

**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Luka Hrupec

IGRA INTERAKTIVNE FIKCIJE - INFORM 7

PROJEKT

Varaždin, 2026.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Luka Hrupec

Matični broj: 1608003320016

Studij: Baze podataka i baze znanja

IGRA INTERAKTIVNE FIKCIJE - INFORM 7

PROJEKT

Mentor:

doc. dr. sc. Bogdan Okreša Đurić

Varaždin, siječanj 2026.

Luka Hrupec

Izjava o izvornosti

Izjavljujem da je ovaj projekt izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI Radovi

Sažetak

U ovom projektu prikazana je implementacija interaktivne tekstualne igre korištenjem deklarativne paradigme programiranja. Aplikacija je razvijena u programskom jeziku Inform 7, koji omogućuje definiranje logike sustava pomoću pravila, predikata i stanja svijeta izraženih u obliku prirodnog jezika. Razvijena igra temelji se na modeliranju prostora, objekata i vremenski ovisnih događaja, pri čemu se ponašanje aplikacije određuje skupom pravila koja se aktiviraju ovisno o stanju sustava i interakcijama korisnika. Poseban naglasak stavljen je na implementaciju turn-based mehanizma te uvjetovanih događaja koji se evaluiraju automatski kroz sustav pravila jezika Inform 7. Projekt uključuje i integraciju multimedijskih elemenata poput slika i zvukova, čime se dodatno obogaćuje korisničko iskustvo. U radu su obrađene teorijske osnove deklarativnog i logičkog programiranja te je analizirana njihova prikladnost za razvoj interaktivnih sustava.

Ključne riječi: deklarativno programiranje; logičko programiranje; Inform 7; interaktivna fikcija; sustavi temeljeni na pravilima; vremenski ovisni sustavi; računalne igre

Sadržaj

1. Uvod	1
1.1. Ciljevi rada	1
1.2. Struktura rada	1
2. Teorijska podloga	2
2.1. Deklarativno programiranje	2
2.2. Logičko programiranje	2
2.3. Predikatni račun	3
3. Razrada teme	4
3.1. Modeliranje prostora i kretanja	4
3.2. Definiranje prostorija i elemenata u prostoru	4
3.3. Definiranje objekata i scenografije	5
3.4. Interaktivni objekti i spremnici	5
3.5. Neigrački likovi (NPC-ovi)	6
3.6. Osnovna komunikacija s NPC-ovima	6
3.7. Uvjetovani dijalozi i grananje razgovora	7
3.8. Uloga NPC-ova u napredovanju igre	7
3.9. Stanja sustava i bilježenje napretka	7
3.9.1. Stanja objekata i tragova	7
3.9.2. Stanja neigračkih likova	8
3.9.3. Sličnosti i zajednički mehanizam stanja	8
3.10. Mehanizmi odlučivanja i kontrola ishoda	9
3.11. Mehanizmi odlučivanja i kontrola ishoda	9
3.12. Bilježenje tragova i sustav bilješki	10
3.13. Završeci igre i evaluacija uspješnosti	10
4. Prikaz rada aplikacije	11
4.1. Skup dostupnih korisničkih naredbi	11
4.1.1. Kretanje između prostorija	11
4.1.2. Pregled prostora i objekata	11
4.1.3. Interakcija s neigračkim likovima	12
4.1.4. Bilješke i praćenje istrage	12
4.1.5. Optuživanje i završne naredbe	12
4.2. Početak igre i inicijalno stanje sustava	13

4.3. Istraživanje prostora i otkrivanje tragova	13
4.4. Interakcija s neigračkim likovima tijekom istrage	14
4.5. Korištenje bilješki i analiza prikupljenih informacija	15
4.6. Donošenje odluke i završetak igre	15
4.7. Zaključak prikaza rada	15
5. Kritički osvrt	17
5.1. Prednosti primijenjenog pristupa	17
5.2. Ograničenja i izazovi	17
5.3. Moguća poboljšanja i budući razvoj	18
6. Zaključak	19
Popis literature	20
Popis slika	21
Popis isječaka koda	22

1. Uvod

Razvoj suvremenih programskih sustava sve se češće suočava s problemima modeliranja složenog ponašanja, upravljanja stanjem te definiranja pravila koja ovise o kontekstu i vremenu izvođenja. U takvim slučajevima imperativni pristup, koji se temelji na eksplisitnom upravljanju tokom programa, može dovesti do nepreglednog i teško održivog koda. Kao alternativa, deklarativna paradigma programiranja omogućuje opisivanje problema na višoj razini apstrakcije, pri čemu se naglasak stavlja na definiranje pravila i odnosa, a ne na detalje njihove provedbe.

Interaktivne tekstualne igre predstavljaju prikladnu domenu za primjenu deklarativnog programiranja jer se njihova logika prirodno izražava kroz pravila, uvjete i promjene stanja svijeta. Takve igre temelje se na interpretaciji korisničkih naredbi, evaluaciji trenutnog stanja sustava te aktivaciji odgovarajućih događaja, što ih čini pogodnima za modeliranje pomoću logičkog programiranja. Programski jezik Inform 7 je korišten jer omogućuje definiranje logike sustava korištenjem prirodnog jezika, čime se dodatno povećava čitljivost i razumljivost programskega koda.

1.1. Ciljevi rada

Glavni cilj rada je implementirati interaktivnu tekstualnu igru korištenjem deklarativne paradigmе programiranja te analizirati prikladnost tog pristupa za razvoj sustava temeljenih na pravilima i stanju svijeta. U sklopu tog cilja postavljeni su sljedeći specifični ciljevi: teorijski obraditi osnovne pojmove deklarativnog i logičkog programiranja, prikazati način modeliranja prostora, objekata i stanja sustava u jeziku Inform 7, implementirati vremenski ovisne mehanizme i uvjetovane događaje te integrirati multimedijalne elemente radi poboljšanja korisničkog iskustva. Dodatni cilj rada je kritički analizirati prednosti i ograničenja primijenjenog pristupa u odnosu na alternativna rješenja.

1.2. Struktura rada

Rad je strukturiran u nekoliko poglavlja. U uvodnom poglavlju predstavljena je motivacija za odabir teme te su definirani ciljevi rada. U sljedećem poglavlju obrađene su teorijske osnove deklarativnog i logičkog programiranja, s naglaskom na formalizam koji se koristi u implementaciji. Nakon toga slijedi opis problema i koncepta rješenja te detaljan prikaz implementacije aplikacije, uključujući korištene tehnologije i ključne mehanizme sustava. Posebno poglavlje posvećeno je prikazu rada aplikacije kroz primjere korištenja. U kritičkom osvrtu analiziraju se prednosti i ograničenja primijenjenog pristupa, dok se u zaključku sažimaju postignuti rezultati i iznose smjernice za mogući daljnji razvoj.

2. Teorijska podloga

U ovom poglavlju prikazane su teorijske osnove na kojima se temelji implementacija razvijene aplikacije. Poseban naglasak stavljen je na deklarativnu paradigmu programiranja i logičko programiranje, budući da predstavljaju temeljni konceptualni okvir korišten u izradi projekta. Cilj poglavlja je objasniti osnovne principe navedenih pristupa te uspostaviti vezu između teorijskog formalizma i njegove praktične primjene u razvoju interaktivnih sustava.

2.1. Deklarativno programiranje

Deklarativno programiranje predstavlja programsku paradigmu u kojoj se programski problem opisuje kroz skup tvrdnji koje definiraju željeno ponašanje sustava, bez eksplicitnog navođenja redoslijeda izvršavanja pojedinih koraka [1]. Za razliku od imperativnog pristupa, u kojem programer detaljno specificira kako se zadatak treba izvršiti, deklarativni pristup usmjeren je na opis onoga što sustav treba postići.

Jedna od ključnih prednosti deklarativnog programiranja jest viša razina apstrakcije. Programer se može usredotočiti na domensku logiku i pravila ponašanja, dok izvršno okruženje preuzima odgovornost za upravljanje tokom izvođenja i evaluaciju uvjeta. Time se smanjuje složenost koda i olakšava razumijevanje programskog rješenja, osobito u sustavima koji uključuju velik broj međusobno ovisnih uvjeta [1].

Deklarativni pristup pokazuje se posebno prikladnim u domenama u kojima je potrebno modelirati složene odnose i pravila, kao što su ekspertni sustavi, sustavi poslovnih pravila i interaktivne aplikacije. Promjene u logici sustava često se mogu provesti dodavanjem ili izmjenom pojedinih pravila, bez potrebe za opsežnim promjenama postojećeg koda, što doprinosi modularnosti i proširivosti rješenja.

2.2. Logičko programiranje

Logičko programiranje je podvrsta deklarativnog programiranja koja se temelji na formalnoj logici. U ovom pristupu program se sastoji od skupa činjenica i pravila, a izvršavanje programa podrazumijeva izvođenje logičkih zaključaka na temelju zadanih tvrdnji [2]. Sustav samostalno pronalazi rješenja koja zadovoljavaju definirane uvjete, bez potrebe za eksplicitnim upravljanjem algoritamskim postupkom.

Osnovna ideja logičkog programiranja jest da se znanje o problemu izražava kroz logičke odnose, dok mehanizam zaključivanja određuje koje se posljedice mogu izvesti iz postojećih činjenica. Takav pristup omogućuje elegantno modeliranje problema u kojima se ponašanje sustava određuje kombinacijom različitih uvjeta i stanja [2].

Logičko programiranje posebno je pogodno za primjene u kojima se zahtijeva fleksibilnost i prilagodljivost ponašanja sustava. Budući da su pravila neovisna jedno o drugome, moguće je postupno proširivati sustav dodavanjem novih pravila, bez narušavanja postojećeg

funkcioniranja. Ova značajka čini logičko programiranje prikladnim izborom za razvoj interaktivnih aplikacija temeljenih na pravilima.

2.3. Predikatni račun

Predikatni račun predstavlja formalni logički sustav koji omogućuje izražavanje tvrdnji o objektima, njihovim svojstvima i međusobnim odnosima. Za razliku od iskaznog računa, koji se bavi jednostavnim istinitosnim vrijednostima, predikatni račun omogućuje opis složenijih struktura kroz uporabu predikata, varijabli i kvantifikatora [3].

U kontekstu programiranja, predikatni račun pruža formalnu osnovu za modeliranje znanja i donošenje zaključaka. Činjenice se mogu promatrati kao osnovne tvrdnje o stanju sustava, dok se pravila koriste za izvođenje novih zaključaka na temelju postojećih informacija. Takav formalizam omogućuje precizno i nedvosmisленo definiranje ponašanja sustava.

Primjena predikatnog računa osobito je izražena u jezicima i sustavima koji se temelje na logičkom programiranju. Korištenjem predikata moguće je prirodno opisati odnose između elemenata sustava, što je posebno korisno u aplikacijama koje se oslanjaju na upravljanje stanjem i uvjetovano ponašanje [3]. Ova teorijska podloga čini važan temelj za razumijevanje načina na koji se deklarativni jezici koriste u praksi.

3. Razrada teme

U ovom poglavlju opisan je problem koji se rješava razvijenom aplikacijom te način na koji je deklarativna paradigma programiranja primjenjena u njegovoj implementaciji. Poseban naglasak stavljen je na modeliranje prostora, stanja sustava, interakcije s korisnikom i vremenski ovisne događaje. Opisi su potkrijepljeni isjećcima programskog koda kako bi se jasnije prikazala veza između teorijskih koncepcata i njihove praktične realizacije.

Programski jezik Inform 7 korišten je jer omogućuje definiranje logike sustava korištenjem prirodnog jezika, pri čemu se svijet igre modelira kao skup objekata, pravila i odnosa između njih, što je detaljno opisano u službenoj dokumentaciji jezika [4].

3.1. Modeliranje prostora i kretanja

Prostor igre modeliran je kao skup međusobno povezanih prostorija. Svaka prostorija ima vlastiti opis, a odnosi među prostorijama definirani su deklarativno, bez eksplicitnog upravljanja kretanjem igrača. Time se omogućuje prirodna navigacija kroz igru, pri čemu izvršno okruženje samo određuje valjanost pojedinog poteza.

```
The Entrance Hall is a room.  
The Bathroom is west of the Entrance Hall.  
The Bedroom is east of the Entrance Hall.  
The Kitchen is north of the Entrance Hall.  
The Living Room is south of the Entrance Hall.  
The Basement is down from the Entrance Hall.
```

Isječak koda 1: Deklarativno modeliranje prostora i odnosa između prostorija

3.2. Definiranje prostorija i elemenata u prostoru

Svaka prostorija u igri definirana je kao zasebna cjelina koja sadrži opis okruženja i pripadajuće elemente s kojima korisnik može stupiti u interakciju. Opisi prostorija imaju važnu ulogu jer korisniku pružaju kontekst, atmosferu i osnovne informacije potrebne za razumijevanje situacije u kojoj se nalazi. U deklarativnom pristupu, opis prostorije predstavlja dio modela svijeta, a ne samo tekstualni ispis.

U programskom jeziku Inform 7 prostorije se definiraju jednostavnim deklarativnim izrazima, pri čemu se opis prostorije navodi kao tekstualni atribut. Takav način definicije omogućuje jasnu razdvojenost između strukture prostora i njegove semantičke interpretacije. Opisi se automatski prikazuju korisniku prilikom ulaska u prostoriju ili korištenjem naredbe za promatranje okoline, bez potrebe za dodatnim upravljačkim kodom. Opis prostorije koristi se za stvaranje narativnog okvira igre te usmjeravanje pažnje korisnika na važne elemente okoline. U prikazanom primjeru naglasak je stavljen na tijelo žrtve i prisutnost lika, čime se igraču odmah sugerira važnost prostora za daljnju istragu. Time se narativni elementi prirodno povezuju s

```
The Kitchen is a room.  
"The kitchen looks disturbed. Edward Blackwood lies motionless on the floor,  
a knife stuck in his back. A cook is also present in the room, visibly shaken  
by the scene."
```

Isječak koda 2: Definicija prostorije s pripadajućim opisom

logikom igre.

3.3. Definiranje objekata i scenografije

Objekti unutar prostorija modelirani su kao zasebni entiteti koji mogu imati različite uloge u igri. Neki objekti služe isključivo kao scenografija i nemaju izravan utjecaj na tijek igre, dok drugi predstavljaju ključne tragove ili interaktivne elemente koji mijenjaju stanje sustava. Razlikovanje tih kategorija omogućuje preciznije upravljanje interakcijama i logikom igre.

Scenografski objekti definirani su kao elementi koji ne zahtijevaju dodatnu obradu događaja, ali doprinose realizmu i uvjerljivosti okruženja. Takvi objekti označeni su kao *scenery*, čime se jasno daje do znanja da su sastavni dio prostora i da se ne mogu uzimati ili premještati. Ovakav pristup omogućuje bogat opis prostora bez nepotrebnog opterećivanja logike igre. Ko-

```
The fridge is scenery in the Kitchen.  
The description is "A large refrigerator. Nothing unusual."  
  
The oven is scenery in the Kitchen.  
The description is "A modern oven. It is turned off."
```

Isječak koda 3: Primjer scenografskih elemenata u prostoriji

risnik može istraživati okolinu i dobivati dodatne informacije, dok se istovremeno jasno razlikuju elementi koji imaju utjecaj na rješavanje problema od onih koji služe isključivo za atmosferu.

3.4. Interaktivni objekti i spremnici

Osim scenografskih elemenata, u igri su definirani i interaktivni objekti koji omogućuju napredovanje u istrazi. Takvi objekti često su spremnici ili skrivene strukture koje mogu sadržavati druge elemente, poput ključeva, dokumenata ili tragova. Njihovo ponašanje definirano je skupom svojstava i pravila koja određuju kada i na koji način korisnik može stupiti u interakciju s njima. Korištenjem spremnika i hijerarhijskih odnosa između objekata omogućuje se implemen-

```
The wardrobe is a container in the Bedroom.  
It is openable and closed.  
  
The false bottom is part of the wardrobe.  
The false bottom is undescribed.
```

Isječak koda 4: Definicija spremnika i skrivenog elementa

tacija složenijih interakcija bez potrebe za eksplisitnim upravljanjem stanjem svakog elementa. Sustav pravila automatski upravlja dostupnošću objekata ovisno o njihovim svojstvima, čime se dodatno potvrđuje prednost deklarativnog pristupa u razvoju interaktivnih sustava.

3.5. Neigrački likovi (NPC-ovi)

Neigrački likovi (engl. *Non-Player Characters – NPC*) predstavljaju ključni element igre jer korisniku pružaju informacije potrebne za napredovanje u istrazi. Svaki lik ima vlastitu ulogu u narativu te skup dijaloga koji se aktiviraju ovisno o interakcijama korisnika i trenutačnom stanju sustava. Umjesto unaprijed definiranog slijeda razgovora, ponašanje likova određeno je pravilima i uvjetima, što omogućuje fleksibilnu i nelinearnu komunikaciju.

U razvijenoj igri definirana su četiri osnovna lika: sobarica, kuhar, supruga i sin žrtve. Svaki od likova inicijalno se nalazi u određenoj prostoriji, čime se dodatno naglašava prostorni kontekst njihove uloge u priči. Likovi su modelirani kao osobe s osnovnim svojstvima, dok

```
The Maid is a woman.  
The Cook is a man.  
The Wife is a woman.  
The Son is a man.  
  
The Maid is in the Bathroom.  
The Cook is in the Kitchen.  
The Wife is in the Living Room.
```

Isječak koda 5: Definicija neigračkih likova i njihove početne pozicije

se njihovo ponašanje i interakcije dodatno definiraju pravilima. Takav pristup omogućuje jednostavno proširivanje igre dodavanjem novih likova ili promjenom postojećih bez narušavanja cjelokupne strukture sustava.

3.6. Osnovna komunikacija s NPC-ovima

Komunikacija s neigračkim likovima ostvarena je putem prilagođenih naredbi koje omogućuju razgovor s odabranim likom. Umjesto korištenja standardnih naredbi, definirana je nova akcija koja prepoznaje različite oblike korisničkih upita, čime se poboljšava prirodnost interakcije i smanjuje potreba za strogo definiranim naredbama. Ovakva definicija omogućuje korisniku da

```
Talking to is an action applying to one visible thing.  
  
Understand "talk to [someone]" or  
"talk [someone]" or  
"speak to [someone]" as talking to.
```

Isječak koda 6: Definicija akcije za razgovor s NPC-ovima

koristi različite prirodne izraze za istu radnju, dok sustav interno mapira te izraze na jedinstvenu akciju. Time se postiže veća fleksibilnost i pristupačnost sučelja.

3.7. Uvjetovani dijalozi i grananje razgovora

Dijalozi s neigračkim likovima nisu statični, već se prilagođavaju ovisno o informacijama koje je korisnik prethodno otkrio. Pojedini odgovori dostupni su tek nakon što su ispunjeni određeni uvjeti, čime se potiče istraživanje prostora i povezivanje tragova. Ovakav pristup omogućuje grananje razgovora bez potrebe za linearnim dijaloškim stablima. U prikazanom

```
Instead of asking the Cook about something:  
    if (the topic understood includes "knife" or  
        the topic understood includes "knives") and  
        the knife-examined is true:  
            say "The cook glances at the body. 'Yes,' he says quietly.  
                  'That knife is one of mine.'";  
    else:  
        say "The cook shrugs. 'I don't really know anything about that.'";
```

Isječak koda 7: Primjer uvjetovanog dijaloga temeljenog na stanju sustava

primjeru odgovor lika ovisi o tome je li korisnik prethodno pregledao određeni objekt. Time se osigurava da se informacije otkrivaju postupno i u skladu s logikom narativa, bez narušavanja dosljednosti priče.

3.8. Uloga NPC-ova u napredovanju igre

Neigrački likovi ne služe samo kao izvor informacija, već i kao aktivni sudionici u napredovanju igre. Njihova ponašanja i reakcije mogu potaknuti daljnje radnje korisnika, otkriti nove tragove ili utjecati na konačni ishod igre. Kroz uvjetovane dijaloge i promjene stanja, NPC-ovi postaju sastavni dio sustava odlučivanja i zaključivanja unutar igre.

3.9. Stanja sustava i bilježenje napretka

Za praćenje napretka korisnika i donošenje odluka unutar igre koristi se skup logičkih varijabli koje predstavljaju stanje sustava. Ta stanja omogućuju pamćenje prethodnih radnji, otkrivenih tragova i ponašanja likova, čime se osigurava dosljednost narativa i logike igre. U jeziku Inform 7 takva stanja najčešće se implementiraju pomoću logičkih varijabli koje se mogu mijenjati tijekom izvođenja programa.

3.9.1. Stanja objekata i tragova

Stanja objekata koriste se za bilježenje informacija o tome je li korisnik otkrio određeni trag ili stupio u interakciju s važnim elementom okoline. Takva stanja predstavljaju temeljni mehanizam za kontrolu dostupnosti informacija i uvjetovanje dalnjih događaja. Svaki važniji objekt ili trag povezan je s pripadajućom logičkom varijablom koja označava je li taj element već primijećen ili iskorišten. Promjena stanja objekta najčešće se događa kao posljedica korisničke radnje, poput pregleda ili korištenja određenog elementa. Na taj način sustav pamti koje su

```
The body-noted is a truth state that varies.  
The knives-noted is a truth state that varies.  
The power-noted is a truth state that varies.  
The safe-noted is a truth state that varies.  
The letter-noted is a truth state that varies.
```

Isječak koda 8: Logička stanja povezana s objektima i tragovima

informacije već dostupne korisniku, a koje tek trebaju biti otkrivene. Ovakav pristup omogućuje

```
After examining the bathtub:  
    now the blood-noted is true;  
    say "Upon closer inspection, you notice a small dried blood stain.";
```

Isječak koda 9: Promjena stanja objekta nakon interakcije

da se tragovi otkrivaju postupno i u logičnom redoslijedu, bez potrebe za dodatnim kontrolnim strukturama. Pravila se automatski aktiviraju ovisno o trenutačnom stanju sustava, što je u skladu s deklarativnom paradigmom.

3.9.2. Stanja neigračkih likova

Osim objekata, stanja se koriste i za praćenje ponašanja i emocionalnih reakcija neigračkih likova. Takva stanja omogućuju modeliranje sumnje, straha ili agresivnosti likova te izravno utječu na dostupne dijaloge i moguće ishode igre. Iako se semantički razlikuju od stanja objekata, tehnički su implementirana na identičan način. Promjene stanja likova najčešće

```
The maid-suspicious is a truth state that varies.  
The wife-suspicious is a truth state that varies.  
The son-angry is a truth state that varies.  
The cook-afraid is a truth state that varies.
```

Isječak koda 10: Logička stanja povezana s ponašanjem neigračkih likova

su posljedica razgovora ili otkrivanja određenih informacija. Time se ponašanje likova prilagođava tijeku istrage, a korisniku se omogućuje dinamična interakcija umjesto unaprijed zadanih odgovora.

```
Instead of asking the Maid about something:  
    now the maid-suspicious is true;  
    say "She hesitates and avoids your gaze.";
```

Isječak koda 11: Promjena stanja neigračkog lika tijekom dijaloga

3.9.3. Sličnosti i zajednički mehanizam stanja

Iako se stanja objekata i stanja neigračkih likova semantički razlikuju, oba se oslanjaju na isti tehnički mehanizam. U oba slučaja koriste se logičke varijable koje se mijenjaju tijekom

igre i utječu na aktivaciju pravila i dostupnost sadržaja. Takav jedinstven pristup pojednostavljuje implementaciju i omogućuje dosljedno upravljanje složenim ponašanjem sustava.

Korištenjem istog formalizma za različite aspekte igre postiže se veća preglednost koda i lakša nadogradnja sustava. Novi tragovi ili ponašanja likova mogu se dodavati bez narušavanja postojećih pravila, čime se dodatno potvrđuje prikladnost deklarativnog programiranja za ovakvu vrstu aplikacija.

3.10. Mehanizmi odlučivanja i kontrola ishoda

Mehanizmi odlučivanja u igri omogućuju korisniku da na temelju prikupljenih informacija doneše zaključke i poduzme ključne radnje koje vode prema završetku igre. Umjesto linearнog slijeda događaja, odluke se evaluiraju na temelju trenutačnog stanja sustava, uključujući otkrivene tragove i ponašanje neigračkih likova. Takav pristup omogućuje postojanje više mogućih ishoda, ovisno o kvaliteti i redoslijedu korisnikovih odluka.

Ključni mehanizam odlučivanja implementiran je kroz akciju optuživanja, kojom korisnik formalno označava osumnjičenog lika. Prije nego što je optužba dopuštena, sustav provjerava jesu li ispunjeni preduvjeti, poput prethodnog ispitivanja lika. Time se sprječava proizvoljno donošenje odluka i potiče sustavno prikupljanje informacija.

3.11. Mehanizmi odlučivanja i kontrola ishoda

Mehanizmi odlučivanja u igri omogućuju korisniku da na temelju prikupljenih informacija doneše zaključke i poduzme ključne radnje koje vode prema završetku igre. Umjesto linearнog slijeda događaja, odluke se evaluiraju na temelju trenutačnog stanja sustava, uključujući otkrivene tragove i ponašanje neigračkih likova. Takav pristup omogućuje postojanje više mogućih ishoda, ovisno o kvaliteti i redoslijedu korisnikovih odluka.

Ključni mehanizam odlučivanja implementiran je kroz akciju optuživanja, kojom korisnik formalno označava osumnjičenog lika. Prije nego što je optužba dopuštena, sustav provjerava jesu li ispunjeni preduvjeti, poput prethodnog ispitivanja lika. Time se sprječava proizvoljno donošenje odluka i potiče sustavno prikupljanje informacija. Nakon što su uvjeti zadovoljeni,

Check accusing someone when the noun is unquestioned:
say "You cannot accuse [the noun] without questioning them first." instead.

Isječak koda 12: Provjera preduvjeta prije donošenja odluke o optuživanju

optužba mijenja stanje lika i omogućuje daljnju eskalaciju događaja. Sustav potom, na temelju dodatnih stanja, određuje konačni ishod igre.

3.12. Bilježenje tragova i sustav bilješki

Kako bi korisniku bio omogućen pregled prikupljenih informacija, implementiran je sustav bilješki koji dinamički prikazuje otkrivene tragove i zapažanja. Sustav bilješki ne pohranjuje podatke zasebno, već se oslanja na postojeća stanja sustava, čime se izbjegava redundancija i osigurava dosljednost prikazanih informacija.

Prikaz bilješki temelji se na uvjetnom ispisu sadržaja ovisno o trenutačnim vrijednostima logičkih varijabli. Time se osigurava da korisnik vidi samo one informacije koje je doista otkrio tijekom igre. Ovakav pristup omogućuje korisniku da u svakom trenutku provjeri tijek istrage

```
Carry out reading notes:  
    say "[bold type]CASE NOTES:[roman type][line break]";  
    if body-noted is true, say "- Victim found in the kitchen.[line break]";  
    if knives-noted is true, say "- Two knives are missing.[line break]";  
    if power-noted is true, say "- A power outage occurred.[line break]";
```

Isječak koda 13: Uvjetovani prikaz bilješki na temelju stanja sustava

bez narušavanja narativa ili potrebe za dodatnim memorijskim strukturama. Sustav bilješki tako djeluje kao refleksija stanja svijeta, a ne kao zaseban modul.

3.13. Završeci igre i evaluacija uspješnosti

Igra sadrži više mogućih završetaka koji ovise o donesenim odlukama i prikupljenim informacijama. Završeci su implementirani kroz skup pravila koja evaluiraju stanje sustava u trenutku ključne odluke, poput uhićenja osumnjičenog lika. Time se osigurava da ishod igre bude izravna posljedica korisnikovih postupaka.

Uspješan završetak moguć je samo ako korisnik poveže ključne tragove i donese ispravnu odluku. Pogrešne ili preuranjene odluke dovode do alternativnih završetaka, čime se dodatno naglašava važnost sustavnog pristupa istrazi. Korištenjem istog mehanizma stanja za

```
Carry out arresting the Maid:  
    if the letter-read is true:  
        end the story finally saying "CASE CLOSED";  
    else:  
        end the story finally saying "CASE FAILED";
```

Isječak koda 14: Određivanje završetka igre na temelju stanja sustava

odlučivanje o završecima igre postiže se jasna veza između korisničkih odluka i njihovih posljedica. Time se osigurava konzistentnost sustava i povećava osjećaj odgovornosti korisnika za ishod igre.

4. Prikaz rada aplikacije

U ovom poglavlju prikazan je način rada razvijene igre kroz tipičan scenarij korištenja. Cilj prikaza je ilustrirati kako korisnik, kroz istraživanje prostora, interakciju s objektima i razgovore s neigračkim likovima, postupno prikuplja informacije i donosi odluke koje vode prema rješenju slučaja. Prikaz rada aplikacije ne opisuje sve moguće putanje, već jednu reprezentativnu istragu koja demonstrira ključne mehanizme sustava.

4.1. Skup dostupnih korisničkih naredbi

Interakcija korisnika s aplikacijom odvija se putem tekstualnih naredbi koje sustav interpretira i povezuje s odgovarajućim pravilima i stanjima svijeta. Naredbe su oblikovane tako da nalikuju prirodnom jeziku, čime se smanjuje potreba za učenjem formalne sintakse i omogućuje intuitivno upravljanje igrom. U nastavku je prikazan pregled najvažnijih naredbi dostupnih korisniku tijekom igranja.

4.1.1. Kretanje između prostorija

Kretanje kroz prostor igre ostvaruje se korištenjem osnovnih smjernih naredbi. Svaka prostorija povezana je s jednom ili više susjednih lokacija, a sustav automatski provjerava ispravnost kretanja ovisno o trenutačnoj poziciji korisnika.

- west, east, north, south – kretanje između prostorija na istoj razini
- down, up – kretanje između katova (ulazni hodnik i podrum)

Primjerice, iz ulaznog hodnika moguće je krenuti prema kupaonici, kuhinji, spavaćoj sobi, dnevnoj sobi ili podrumu, dok se povratak iz podruma omogućuje samo ako su ispunjeni uvjeti vidljivosti. Važno je napomenuti da se vraćamo u prethodnu sobu sa naredbom suprotog smjera, npr. ako smo u kuhinju došli sa north, iz nje se vraćamo pomoću south.

4.1.2. Pregled prostora i objekata

Za istraživanje okoline i prikupljanje tragova korisnik koristi naredbe za pregled prostorija i pojedinih objekata. Ove naredbe predstavljaju temeljni mehanizam za otkrivanje novih informacija.

- look – prikaz opisa trenutačne prostorije
- examine <object> – detaljan pregled objekta
- open <container> – otvaranje spremnika

Dostupni objekti ovise o prostoriji u kojoj se korisnik nalazi, čime se dodatno naglašava kontekstualna priroda interakcije.

4.1.3. Interakcija s neigračkim likovima

Komunikacija s neigračkim likovima odvija se kroz razgovore i ispitivanja. Sustav podržava slobodno formulirane teme razgovora, pri čemu se odgovori likova prilagođavaju trenutačnom stanju istrage.

- talk to <npc> – započinjanje osnovnog razgovora s likom
- ask <npc> about <topic> – ispitivanje lika o određenoj temi
- search <npc> – pretraživanje lika, koje može otkriti skrivenе predmete ili pokrenuti kritične događaje

Odgovori likova mogu se mijenjati ovisno o prethodnim radnjama korisnika, primjerice nakon otkrivanja određenih tragova ili objekata. Pretraživanje likova predstavlja rizičnu, ali potencijalno ključnu radnju. U određenim slučajevima ona može dovesti do pronalaska važnih dokaza, dok u drugim situacijama može rezultirati nasilnom reakcijom lika i trenutnim završetkom igre. Takav mehanizam potiče korisnika na promišljeno donošenje odluka i procjenu rizika.

4.1.4. Bilješke i praćenje istrage

Kako bi korisnik imao pregled dosadašnjeg tijeka istrage, implementirana je naredba za prikaz bilješki koje sažimaju prikupljene informacije.

- notes – prikaz trenutačnih bilješki i otkrivenih tragova

Prikaz bilješki temelji se isključivo na stanju sustava, čime se osigurava da su prikazane samo one informacije koje je korisnik stvarno otkrio tijekom igre.

4.1.5. Optuživanje i završne naredbe

Završna faza igre uključuje donošenje odluka koje izravno utječe na ishod istrage. Te se odluke provode pomoću specijaliziranih naredbi koje pokreću evaluaciju stanja sustava.

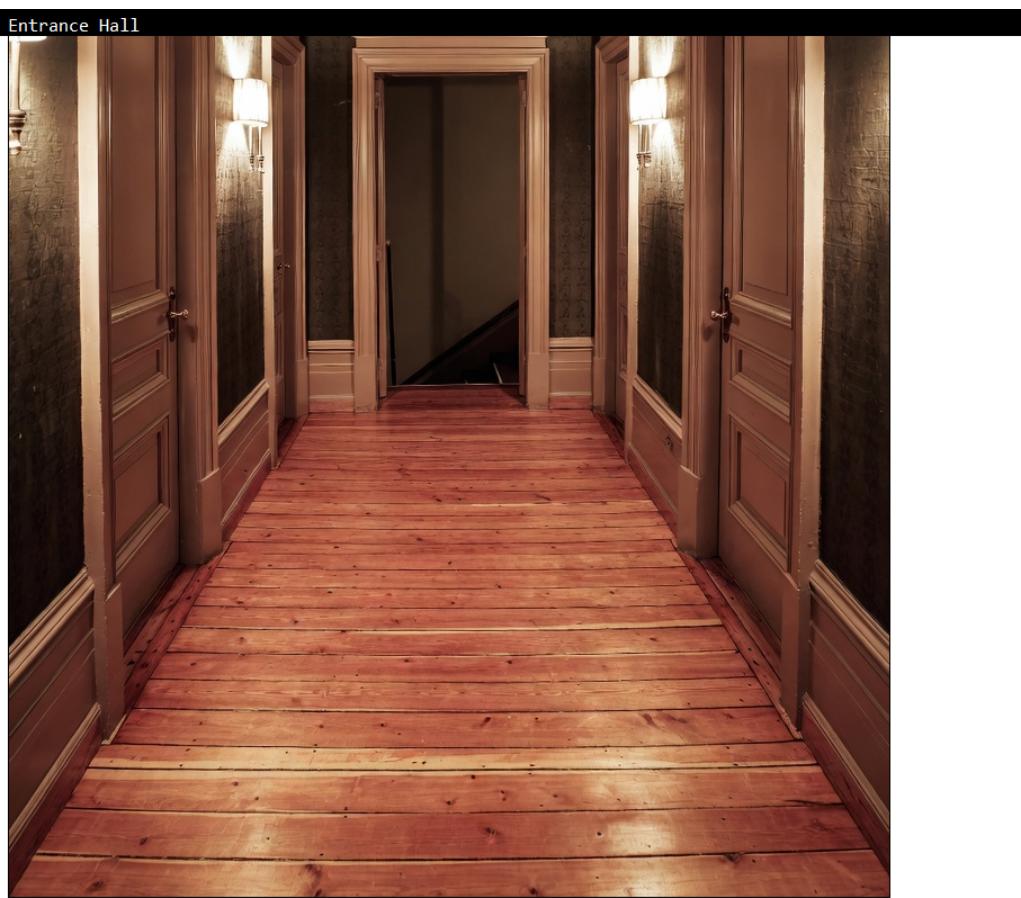
- accuse <npc> – formalno optuživanje lika
- arrest <npc> – uhićenje optuženog lika
- leave – napuštanje kuće i prekid istrage

Korištenje ovih naredbi dovodi do jednog od mogućih završetaka igre, ovisno o prikupljenim dokazima i prethodnim odlukama korisnika.

4.2. Početak igre i inicijalno stanje sustava

Nakon pokretanja aplikacije korisniku se prikazuje uvodna poruka koja postavlja narativni kontekst i definira osnovni zadatak istrage. Igra započinje u ulaznom hodniku kuće, koji predstavlja središnju lokaciju iz koje je moguće pristupiti svim ostalim prostorijama. U ovom trenutku sva stanja sustava inicijalizirana su na početne vrijednosti, čime se osigurava konzistentan početak svakog novog igranja.

Prilikom prvog prikaza lokacije sustav automatski prikazuje odgovarajuću sliku i reproducira zvučni efekt, čime se tekstualna interakcija nadopunjuje multimedijskim elementima. Time se povećava osjećaj uronjenosti korisnika u virtualni svijet, bez narušavanja temeljne tekstualne prirode igre. Na slici 1 prikazan je početni izgled ulaznog hodnika pri pokretanju igre.



A violent storm rages outside. POLICE REPORT (confidential): Mr. Blackwood, a wealthy businessman, has been murdered. The crime took place inside his house. Several people were present at the time of the incident. Your task is to uncover the truth.

Slika 1: Početni prikaz ulaznog hodnika pri pokretanju igre

4.3. Istraživanje prostora i otkrivanje tragova

Korisnik tijekom igre istražuje prostor kretanjem između prostorija i pregledavanjem dostupnih objekata. Svaka prostorija ima vlastiti opis koji se prikazuje pri ulasku, dok se dodatne

informacije otkrivaju detaljnijim pregledom pojedinih elemenata. Takav pristup potiče korisnika na aktivno istraživanje i pažljivo čitanje tekstualnih opisa.

Otkrivanje tragova provodi se kroz interakciju s objektima, pri čemu se promjenom stanja sustava bilježi da je određena informacija pronađena. Sustav osigurava da se isti trag ne može ponovno „otkriti”, čime se održava dosljednost narativa i sprječava redundantno ponavljanje informacija.

Kitchen

The kitchen looks disturbed. Edward Blackwood lies motionless on the floor, a knife stuck in his back. A cook is also present in the room, visibly shaken by the scene. You notice a refrigerator, an oven, a kitchen table, spice cabinets and a knife rack on the wall.

You can see a Cook here.



Slika 2: Prostorija kuhinje s ključnim tragom tijekom istrage

4.4. Interakcija s neigračkim likovima tijekom istrage

Nakon početnog istraživanja prostora korisnik započinje interakciju s neigračkim likovima. Razgovori i ispitivanja omogućuju prikupljanje dodatnih informacija koje nisu dostupne isključivo kroz pregled objekata. Odgovori likova ovise o trenutačnom stanju sustava, čime se postiže dinamična i kontekstualno osjetljiva komunikacija.

Osim razgovora, korisnik može poduzeti i izravne radnje nad likovima, poput pretraživanja. Takve radnje mogu dovesti do pronalaska ključnih dokaza, ali i do negativnih posljedica ako se koriste nepromišljeno. Time se dodatno naglašava važnost donošenja odluka temeljenih na prikupljenim informacijama.

4.5. Korištenje bilješki i analiza prikupljenih informacija

Kako istraža napreduje, količina prikupljenih informacija raste. Korisniku je u svakom trenutku dostupna naredba za prikaz bilješki, koje sažimaju do sada otkrivene tragove i zapožanja. Sustav bilješki ne uvodi nove informacije, već reflektira trenutačno stanje sustava, čime se izbjegava mogućnost pogrešnog zaključivanja temeljenog na nepotpunim podacima.

Bilješke služe kao pomoć pri donošenju odluka, ali ne zamjenjuju proces zaključivanja. Korisnik mora samostalno povezati informacije iz različitih izvora i procijeniti njihovu važnost u kontekstu istrage.

>notes

CASE NOTES:

- Victim found in the kitchen, stabbed with a kitchen knife.
- Blood was found in the bathroom, away from the crime scene.
- The maid seems to be hiding something.

>

Slika 3: Primjer prikaza bilješki tijekom istrage

4.6. Donošenje odluke i završetak igre

Završna faza igre započinje kada korisnik procijeni da je prikupio dovoljno informacija za donošenje odluke. Formalnim optuživanjem jednog od likova aktivira se mehanizam evaluacije koji, na temelju stanja sustava, određuje ishod igre. Sustav provjerava jesu li ispunjeni preuvjeti za takvu odluku i jesu li ključni tragovi pravilno interpretirani.

Ovisno o točnosti odluke i potpunosti prikupljenih dokaza, igra završava jednim od nekoliko mogućih završetaka. Time se naglašava povezanost između korisničkih postupaka i njihovih posljedica, što je jedno od temeljnih obilježja interaktivnih sustava.

4.7. Zaključak prikaza rada

Prikaz rada igre pokazuje kako se deklarativnim modeliranjem pravila i stanja može ostvariti složen i nelinearan tijek igre bez eksplicitnog upravljanja tokom izvođenja. Korisnik ima slobodu istraživanja i donošenja odluka, dok sustav osigurava konzistentnost i logičku povezanost svih događaja.

>accuse maid

The maid stiffens. "You're wrong," she says coldly. "I have nothing more to say."

>arrest maid

As you move to arrest her, panic flashes in the maid's eyes.

A hidden knife appears in a blur of motion.

*** CASE FAILED

You underestimated her.

The investigation ends in blood. ***

Would you like to RESTART, RESTORE a saved game, QUIT or UNDO the last command?

>|

Slika 4: Primjer završetka igre nakon donošenja konačne odluke

5. Kritički osvrt

Razvijena aplikacija demonstrira mogućnosti primjene deklarativne paradigme programiranja u razvoju interaktivnih sustava, osobito u domeni tekstualnih igara. Korištenjem jezika Inform 7 omogućeno je modeliranje složenog ponašanja sustava kroz skup pravila i stanja, bez potrebe za eksplicitnim upravljanjem tokom izvođenja programa. Takav pristup pokazao se pogodnim za implementaciju nelinearne istrage u kojoj korisnik slobodno istražuje prostor i donosi odluke.

5.1. Prednosti primjenjenog pristupa

Jedna od glavnih prednosti korištenog rješenja jest visoka razina čitljivosti i razumljivosti programskog koda. Pravila i stanja u jeziku Inform 7 izražena su u obliku prirodnog jezika, što olakšava povezivanje implementacije s domenskom logikom igre. Time se smanjuje opterećenje prilikom razvoja i održavanja sustava.

Deklarativni pristup također omogućuje jednostavno proširivanje igre. Novi objekti, likovi ili događaji mogu se dodavati definiranjem dodatnih pravila i stanja, bez potrebe za izmjenom postojećeg toka izvođenja. Takva modularnost pokazala se osobito korisnom u implementaciji dijaloga, bilješki i alternativnih završetaka.

Dodatna prednost sustava jest dosljedno upravljanje stanjem. Korištenjem jedinstvenog mehanizma logičkih varijabli za objekte, likove i završetke, ostvarena je jasna veza između korisničkih radnji i posljedica unutar igre. Time se postiže predvidljivo i logički konzistentno ponašanje sustava.

5.2. Ograničenja i izazovi

Unatoč navedenim prednostima, razvijeno rješenje ima i određena ograničenja. Jedan od glavnih izazova odnosi se na skalabilnost sustava. Kako broj pravila i stanja raste, održavanje preglednosti koda postaje složenije, osobito bez dodatnih mehanizama za grupiranje i hijerarhijsku organizaciju pravila.

Također, iako Inform 7 omogućuje vrlo prirodan opis logike, ta prednost može predstavljati i izazov pri rješavanju složenijih problema. Dijagnosticiranje pogrešaka i praćenje aktivacije pravila ponekad zahtijeva detaljno razumijevanje unutarnjeg mehanizma izvršnog okruženja, što može otežati razvoj složenijih sustava.

Ograničenje se može uočiti i u pogledu korisničkog sučelja. Tekstualna interakcija, iako pogodna za prikaz logike i stanja, može biti manje pristupačna korisnicima koji nisu navikli na ovakav oblik komunikacije s aplikacijom. To ograničenje djelomično je ublaženo integracijom multimedijiskih elemenata, no ostaje prisutno kao karakteristika same domene.

5.3. Moguća poboljšanja i budući razvoj

Moguća poboljšanja razvijenog sustava uključuju dodatnu modularizaciju pravila i stanja, primjerice kroz tematsko grupiranje logike vezane uz pojedine likove ili faze istrage. Time bi se poboljšala preglednost koda i olakšalo održavanje većih projekata.

Daljnji razvoj aplikacije mogao bi uključivati složenije modele ponašanja neigračkih likova, poput adaptivnih reakcija ili dinamičke promjene dijaloga temeljem duljeg niza prethodnih interakcija. Također, moguće je dodatno proširiti sustav dodatnim završecima ili alternativnim narativnim granama, čime bi se povećala ponovna igrivost igre..

6. Zaključak

U okviru ovog projekta razvijena je interaktivna tekstualna igra koja demonstrira primjenu deklarativne paradigme programiranja u modeliranju složenog, nelinearnog sustava. Korištenjem programskog jezika Inform 7 omogućeno je izražavanje logike aplikacije kroz skup pravila i stanja, čime je naglasak stavljen na opis ponašanja sustava, a ne na eksplizitno upravljanje tokom izvođenja.

Razvijena aplikacija uspješno objedinjuje modeliranje prostora, objekata i neigračkih likova s mehanizmima interakcije, odlučivanja i evaluacije ishoda. Posebna pažnja posvećena je implementaciji stanja sustava koja omogućuju dosljedno praćenje napretka korisnika, uvjetovanje dijaloga i određivanje završetaka igre. Time je ostvaren sustav u kojem su korisničke odluke izravno povezane s posljedicama unutar virtualnog svijeta.

Kroz provedenu implementaciju potvrđena je prikladnost deklarativnog pristupa za razvoj interaktivnih sustava temeljenih na pravilima i stanju. Deklarativno modeliranje pokazalo se posebno pogodnim za domenu interaktivne fikcije, gdje se ponašanje sustava prirodno izražava kroz odnose, uvjete i promjene stanja. Iako takav pristup ima određena ograničenja u pogledu skalabilnosti i složenosti, njegova čitljivost i fleksibilnost čine ga vrijednim alatom za razvoj ovakvih aplikacija.

Zaključno, projekt je ostvario postavljene ciljeve te pruža kvalitetnu osnovu za daljnje proširenje i nadogradnju sustava. Dobivena igra ukazuju na potencijal deklarativnog programiranja kao učinkovite paradigme za razvoj interaktivnih i logički složenih aplikacija.

Popis literature

- [1] J. W. Lloyd, „Practical advantages of declarative programming,” *Joint Conference on Declarative Programming*, 1994., str. 18–30.
- [2] I. Bratko, *Prolog Programming for Artificial Intelligence*, 1. izdanje. Harlow, UK: Addison-Wesley, 1986., 696 str., ISBN: 978-0-321-41746-6.
- [3] Stanford Encyclopedia of Philosophy. „First-order logic model theory,” Stanford University, pogledano 27. prosinca 2025. adresa: <https://plato.stanford.edu/entries/modeltheory-fo/>.
- [4] G. Nelson. „The inform designer’s manual,” Inform 7, pogledano 27. prosinca 2025. adresa: <https://www.inform-fiction.org/manual/DM4.pdf>.

Popis slika

1.	Početni prikaz ulaznog hodnika pri pokretanju igre	13
2.	Prostorija kuhinje s ključnim tragom tijekom istrage	14
3.	Primjer prikaza bilješki tijekom istrage	15
4.	Primjer završetka igre nakon donošenja konačne odluke	16

Popis isječaka koda

1.	Deklarativno modeliranje prostora i odnosa između prostorija	4
2.	Definicija prostorije s pripadajućim opisom	5
3.	Primjer scenografskih elemenata u prostoriji	5
4.	Definicija spremnika i skrivenog elementa	5
5.	Definicija neigračkih likova i njihove početne pozicije	6
6.	Definicija akcije za razgovor s NPC-ovima	6
7.	Primjer uvjetovanog dijaloga temeljenog na stanju sustava	7
8.	Logička stanja povezana s objektima i tragovima	8
9.	Promjena stanja objekta nakon interakcije	8
10.	Logička stanja povezana s ponašanjem neigračkih likova	8
11.	Promjena stanja neigračkog lika tijekom dijaloga	8
12.	Provjera preduvjeta prije donošenja odluke o optuživanju	9
13.	Uvjetovani prikaz bilješki na temelju stanja sustava	10
14.	Određivanje završetka igre na temelju stanja sustava	10