Eliksiry
Zadanie zaliczeniowe
z Programowania Obiektowego 2OO3/2OO4
wersja 1.7, poprawiona 6 stycznia 2OO3

Michał J. Gajda
mgajda@mimuw.edu.pl

Warszawa, 5 stycznia 2004

Gra symuluje lekcję Eliksirów udzielaną w szkole czarodziejów przez profesora Snape'a uczniom na roku Harry'ego Pottera. Profesor Snape dzielił uczniów na grupy i zlecał im sporządzenie odpowiedniego eliksiru. Na lekcji byli obecni studenci z dwóch różnych domów (akademików), między którymi istniał silny duch konkurencji. Zadaniem gracza jest pokierowanie parą uczniów przygotowujących swój eliksir i przeszkadzających w stworzeniu eliksiru uczniowi z domu konkurencyjnego. Przeszkadzanie może polegać na blokowaniu dojścia do składników eliksiru lub podrzucaniu niewłaściwych składników.

1 Plansza

Plansza składa się z kwadratowych pól o boku długości jednego metra. Dwa pola nazywamy sąsiadującymi ze sobą, kiedy stykają się bokami lub rogami. Dla uproszczenia odległość między dwoma polami możemy liczyć jako maksimum z różnicy współrzędnych.

Występują poniższe rodzaje pól:

- pusta przestrzeń,
- mur,
- pojemnik ze składnikiem.

2 Czarodziejki i czarodzieje występujący w grze

Każdy z czarodziejów (tzn. uczniów i profesora Snape'a) występujących w grze zajmuje dokładnie jedno pole (tzn. nie może być dwóch na jednym polu w tym samym momencie). Poruszają się kolejno, przechodząc przez co najwyżej n sąsiadujących pól nie zajętych przez innych. (Parametr n jest określony w pliku konfiguracyjnym.) Czarodzieje nie mogą przechodzić przez ściany, ani pojemniki ze składnikami.

2.1 Uczniowie

Uczniowie profesora Snape'a mają kociołki, w których jest tylko jeden rodzaj składnika lub mieszanki naraz. Każdy uczeń widzi zawartość swojego kociołka i kociołków uczniów sąsiadujących.

Każdy uczeń należy do jednego z dwóch domów: Gryffindoru i Slytherinu. Studenci należący do tego samego domu są przyjaźni wobec siebie: jeżeli ktoś z tego samego domu próbuje wejść na pole zajmowane przez studenta, to studenci zamieniają się miejscami.

Uczniowie po ruszeniu się mogą w każdej turze wykonać jedną z następujących akcji:

- dodać do kociołka składnik, przy którego pojemniku stoją,
- wrzucić lub wlać całą zawartość własnego kociołka do kociołka sąsiada,
- pokazać zawartość kociołka profesorowi do sprawdzenia (Snape musi być na polu sąsiadującym).

Zamiast poruszania się w danej turze uczniowie mogą zakrywać swój kociołek. Nie można nic wrzucić do zakrywanego kociołka, ani zobaczyć jego zawartości. Efekt zakrywania trwa do momentu, gdy uczeń się ruszy lub wykona jakąś inną akcję.

2.2 Zwykli uczniowie

Dla uproszczenia nie symulujemy tworzenia eliksiru przez większość uczniów, a tylko przez wyróżnionych.

Zwykli uczniowie nie mają w grze żadnego celu; ich strategią może być losowe poruszanie się po planszy. Nie próbują też przepychać (zamieniać się miejscami) z innymi uczniami, ani z nauczycielem. Nie próbują też oddawać wyników swojej „pracy" nauczycielowi.

2.3 Malfoy i jego kumple

Trzech uczniów należących do Slytherinu: Malfoy, Crabbe i Goyle, konkuruje z parą Harry'ego. Malfoy próbuje przy użyciu dowolnej strategii (według uznania implementatora) stworzyć eliksir, a Crabbe i Goyle aktywnie szukają destruktywnych składników i próbują je wrzucić do kociołków przenoszonych przez parę Harry'ego. Składnik uważa się za destruktywny, jeśli jego wrzucenie na jednym z etapów pracy uniemożliwi stworzenie eliksiru i/lub uszkodzi studenta.

2.4 Para Harry'ego

Gracz w dowolnie wybrany przez siebie sposób steruje parą uczniów należących do Gryffindoru: Hermioną i Harrym Potterem.

2.5 Profesor Snape

Profesor Snape stara się pilnować, żeby uczniowie uczciwie pracowali nad eliksirem, zamiast sobie przeszkadzać. Porusza się losowo, a kierunek w którym spogląda jest kierunkiem w którym się poruszał, wchodząc na ostatnie pole w danej turze. Profesor obserwuje wszystkich niezasłoniętych przez ściany studentów pod kątem $\pm 45°$ od kierunku w którym patrzy.

Przykładowo: jeśli profesor jest zwrócony na zachód, to obserwuje wszystkich uczniów pomiędzy przekątną północno-zachodnią i południowo-zachodnią.

Jeżeli profesor zauważy, że Harry lub Hermiona wrzucają coś do cudzego kociołka, to zmusza niegrzecznego studenta do tłumaczenia się przez 3 tury. Przez ten czas niegrzeczny student nie może się poruszyć, ani zakryć kociołka. Profesor Snape faworyzuje uczniów Slytherinu i nie zwraca im uwagi, gdy wrzucają coś do kociołków studentów z Gryffindoru. Nie karci też Hermiony za wlewanie czegoś do kociołka Harry'ego i odwrotnie.

3 Eliksiry

Kolejność łączenia składników ma znaczenie. Wrzucanie czegoś do mieszanki, w której to coś już jest zawarte, nie daje żadnego efektu. (Relacja zawierania jest przechodnia.) Wrzucanie czegoś w kolejności innej, niż opisana w pliku konfiguracyjnym „psuje" eliksir. Dodawanie następnych składników do „zepsutego" eliksiru nadal daje „zepsuty" eliksir. Każdy rodzaj składnika i eliksiru ma opis, umożliwiający rozróżnienie między niektórymi rodzajami. Nie posiadają opisu tylko eliksiry, które natychmiast po powstaniu ulegają zniszczeniu. Gracz może oglądać opis eliksiru pary Harry'ego, jak i opisy eliksirów przenoszonych w odległości jednego pola od pary Harry'ego.

Dla uproszczenia gry nie przejmujemy się proporcjami między poszczególnymi składnikami.

3.1 Składniki

Każdy składnik jest dostępny w wyznaczonym miejscu na planszy w ilości nieograniczonej. Uczeń może przenosić tylko jeden rodzaj składnika lub mieszanki składników naraz.

3.2 Efekty

Po zmieszaniu składnika z eliksirem mogą nastąpić różne ciekawe efekty:

- Wybuch może wysłać wszystkich uczniów w promieniu kilku metrów do skrzydła szpitalnego. Ranieni w ten sposób studenci nie mogą już uczestniczyć w tej lekcji. (Profesor Snape, jako doświadczony czarodziej, potrafi się łatwo obronić przed skutkami wybuchu.)

- Intensywne musowanie eliksiru może rozchlapać eliksir w promieniu kilku metrów. Daje to podobny efekt, jak dodanie tego eliksiru do wszystkich eliksirów w niezakrytych kociołkach na tym obszarze. Twórca traci rozchlapany eliksir. Jeżeli rozchlapywany eliksir trafia do pustego kociołka, to staje się zepsuty. Próba zakrywania musującego eliksiru się nie udaje.

- Eliksir może być szkodliwy dla zdrowia lub nie. Zmienia to zachowanie profesora Snape'a i studenta w czasie oddawania mu eliksiru.

- Eliksir może zacząći reagować, by po n turach przemienić się w inny eliksir. Dla uproszczenia, jeżeli wrzucimy coś do eliksiru przed zakończeniem reakcji, to zachowuje się on jakby wrzucono to coś na początku reakcji.

Składniki w pojemnikach są zabezpieczone przed powyższymi efektami. Zarówno wybuch, jak i rozchlapanie wpływają tylko na uczniów i zawartość kociołków niezasłoniętych przez ścianę.

3.3 Oddawanie eliksiru

Sposobem zwycięskiego zakończenia gry jest przekazanie właściwego eliksiru profesorowi. Podczas oddawania eliksiru mogą się zdarzyć trzy przypadki:

1. Eliksir pokazany profesorowi przez studenta jest właściwym eliksirem. Wtedy gra się kończy zwycięstwem tej strony, której reprezentant dostarczył ten eliksir.

2. Eliksir pokazany profesorowi przez studenta jest niewłaściwym eliksirem, a właściwy eliksir byłby szkodliwy dla zdrowia – wtedy profesor mówi uczniowi, że ten przygotował zły eliksir.

3. Pokazany eliksir jest niewłaściwy, a właściwy eliksir nie jest szkodliwy dla zdrowia – wtedy profesor wymaga od studenta „spróbowania" tego eliksiru. Jeśli podawany eliksir jest szkodliwy (`poisons`), to student spędzi pozostałą część lekcji poza planszą (w skrzydle szpitalnym szkoły).

Gra kończy sie remisem, gdy cała trójka: Malfoy, Hermiona, jak i Harry dostaną się do skrzydła szpitalnego.

4 Plik konfiguracyjny

Plik konfiguracyjny ma składnię kodu w Smalltalku, którego wynikiem ma być obiekt opisujący scenariusz gry. Występujące w nim konstrukcje mają następujące znaczenie:

`Ingredient new:` *nazwa* `mark:` *oznaczenie na planszy* – wprowadza nowy składnik. Nazwa składnika jest widoczna na pojemniku.
x `description:` *opis* – dodaje opis do składnika lub mieszanki. Dwa obiekty o tym samym opisie są nierozróżnialne dla czarodziejów.
a `+` b – mieszanka tworzona przez dodanie składnika b do a.
a `poisons` – zaznacza, że a jest szkodliwy dla zdrowia.
a `explodes:` *zasięg* – mieszanka a powstała w kociołku natychmiast eksploduje, raniąc studentów w promieniu *zasięg* metrów.
a `splash:` *zasięg* – stworzona mieszanka a powstała w kociołku gwałtownie musuje, rozchlapując się w promieniu *zasięg* metrów.
r `:=` a `after:` *liczba tur* – po kilku turach mieszanka a zamienia się w r.
a `wins` – celem scenariusza jest stworzenie eliksiru a.
`GameLevel map:` *mapa* `ingredients:` *tablica* `speed:` *liczba* – tworzy poziom z podaną *mapą*, składnikami podanymi w *tablicy* (aby wiadomo było jak interpretować symbole na mapie) i podaną szybkością poruszania się czarodziejów.

Mapa poziomu jest tablicą Stringów, w których kolejne znaki oznaczają:
`A-Z` i `0-9` – składniki;
`#` – ścianę;
spacja – pustą przestrzeń;
`@` – Harry'ego lub Hermionę;
`!` – profesora Snape'a;
`&` – Malfoya;
`%` – kolegów Malfoya (Crabbe i Goyle);
`+` – zwykłych studentów Gryffindoru;
`*` – zwykłych studentów Slytherinu.

UWAGA:

- konstrukcja `GameLevel map:`... powinna być ostatnią konstrukcją w pliku konfiguracyjnym – dzięki temu zwracana przez nią wartość będzie również wartością pliku konfiguracyjnego, po obliczeniu jej przez Smalltalka;
- mapa poziomu powinna mieć te same wymiary we wszystkich wierszach;
- w przypadku wystąpienia błędu semantycznego w pliku konfiguracyjnym należy wypisać zrozumiały komunikat umożliwiający łatwe znalezienie błędu;
- w przypadku znalezienia błędu syntaktycznego w pliku konfiguracyjnym wystarczą komunikaty Smalltalka.

Poniżej widać przykładowy plik konfiguracyjny:

```
| hair bezoar blood juice
  silvery murky red muddy muddyRed white
  dryIngredients elixir
  ingredients levelMap |
```

```
"Opisy są po angielsku, aby uniknąć konieczności
 dodawania rodzaju gramatycznego i odmiany."
hair   := Ingredient new: 'Unicorn hair'
                     mark: '1'.
bezoar := Ingredient new: 'Bezoar'
                    mark: '2'.
blood  := Ingredient new: 'Dragon blood'
                    mark:'3'.
juice  := Ingredient new:
  'Juice from blueberries gathered at full moon'
                    mark:'4'.

hair    description: 'Silver hair'.
bezoar  description: 'Stone'.
blood   description: 'Red potion'.
juice description: 'Purple potion'.

silvery := juice + hair.
silvery description: 'Silvery potion'.
silvery poisons.

blood + hair explodes: 3.

murky   := blood + bezoar.
murky   description: 'Murky red potion'.

red     := murky after: 6.
red     description: 'Red potion'.

muddyRed := red + juice.
muddyRed description: 'Muddy red potion'.
muddyRed + hair splash: 1.

muddy   := murky + juice.
muddy   description: 'Muddy white potion'.

white   := muddy after: 2.
white   description: 'White potion'.

muddy + hair splash: 2.

dryIngredients := hair + bezoar.
dryIngredients description: 'Stone and hair'.

(Array with: blood with: juice with: murky) do:
[ :x | (hair + x) splash: 1 ].
```

```
(Array with: red with: silvery with: murky with: muddy) do:
[ :x | (dryIngredients + x) splash: 1.
     (x + dryIngredients) explodes: 3  ].

elixir  := white + hair.
elixir  description: 'Silvery potion'.
elixir  wins.

levelMap := Array new.
levelMap add: '###############'.
levelMap add: '#      1      #'.
levelMap add: '#             #'.
levelMap add: '#             #'.
levelMap add: '# #    ####   #'.
levelMap add: '# # +++@@++++ #'.
levelMap add: '# # ++     ++ #'.
levelMap add: '#4# ++     ++2#'.
levelMap add: '# # ** !   ** #'.
levelMap add: '# # **     ** #'.
levelMap add: '# # ***%&%*** #'.
levelMap add: '#             #'.
levelMap add: '#             #'.
levelMap add: '#      3      #'.
levelMap add: '###############'.

GameLevel map: levelMap
          ingredients: (Array with: blood with: juice with: hair with: bezoar)
          speed: 6
```

5 Kryteria oceniania

Wykonanie zadania zaliczeniowego będzie oceniane według następujących kryteriów:

- jasności i precyzyjności projektu programu,
- poprawności programu,
- metodologicznie (obiektowo) poprawnej architektury programu,
- przejrzystości i efektywności kodu programu,
- zgodności ze specyfikacją,
- zgłoszonych sprzeczności i niejasności w specyfikacji,
- ergonomii interfejsu użytkownika.

Następujące kryteria nie powinny mieć wpływu na ocenę zadania:

- zgodne ze specyfikacją rozszerzenia nieuzgodnione z prowadzącym,
- uroda interfejsu użytkownika.

Zakłada się, że jeśli specyfikacja jest niejasna lub sprzeczna i implementator nie odkryje tego, to jest to zaniedbanie implementatora (jako osoby zobowiązującej się do wykonania specyfikacji).

Wątpliwości będą rozwiązywane zgodnie z zasadą wzajemnej życzliwości obu stron (przy czym niezbywalnym celem prowadzących jest zarówno prawidłowo działający program, jak i poprawnie wyedukowani studenci).