**Student Life Simulator**

Zbliża się czas sesji. To trudny czas dla studentów, jednak nie wszyscy pilnie się uczą. Niektórzy zdecydowali udać się na wyspę wspólnie raczyć się P.I.W.E.M. Zabawa się rozkręca, rozweseleni studenci zaczynają zapraszać swoich znajomych, a przecież takie zaproszenie nie sposób zignorować. Czy wszystkim studentom uda się zaliczyć studia?

Symulacja zakłada utworzenie planszy z dwoma, losowo umieszczonymi agentami – Studentami i Egzaminatorami. Ilość agentów jest zdefiniowana na początku symulacji (w zakresie od 1 do rozmiar\_planszy \* 2 - 1). Symulacja zaczyna się z określoną liczbą studentów po piwie (też w zakresie od 1 do rozmiar\_planszy \* 2 - 1). Egzaminatorzy przemieszczają się wśród studentów i przeprowadzają napotkanym egzaminy, ci, którzy są po piwie mają mniejsze szanse na zaliczenie. (trzeźwi zawsze zaliczają) Ma na to wpływ ilość wypitych przez studenta piw oraz podejrzliwość egzaminatora, która określa jak dobrze rozpoznaje on studentów po piwie (albo na ile chce to zauważyć). Student musi zdać cztery egzaminy, żeby dostać się na kolejny semestr, nie ma też wygórowanych ambicji – wystarczający jest dla niego status inżyniera (7 semestrów). Niestety, jeśli 4 razy nie zaliczy egzaminu to zostaje natychmiast skreślony z roku. Student zostając skreślonym bądź otrzymując tytuł inżyniera znika z planszy. Student po piwie spotykając innego studenta proponuje mu wspólne napicie się, przed tą pokusą chroni go chęć do nauki – parametr określający prawdopodobieństwo, że do napicie się nie dojdzie. W przypadku, kiedy obaj studenci są po piwie decyduje mniejsza wartość, tzn. studenci z większym prawdopodobieństwem wypiją piwo. To dziwne, ale nie każdy ma zawsze taką silną głowę. Każdego studenta charakteryzuje parametr odporności na alkohol, jeżeli go przekroczy, to zasypia w miejscu, w którym stoi. Egzaminator spotykając takiego osobnika wpada w szał i wystawia mu od razu ocenę niedostateczną. Po czterech turach student budzi się trzeźwy, dociera też do niego, że faktycznie czas zabrać się do nauki i jego chęć do nauki zwiększa się. Co 4 tury studenci trzeźwieją o jedno piwo.

Symulacja kończy się, kiedy wszyscy studenci znikną z planszy. Można wtedy przedstawić ilość studentów, którzy zdali i w jakim czasie oraz tych, którzy zostali skreśleni i na jakim semestrze/czasie. Można utworzyć wykres, który będzie przedstawiał w czasie ilość studentów po piwie, studentów trzeźwych, studentów śpiących; oblanych egzaminów, zdanych egzaminów w danej turze.

**Agenci:**

1. **Student:**

Rodzi się z dwoma losowo dobranymi parametrami: odpornością na alkohol oraz chęcią do nauki. Odporność na alkohol określa, ile piw student jest w stanie wypić zanim zaśnie w miejscu, w którym stoi. Student, który zasnął, budzi się trzeźwy po czterech turach. Po przebudzeniu się zdaje sobie sprawę, że faktycznie czas zabrać się do nauki, jego chęć do nauki zwiększa się. Chęć do nauki określa prawdopodobieństwo z jakim odmówi koledze wypicia z nim piwa (bo zdarza się, że niektórzy studenci się czasem uczą). Co 4 tury student trzeźwieje o jedno piwo.

1. **Egzaminator:**

Egzaminator rodzi się z jednym parametrem – podejrzliwością, określa ona zdolność egzaminatora do wyczucia zamglonego studenta. Im wyższy poziom podejrzliwości tym większe prawdopodobieństwo, że egzaminator nie zaliczy studenta po piwie.

**Interakcje między agentami:**

1. **Student - egzaminator:**

Egzaminator spotykając się ze studentem natychmiast przeprowadza mu egzamin chcąc sprawdzić poziom jego wiedzy. Studenci trzeźwi zawsze zaliczają egzamin, niestety, ci po piwie nie zawsze mają zawsze tyle szczęścia. Szansa, że pijany student nie zda egzaminu wynosi ilości wypitych piw oraz podejrzliwości egzaminatora.

1. **Śpiący\_student - egzaminator:**

Egzaminator spotykając śpiącego studenta bardzo się denerwuje i od razu wystawia studentowi ocenę niedostateczną.

1. **Student\_po\_piwie – student / Student\_po\_piwie - Student\_po\_piwie:**

Student po piwie spotykając swojego pobratymca proponuje mu wspólne wypicie kolejnego piwa. Nie zawsze się mu to udaje, decyduje o tym chęć do nauki. Jeżeli oboje są po piwie, wtedy decyduje mniejszy parametr chęci do nauki (jest większe prawdopodobieństwo, że wypiją piwo).

1. **Egzaminator - egzaminator:**

Spotykając się mogą jedynie ponarzekać na studentów i powspominać stare, dobre czasy. Nic się nie dzieje.

**Analiza czasownikowo-rzeczownikowa**

Projektujemy symulacje agentową, w której występują dwa rodzaje osobników, studenci i egzaminatorzy. Badamy ilu studentów odpadnie ze studiów w danym semestrze oraz ilu zaliczy studia, tj. zda 7 semestrów. Symulacja odbywa się na dwuwymiarowej planszy podzielonej na pola.

Zachowanie się osobników:

* każdy z osobników co turę wykonuje ruch o jedno pole w losowo wybranym kierunku
* studenci po piwie spotykając innych studentów proponują im piwo, wskaźnik chęć\_do\_nauki określa z jakim prawdopodobieństwem studenci nie wypiją piwa, jeżeli student przekroczy swoją odporność na alkohol to zasypia w miejscu, w którym stoi
* egzaminator spotykając studenta przeprowadza mu egzamin, im więcej wypił on piw i im wyższa podejrzliwość egzaminatora tym mniejsza szansa, że zaliczy on egzamin, egzaminator spotykając śpiącego studenta wystawia mu ocenę niedostateczną
* student, który nie zdał 4 razy zostaje wyrzucony - znika z planszy, student, który zdał 4 razy otrzymuje promocje na następny semestr, kończąc 7 semestr znika z planszy

Parametry symulacji:

* rozmiar planszy (od 10 do 100; przeskok co 10)
* liczba agentów, w tym początkowa liczba studentów po piwie (od 1 do rozmiar\_planszy \* 2 - 1; przeskok co 1)
* średnia podejrzliwość egzaminatorów, wytrzymałość na alkohol studentów oraz ich chęć do nauki (wybór zakresu generowania od 1 do rozmiar\_planszy \* 2 - 1)

**Karty CRC**

1. Simulation
   1. Superclass: -
   2. Subclass: -
   3. Responsibilities:
      1. Initializes the simulation
      2. Runs the simulation
      3. Exports data
   4. Collaboration:
      1. Agent
      2. Board
      3. BoardStatus
2. Board
   1. Superclass: -
   2. Subclass: -
   3. Responsibilities:
      1. Manages the board fields
   4. Collaboration:
      1. BoardField
      2. Agent
3. BoardField
   1. Superclass: Board
   2. Subclass: -
   3. Responsibilities:
      1. Serves information about single field
   4. Collaboration:
      1. Board
      2. Agent
4. BoardStatus
   1. Superclass: -
   2. Subclass: -
   3. Responsibilities:
      1. Serves statistics about each simulation round
   4. Collaboration: Simulation
5. Agent
   1. Superclass: -
   2. Subclass: Student, Examiner
   3. Responsibilities:
      1. Tells agent position
      2. Moves the agent
      3. Tells agent id
   4. Collaboration:
      1. Simulation
6. Student
   1. Superclass: Agent
   2. Subclass: -
   3. Responsibilities:
      1. Informs about student current status
      2. Holds unique student parameters
      3. Manages student stats
   4. Collaboration:
      1. Simulation
7. Examiner
   1. Superclass: Agent
   2. Subclass: -
   3. Responsibilities:
      1. Examinates students
      2. Tells unique examiner suspicion
   4. Collaboration:
      1. Simulation