**Opis frameworka**

Kao što je najavljeno, ove ćete godine tehnike multiagentskih inteligentnih sustava isprobavati na novom frameworku. Igra je ista - Pacman.

Framework je pisan u Javi, a kao osnova je korišten JMonkeyEngine (<http://jmonkeyengine.org/>), open source game engine pod BSD licencom za koji je dostupan open source development SDK temeljen na NetBeans IDE-u <http://hub.jmonkeyengine.org/downloads/>.

Pacman framework sadržavat će sve potrebno da se igra izvodi osim inteligencije samih agenata koji sudjeluju u igri - pacmana i duhova. Vaš zadatak je reimplementirati klasu **AgentAI** iz frameworka odnosno njezinu metodu **decideMove** koja, na temelju dostupnih poteza i okoline koje vam za odabranog agenta u promatranom trenutku daje framework, odlučuje o sljedećem potezu odnosno smjeru kretanja agenta. Vašu će implementaciju biti moguće dodati u sam framework i agenti će se onda ponašati kako vaš algoritam specificira.

Klasu AgentAI treba implementirati za svaku od klasa agenata, dakle jedna implementacija mora biti za agenta pacman, a druga za duhove. Sam framework će imati defaultnu implementaciju za obje klase agenata, a vi morate reimplementirati **obje**.

Vaš(i) agent(i) **će** u svakom trenutku od frameworka **dobiti**:

* Informaciju o tome što odnosno tko se nalazi u njegovoj neposrednoj okolini (duhovi, pacman, zid, kolačići ili powerup) i koja su mu svojstva. Veličina dostupne okoline nije pod vašom kontrolom i podložna je promjeni, ali dostupna kao informacija.
* Informaciju o metapodacima koje je sam prethodno upisao u svoju okolinu (za pacmana) odnosno informacije koje su u njegovu okolinu upisali on i njegovi kompanjoni (za duhove, postoji mogućnost koordinacije među njima). O obliku metainformacija odlučuje sam agent.
* Informaciju o svojim svojstvima (i duhovima i samom pacmanu važno je znati je li pacman pod utjecajem powerupa odnosno je li upravo bio pojeden).

Vaš(i) agent(i) **neće dobiti**:

* Informaciju o svojoj apsolutnoj lokaciji u svijetu.
* Informaciju o veličini svijeta odnosno njegovom izgledu.
* Informaciju o broju odnosno poziciji duhova u svijetu.
* Informaciju o broju kolačića u svijetu.
* Informaciju o broju odnosno poziciji i trajanju powerupova u svijetu.

Sve od ovih varijabli koje nećete imati također su podložne promjeni koja nije pod kontrolom vaših agenata.

Još neke propozicije:

* **Pacman pobjeđuje kada pojede sve kolačiće, a gubi ako njega prije nego što uspije u tome pojedu duhovi i to onoliko puta koliko pacman ima života.**
* Duhovi pobjeđuju kada pacman izgubi.
* Pacmanov svijet je 2D svijet (smješten u 3D prostor).
* Duhovi mogu pojesti pacmana uvijek kada nije pod utjecajem powerupa.
* Kada pacman pojede powerup onda može jesti duhove neko vrijeme (bit će uočljivo vizualno, a i kroz pacmanova svojstva).
* Kada pacman ili duhovi budu pojedeni vraćaju se na svoju početnu poziciju (resetiraju se).
* Pacman ima poseban izgled (i svojstvo) neposredno nakon što izgubi život (trenutno nije zaštićen od toga da bude pojeden).
* Agenti igraju prema prioritetu - pacman ima najveći prioritet i igra prvi, a duhovi igraju nakon njega.
* Pacman se na početku igre stvara na slučajnoj lokaciji (osim u kućici duhova, bilo bi nezgodno).
* Duhovi se uvijek stvaraju u kućici duhova koja je u centru svijeta i obično je unutra gužva.

Vaš model inteligencije treba se pobrinuti da vaš agent na temelju onog što dobije iz okoline izgradi model svijeta koji će vam onogućiti da efikasno odlučuje i pobijedi u igri.

**Najnovija verzija frameworka (i sve prije nje) nalaze se u repozitoriju sa strane.** **Za sva pitanja, prijedloge i bugove možete mi se javiti na mail, ili na forum koji je otvoren ispod i kojeg ću isto pratiti (forum je bolja opcija da svi mogu pratiti). Pratite obavjesti i repozitorij na ovoj stranici!**

**Tijek igre**

**Pokretanje**

Za pokretanje igre trebate Java 7 JRE <http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>. Igra je zapakirana u .rar arhivu i sadrži datoteku Pacman.jar koja predstavlja samu igru, lib folder gdje se nalaze jar arhive potrebne za rad jMonkeyEngine-a te javadoc folder gdje se nalazi dokumentacija za klase koje implementiraju igru (za sada su dokumentirane samo klase koje ćete izravno koristiti pri razvoju, navedene su u nastavku).

Arhivu rasparkirajte u folder po želji. Igru pokrećite dvoklikom na Pacman.jar odnosno iz komandne linije pozivom **java -jar Pacman.jar** ako se nalazite u folderu u kojem je Pacman.jar.

Na naslovnom ekranu engine-a odaberite željenu rezoluciju te pritisnite **continue**.

**Glavni meni**

Na naslovnom ekranu igre postoje sljedeća polja koja određuju koju će klasu odnosno klase pacman i agenti koristiti za svoju inteligenciju:

* **classPath** - putanja unutar koje se nalazi hijerarhija foldera i fileova koji predstavljaju java pakete i klase koje sadrže inteligenciju za agente.
* **PacmanAIClass** - Ime klase koju pacman koristi kao inteligenciju (uključuje ime paketa).
* **GhostAIClass** - Ime klase koju agenti koriste kao inteligenciju (uključuje ime paketa).

**Primjer**: Ako se Pacman.jar nalazi u D:\Projekti\MonkeyEngine\GameAIFramework\dist\ onda u tom folderu možemo napraviti folder AITest i u njega staviti file DummyAI.class. DummyAI.class sadrži klasu DummyAI čiji je paket AITest (kojeg predstavljamo folderom AITest), a koja reimplementira klasu AgentAI. Da bismo koristili ovu klasu (i za pacmana i za ghostove) u classPath polje na glavnom ekranu upisujemo "**file:**D:\Projekti\MonkeyEngine\GameAIFramework\dist**\"** (protokol "file:" i posljednji slash po kojem se zna da se radi o folderu su obavezni!), a u PacmanAIClass i GhostAIClass upisujemo ime klase zajedno sa punom putanjom paketa u kojem se nalazi dakle "AITest.DummyAI".

Ako polja za inteligenciju ostavite praznima pacman će koristiti klasu PacmanAI, a ghostovi klasu GhostAI. Obje navedene klase se nalaze u repozitoriju kao primjeri koji vam mogu pomoći da izgradite svoju inteligenciju. Ako se desi pogreška kod učitavanja klase za inteligenciju umjesto nje će se koristiti AgentAI klasa koja nasumice pomiče agente.

Nakon što po želji podesite opcije za inteligenciju (koje se pamte za poslije u konfiguracijskom fileu) pritisnite **Start** za pokretanje igre ili **Quit** za izlazak.

**Tijekom igre**

Pogled na svijet unutar igre možete kontrolirati **mišem ili strelicama (rotacija) odnosno tipkama W, A, S, D (translacija)**. Igru možete (od)pauzirati pritiskom na tipku **P (pauza)**.

Iz pokrenute igre se izlazi pritiskom na tipku **Escape (Esc)**.

Tijekom igre postoji korisničko sučelje (overlay) u kojemu se ispisuju poruke o tome što se događa u igri. Svaka poruka ima redni broj pri čemu je poruka s većim rednim brojem nastala kasnije tijekom igre. Tipične poruke su:

* poruke o tome koje inteligencije su učitane i korištene (i sve greške do kojih je pri tome došlo).
* poruke o tome koliko života ima Pacman.
* **poruke koje je klasa inteligencije poslala sučelju preko metode printStatus klase AgentAI** korištene unutar metode decideMove. Sami odabirete kada poslati ovakvu poruku kao i njezin sadržaj.
* **poruke o greškama koje su se dogodile prilikom izvođenja metode decideMove klase koja implementira inteligenciju. Osim poruke o grešci svaka pogreška uz sebe ima i informaciju o liniji u kojoj se pogreška dogodila kao i potpunu informaciju o stanju na stogu aplikacije u tom trenutku.** Koristite ove informacije za debuggiranje zajedno sa ispisivanjem korisnih poruka preko metode **printStatus**.

Igra završava nakon što pacman izgubi sve živote (odnosno njihov broj bude jednak 0). Nakon toga otvara se ekran na kojem se ispisuje rezultat igre - pacmanova pobjeda odnosno poraz uz naveden broj bodova koje je pacman uspio pojesti. Sa tog se ekrana moguće vratiti u glavni menu.

**Razvoj inteligencije**

**Razvojne okoline**

Za razvoj/prevođenje trebate Java 7 JDK <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html> te možete koristiti **Eclipse** odnosno **Netbeans** razvojne okoline, obje su besplatne i dostupne za download na <https://www.eclipse.org/downloads/> (standard opcija) i <https://netbeans.org/downloads/>.

Debugging se može izvoditi unutar samog frameworka - pomoću metode **printStatus** iz **AgentAI** klase što je ekvivalentno debuggiranju pomoću printf-ova u C-u. Učinkovito, ali ne baš glamurozno ovaj čas.

Poželjan način za pisanje odnosno prevođenje koda u **Netbeans** okolini je sljedeći:

* Otvorite novi projekt tipa Java Class Library: File->New Project -> (Categories) Java -> (Projects) Java Class Library. Smjestite ga na proizvoljnu lokaciju i imenujte po želji.
* Dodajte Pacman.jar u listu dostupnih library-a.: Project explorer (lijevi dio ekrana) -> Ime projekta -> Libraries -> (lijevi klik) Add JAR/Folder... -> Nađite Pacman.jar i odaberite Open. Pacman.jar će se pojaviti na popisu library-a.
* Dodajte u projekt novu klasu koja extenda mmaracic.gameaiframework.AgentAI klasu i reimplementirajte metodu decideMove (prototip mora biti identičan onom u AgentAI klasi).

Poželjan način za pisanje odnosno prevođenje koda u **Eclipse** okolini je sljedeći:

* Otvorite novi projekt: File->New-> Java Project.
* Na inicijalnom prozoru za kreiranje projekta postavite željeno ime. Na drugom prozoru "Settings" odaberite "Libraries" tab pa zatim gumb "Add External JARs" te nađite i dodajte Pacman.jar. Jar bi se sada trebao pojaviti na popisu pa stisnite gumb "Finish" nakon čega se otvara GUI projekta.
* Dodajte novu klasu File->New->Class. Na inicijalnom prozoru postavite joj ime i paket te stisnite "Browse" gumb pored opcije Superclass. U prozoru koji se zatim otvori upišite "AgentAI" u textbox za pretragu te na popisu ispod odaberite klasu mmaracic.gameaiframework.AgentAI i kliknite "Ok". Na prozoru za kreiranje klase kliknite "Finish".
* Otvorio vam se prozor za vašu klasu i u nju morate dodati reimplementaciju metode decideMove (prototip mora biti identičan onom u AgentAI klasi).

**Naputci za razvoj**

U vašoj implementaciji klase AgentAI možete klasi dodavati svojstva (polja, properties) koja će vam služiti da izgradite strukture podataka o svijetu koje vam trebaju za odlučivanje tijekom postojanja agenta. Isto tako možete dodavati dodatne metode kako bi vam kod bio grupiran u smislene cjeline, ali uzmite u obzir da će framework pozivati samo metodu **decideMove** i ona mora pozivati sve korisne metode koje služe odlučivanju.

Više o klasi **AgentAI** te klasama **PacmanVisibleWorld** (opisuje okolinu agenta i objekte u njoj) i **WorldEntity.WorldEntityInfo** (opisuje svojstva objekata u svijetu) koji su ulaz u metodu decideMove pogledajte u dokumentaciji priloženoj u arhivi frameworka.

Svojstvo agenta pacman koje će vas zanimati prilikom odlučivanja je svojstvo imena **PowerUp** koje će imati vrijednost **"True"** kad je pacman pod utjecajem powerupa odnosno **"False" ili neće postojati** kad nije. Pacman će imati svojstvo **Death** neko vrijeme neposredno nakon što mu duhovi oduzmu život. Imena svojstava su stringovi i dostupna su vam kao konstante klase **PacmanAgent**. Ime im je oblika *svojstvo*PropertyName, a za PowerUp ime konstante je powerupPropertyName i vrijednost joj je "PowerUp".

Kako bi identificirali koga odnosno što predstavlja klasa **WorldEntity.WorldEntityInfo** sa nekog elementa susjedstva trebate koristiti njezinu funkciju **getIdentifier()** koja vraća string koji u igri pacman može imati vrijednosti: **"Pacman"**, **"Ghost"**, **"Point"**, **"Powerup"** i **"Wall"**.

Za debugging unutar svoje AI klase koristite metodu **printStatus** klase AgentAI koja će String koji joj se preda ispisati u oevrlayu koji se prikazuje u igri.

Nakon što se vaša klasa prevede i generira se .class file taj je file moguće koristiti u igri podešavanjem opcija pri pokretanju igre.

**State of the game**

**Problemi i nedostaci:**

* Loša grafika (nisam umjetnik, svaka pomoć je dobrodošla :) )
* Dodati vremenska ograničenja.
* Dodati višedretvenost.

**Ciljevi:**

Osim popraviti navedeno cilj je, ako bude moguće do kraja semestra koristiti ovu aplikaciju kao applet te izgraditi serversku stranu koja će čuvati vaše AI implementacije i kao i statistiku koju su postigle jedne protiv drugih.