero di Iterazioni effettuate:817
Execution time: 1745200

Ricerca Lineare:
Valutazione Min: 2
Tipologia: Horror
Size catalogo: 1010
Numero di Videogiochi trovati:118
Numero di Iterazioni effettuate:1010
Execution time: 4099900
```

6.1.4 Valutazione minima = 3 , size = 510

```
Sort time: 5432700

Ricerca a Costo Uniforme:
ValutazioneMin: 3
Tipologia: Horror
Size catalogo: 510
Numero di Videogiochi trovati:41
Numero di Iterazioni effettuate:313
Execution time: 1298100

Ricerca Lineare:
Valutazione Min: 3
Tipologia: Horror
Size catalogo: 510
Numero di Videogiochi trovati:41
Numero di Iterazioni effettuate:510
Execution time: 1225400
```


6.1.5 Valutazione minima = 4 , size = 510

```
Sort time: 7890600

Ricerca a Costo Uniforme:
ValutazioneMin: 4
Tipologia: Horror
Size catalogo: 510
Numero di Videogiochi trovati:28
Numero di Iterazioni effettuate:208
Execution time: 4083400

Ricerca Lineare:
Valutazione Min: 4
Tipologia: Horror
Size catalogo: 510
Numero di Videogiochi trovati:28
Numero di Iterazioni effettuate:510
Execution time: 1775900
```

6.1.6 Valutazione minima = 2 , size = 510

```
Sort time: 9460300

Ricerca a Costo Uniforme:
ValutazioneMin: 2
Tipologia: Horror
Size catalogo: 510
Numero di Videogiochi trovati:63
Numero di Iterazioni effettuate:417
Execution time: 2135400

Ricerca Lineare:
Valutazione Min: 2
Tipologia: Horror
Size catalogo: 510
Numero di Videogiochi trovati:63
Numero di Iterazioni effettuate:510
Execution time: 2627500
```

6.2 Test tipologia = “Fantascienza”

6.2.1 Valutazione minima = 2 , size = 510

```
Sort time: 9440900

Ricerca a Costo Uniforme:
ValutazioneMin: 2
Tipologia: Fantascienza
Size catalogo: 510
Numero di Videogiochi trovati:71
Numero di Iterazioni effettuate:429
Execution time: 1527700

Ricerca Lineare:
Valutazione Min: 2
Tipologia: Fantascienza
Size catalogo: 510
Numero di Videogiochi trovati:71
Numero di Iterazioni effettuate:510
Execution time: 1870800
```

6.2.2 Valutazione minima = 3 , size = 510

```
Sort time: 6057700

Ricerca a Costo Uniforme:
ValutazioneMin: 3
Tipologia: Fantascienza
Size catalogo: 510
Numero di Videogiochi trovati:52
Numero di Iterazioni effettuate:323
Execution time: 1788400

Ricerca Lineare:
Valutazione Min: 3
Tipologia: Fantascienza
Size catalogo: 510
Numero di Videogiochi trovati:52
Numero di Iterazioni effettuate:510
Execution time: 2017000
```

6.2.3 Valutazione minima = 4 , size = 510

```
Sort time: 11406300

Ricerca a Costo Uniforme:
ValutazioneMin: 4
Tipologia: Fantascienza
Size catalogo: 510
Numero di Videogiochi trovati:31
Numero di Iterazioni effettuate:186
Execution time: 1069100

Ricerca Lineare:
Valutazione Min: 4
Tipologia: Fantascienza
Size catalogo: 510
Numero di Videogiochi trovati:31
Numero di Iterazioni effettuate:510
Execution time: 1222000
```

6.2.4 Valutazione minima = 4 , size = 1010

```
Sort time: 8732800

Ricerca a Costo Uniforme:
ValutazioneMin: 4
Tipologia: Fantascienza
Size catalogo: 1010
Numero di Videogiochi trovati:72
Numero di Iterazioni effettuate:422
Execution time: 1828400

Ricerca Lineare:
Valutazione Min: 4
Tipologia: Fantascienza
Size catalogo: 1010
Numero di Videogiochi trovati:72
Numero di Iterazioni effettuate:1010
Execution time: 1662100
```

6.2.5 Valutazione minima = 3 , size = 1010

```
Sort time: 10374500

Ricerca a Costo Uniforme:
ValutazioneMin: 3
Tipologia: Fantascienza
Size catalogo: 1010
Numero di Videogiochi trovati:80
Numero di Iterazioni effettuate:601
Execution time: 1528500

Ricerca Lineare:
Valutazione Min: 3
Tipologia: Fantascienza
Size catalogo: 1010
Numero di Videogiochi trovati:80
Numero di Iterazioni effettuate:1010
Execution time: 1471500
```

6.2.6 Valutazione minima = 2 , size = 1010

```
Sort time: 8056400

Ricerca a Costo Uniforme:
ValutazioneMin: 2
Tipologia: Fantascienza
Size catalogo: 1010
Numero di Videogiochi trovati:126
Numero di Iterazioni effettuate:808
Execution time: 1819200

Ricerca Lineare:
Valutazione Min: 2
Tipologia: Fantascienza
Size catalogo: 1010
Numero di Videogiochi trovati:126
Numero di Iterazioni effettuate:1010
Execution time: 1306300
```

6.3 Test tipologia = “Open World”

6.3.1 Valutazione minima = 3 , size = 2010

```
Sort time: 7272400

Ricerca a Costo Uniforme:
ValutazioneMin: 3
Tipologia: Open World
Size catalogo: 2010
Numero di Videogiochi trovati:194
Numero di Iterazioni effettuate:1202
Execution time: 1578000

Ricerca Lineare:
Valutazione Min: 3
Tipologia: Open World
Size catalogo: 2010
Numero di Videogiochi trovati:194
Numero di Iterazioni effettuate:2010
Execution time: 1825900
```

7 Conclusioni

Osservando i risultati dei test si evince dal numero di confronti che la ricerca a costo uniforme ne effettua di meno rispetto a quella lineare, questa differenza è ancora più evidente controllando i tempi di esecuzione dei due algoritmi. Considerando inoltre il tempo di esecuzione dell'ordinamento richiesto dalla ricerca a costo uniforme, risulta che il tempo di esecuzione è persino superiore a quello della ricerca lineare. A causa della scarsa presenza di informazioni utilizzabili, abbiamo optato per un approccio ad algoritmi più semplici. In particolare la nostra ricerca si è soffermata sul parametro “tipologia”, ovvero il genere di videogioco preferito dall'utente. La presenza di altre informazioni sarebbe stata sicuramente utile per l'individuazione di algoritmi più complessi ed efficienti.