

ЗАДАЧІ ХІ ВСЕУКРАЇНСЬКОГО ТУРНІРУ ЮНИХ ІНФОРМАТИКІВ

Мельник Валентин Іванович,

Учитель інформатики Полтавського обласного ліцею-інтернату для обдарованих дітей із сільській місцевості при Кременчуцькому педагогічному училищі імені А.С. Макаренка, заслужений учитель України,

Мотурнак Євген Володимирович,

учитель КЗО «Навчально-виховний комплекс №100 «Загальноосвітній навчальний заклад — ліцей» Дніпропетровської міської ради, переможець конкурсу «Учитель року 2013» у номінації «Інформатика», заслужений учитель України.



Формат турніру ми докладно розглядали у статті в №4 журналу «Комп'ютер у школі та сім'ї» за 2012 рік. Тож розглянемо більш докладно цьогорічні завдання турніру.

Завдання I етапу Всеукраїнського турніру юних інформатиків 2013

Планета Земля, колыска людства, уже досить переповнена. Усе частіше про колонізацію інших космічних тіл люди говорять не як про теоретичну, а про практичну можливість. Але умови цих космічних тіл зовсім не пристосовані для людини. Тому першопрохідцями, тими, хто все підготує до приходу людей, будуть роботи.

Нині роботи займають все більше місце у нашому житті. Спочатку дослідницькі, потім промислові, військові і, нарешті, побутові. Роботи-іграшки, роботи-друзі, роботи-пилососи — усе це ми можемо побачити на полицях магазинів. Розвиток усіх цих видів роботів відбувається паралельно. Вони працюють за однаковими принципами і взагалі між ними немає жодної різниці. Чи ні?

Відчуйте себе розробником робототехнічної системи. Виробництво роботів є досить коштовним, тому більшість попередніх досліджень варто здійснювати на моделях. Пропонується створити комп'ютерну модель робота згідно представлених завдань.

Завдання турніру носить дослідницький і творчий характер і тому є відкритим. Команди-учасниці деталізують завдання, виходячи зі своєї концепції розв'язання. Цю концепцію вони мають викласти у вигляді технічного завдання і надати журі разом з іншою технічною документацією у файлі **tech_zavd.docx (doc)**.

Кожне наступне завдання є ускладненням попереднього. Але команди-учасниці можуть комбінувати вимоги різних завдань у своєму програмному продукті. Усі математичні моделі, створені у рамках проекту, мають бути описані і збережені у файлі **mat_mod.docx (doc)**.

Завдання 1

1.1. Розробити програму — виконавець алгоритмів. Під виконавцем ми розуміємо комп'ютерну модель робота.

1.2. Програма повинна містити редактор алгоритму, яка підтримує систему команд для керування виконавцем, що рухається площиною. Систему команд учасники розробляють самостійно, виходячи з функціональної повноти та ефективності.

1.3. Реалізувати в програмі двовимірну візуалізацію виконання алгоритму, заданого користувачем.

1.4. Забезпечити збереження алгоритму в файлі.

1.5. Забезпечити зчитування алгоритму з файлу.

1.6. Надати користувачеві засоби налагодження алгоритму.

Завдання 2

2.1. Побудувати модель руху робота в тривимірному просторі поверхнею довільного рельєфу. За необхідністю розширити систему команд.

2.2. Забезпечити завантаження рельєфу поверхні з файлу зображення в сірих тонах у форматі BMP. Висота поверхні визначається яскравістю пікселя. Користувачеві надати можливість керувати масштабом відображення горизонтальних та вертикальних вимірів поверхні.

2.3. Забезпечити тривимірну візуалізацію моделі руху робота заданою поверхнею за побудованим користувачем алгоритмом.

2.4. Надати можливість зміни ракурсу, наближення та віддалення камери.

Завдання 3

3.1. Додати в модель можливість модульної побудови робота. Приклади модулів: шасі, сенсор доотику, сенсор відстані, сенсор орієнтації у просторі, акселерометр, сервомотор, маніпулятор та інші. Модель повинна враховувати взаємне положення модулів, їх орієнтацію у просторі.

3.2. Розробити програму — виконавець алгоритмів для модульного робота. Визначити відповідну систему команд.

3.3. Розробити формат файлу для збереження даних про робота. Забезпечити запис та зчитування цих даних.

3.4. Програма повинна містити редактор алгоритму, що підтримує систему команд для керування моделлю робота, що рухається поверхнею довільного рельєфу та може взаємодіяти з різними об'єк-

тами. Система команд повинна містити можливість опитування датчиків, керування сервомоторами, іншими приладами.

3.5. Врахувати можливість виходу з ладу модулів робота та його часткове руйнування.

Завдання 4

4.1. Додати у модель можливість роботи робота в умовах різних планет. Змодельовати систему живлення робота.

4.2. Змодельовати систему керування роботом із Землі або орбітальної станції.

4.3. Врахувати характеристики поверхні, якою рухається робот: вологість, слизькість, осипання ґрунту, буксування.

4.4. Врахувати вплив навколишнього середовища: умови освітлення, атмосферні явища, стихійні лиха тощо.

Завдання 5. Автономний робот

5.1. Запрограмувати робота на виконання завдань пунктів 5.2.–5.4. для довільного рельєфу.

5.2. Починаючи з точки з координатами (x_0, y_0) , переміститися в точку (x_1, y_1) .

5.3. Починаючи з точки з координатами (x_0, y_0) , забрати в точці (x_1, y_1) об'єкт та доставити його в точку (x_2, y_2) . Об'єкт має форму прямокутного паралелепіпеда та має характеристики: розміри rx, ry, rz ; одиничний вектор напрямку OY локальної системи координат об'єкта в системі світових координат (nx, ny, nz) .

5.4. Починаючи з точки з координатами (x_0, y_0) , відшукати об'єкт та перемістити його в точку (x_1, y_1) .

5.5. Забезпечити функціонування робота у заданий період часу N (днів, годин, хвилин, секунд) в умовах агресивного середовища.

Розбір завдання I туру

Позаяк основою завдання є проблема реального життя, для вирішення якої необхідно звернутися до одного з методів наукового дослідження — моделювання. За фабулу в 2013 році була взята робототехніка як галузь, що швидко розвивається уже нині, а з плином часу потреба у спеціалістах у цій галузі буде тільки зростати.

У завданні 1 необхідно було створити систему виконання алгоритмів з двовимірною візуалізацією. Як приклади можна було розглянути такі системи, як Logo, Roo, Kodu.

Така система повинна складатися з трьох основних частин:

1. Редактор алгоритмів.
2. Підсистема аналізу і виконання алгоритмів.
3. Візуалізація виконання алгоритму.

Найскладнішою тут є підсистема аналізу й виконання алгоритмів. Але, насправді, складність її програмування тісно пов'язана з підходом, який було використано у редакторі алгоритмів. Якщо було використано стандартну систему текстового запису алгоритмів розробленою мовою програмування, то, дійсно, необхідно було створити повноцінний синтаксичний аналізатор. А його створення — це досить нетривіальне завдання. Інший підхід — використання графічної системи запису алго-

ритмів є само собою доволі складним, але полегшує процедуру аналізу й виконання алгоритму.

Розроблена система команд оцінювалася з позицій ефективності та функціональної повноти. З одного боку, команд не має бути багато, з іншого, вони мають забезпечувати створення будь-якого алгоритму, як то — лінійного, розгалуженого, циклічного, і використання підпрограм, а також роботу з даними.

Окрема увага приділялася можливостям налагодження алгоритму, пошуку помилок, покрокового виконання.

Традиційно, завдання 2 — це перехід до тривимірної візуалізації. Наразі можна було як скористатися відомими бібліотеками тривимірної візуалізації, так і створити власну. З огляду на те, що завдання візуалізації тривимірних ландшафтів стало вже стандартним у комп'ютерному моделюванні, існує досить багато засобів його виконання. Але слід зауважити, що такі бібліотеки як OpenGL, DirectX призначені лише для роботи з графікою, фізичне моделювання має міститись в програмі користувача. А, наприклад, Unity 3D, містить і обмежені можливості фізичного моделювання. Слід розуміти природу цих обмежень.

Unity 3D створена насамперед для розробки комп'ютерних ігор, а не справжніх наукових інструментів, тому результат її використання тільки схожий на реальність, і навіть схожий не у всіх випадках. Складне фізичне моделювання, якого вимагає завдання 3, можна було б реалізувати, використавши такі професійні інструменти, як MatLab, але тут уже складність інструмента перевищує складність завдання.

Завдання 3 вимагало створення докладної моделі робота, який складався б з певних частин — модулів. Кожен з модулів реального робота має певне розташування відносно інших модулів робота, яке може змінюватись в разі рухомого модуля. Також модуль має масу, момент інерції, розміри, певну міцність та інші характеристики. Формат даних для збереження робота мав містити всю цю інформацію. Робот у зібраному стані — це цілісний фізичний об'єкт, який має центр мас, момент інерції, засоби руху поверхнею та інше. Взаємодія модулів між собою і з навколишнім середовищем є предметом моделювання. Наприклад, якщо маніпулятор робота витягується вперед і підхоплює вантаж, центр мас робота зміщується, він може втратити рівновагу та перекинутись.

Завдання 4 — це перехід до більш реалістичної моделі, яка враховує систему живлення робота. Система живлення складається з трьох частин:

- підсистема отримання енергії;
- підсистема зберігання енергії;
- підсистема живлення модулів.

Кожен з модулів може бути джерелом енергії, контейнером для її зберігання, а також споживачем. У реальних робототехнічних системах енергетичний баланс дуже важливий. Падіння напруги акумулятора в процесі роботи може привести до зниження потужності сервомоторів, і, як наслідок, робота не

зможе підніматися по схилу, на який раніше легко заїжджав.

Також мають бути враховані характеристики ґрунту (вплив на рух), освітлення (вплив на датчики і сонячні батареї) та інші параметри зовнішнього середовища. Зокрема, слід було приділити увагу можливості руйнування робота під впливом навколишнього середовища.

Завдання 5 пропонувало низку тестових прикладів, на яких можна було перевірити функціонал робота і розробленої системи команд.

Загалом, детальні критерії оцінювання завдання І туру з боку досконалості й ефективності реалізації мали такий вигляд, як у табл. 1.

Таблиця 1

Завдання 1	
1.1	Наявність реалізованої програми-виконавця алгоритмів
1.2	Наявність опису системи команд
	Реалізація роботи з даними
	Реалізація лінійних алгоритмів
	Реалізація розгалужених алгоритмів
	Реалізація циклічних алгоритмів
1.3	Реалізація підпрограм
1.3	Реалізована двовимірна візуалізація
1.4	Можливість зберегти алгоритм у файлі
1.5	Можливість зчитувати алгоритм з файлу
1.6	Можливість покрокового виконання
	Можливість поблокового виконання
	Можливість перегляду значень параметрів робота користувачем
Завдання 2	
2.1	Наявність реалізованої програми-виконавця алгоритмів в 3D
	Візуалізація моделі робота в 3D
2.2	Можливість завантаження даних про рельєф з BMP
	Можливість задання горизонтального масштабу
	Можливість задання вертикального масштабу
2.3	Створена система 3D-візуалізації
	Використана готова бібліотека (OpenGL, DirectX,...)
	Використана бібліотека власного створення
2.4	Можливість зміни ракурсу перегляду
	Можливість наближення/віддалення камери
	Відсутність помилок відтворення зображення
Завдання 3	
3.1	Створена модель робота, що має модульну структуру
	Створені модулі введення даних (датчики)
	Створені модулі виведення даних (сервомотори, маніпулятори)
	Створені модулі керування

Продовження таблиці 1

3.1	Є можливість руху модулів відносно один одного
	Існує можливість редагування роботи в програмі
3.2., 3.4	Система команд враховує керування модулями
	Система команд включає введення даних з датчиків
	Система команд керування сервомоторами/іншими пристроями
3.3	Система команд включає перевірку систем робота
	Існує можливість запису робота до файлу
3.5	Існує можливість зчитування робота з файлу
	Врахована можливість руйнування робота
3.5	Врахована можливість руйнування окремих модулів
	Завдання 4
4.1	Змодельована система живлення робота в умовах автономності
	Враховані енерговитрати кожної дії робота
4.2	Змодельоване дистанційне керування роботом (із затримкою)
	Змодельовані можливі перешкоди зв'язку з роботом
4.3	Враховані вологість/слизкість ґрунту
	Враховане осипання ґрунту
	Враховане буксування
4.4	Врахований вплив умов освітлення
	Врахований вітер/поток рідини
	Враховані температура/атмосферний тиск/електромагнітні явища
Завдання 5	
5.1	Врахована взаємодія з об'єктами середовища
5.2	Виконане завдання для гладкого рел'єфу
	Виконане завдання для лабіриту
5.3	Виконане завдання для гладкого рел'єфу
	Виконане завдання для лабіриту
5.4	Виконане завдання пошуку
5.5	Робот виживає в умовах агресивного середовища

Завдання II етапу Всеукраїнського турніру юних інформатиків 2013

Індіана Джонс і пастка Майя

Усім вам добре відомі древні пророцтва племені Майя про кінець світу. На щастя, це пророцтво не збулось (або поки не збулось), але історія зовсім не про це...

Якось, відомий на весь світ шукач скарбів, Індіана Джонс потрапив до загадкової печери майя. Як веліли закони жанру кам'яні двері за ним зачинились і шляху назад не було. На щастя в печері горіло кілька ліхтарів і з малюнків на стінах професор Джонс зрозумів, що йому треба зробити, щоб відкрити шлях до подальших пригод!

Виявилось, що печера є прямокутником $N \times M$ клітинок. У деяких клітинках розташовані важкі каменюки, які Індіана Джонс не може ні посунути,

ні підняти. В інших клітинках стоять кілька тотемів язичьких богів, які він може зсунути з місця. Також на підлозі він знайшов відмітки, на яких, судячи з настінних малюнків, і стояли ці тотеми. Для виходу з печери, нашому герою необхідно повернути всі тотеми на свої місця (вважається що всі місця і всі тотеми однакові). Він може виконувати лише такі дії:

- переходити в сусідню вільну клітинку (по горизонталі або по вертикалі);
- якщо у сусідній з Індіаною клітинці стоїть тотем і наступна в цьому напрямку клітинка вільна (там немає ні тотему, ні каменю, ні межі печери), то він може пересунути тотем у вільну клітинку.

Завдання

А. Допоможіть професору Джонсу написати програму, що дасть йому можливість знайти правильну послідовність дій для виходу з печери.

Б. Проявіть творчий підхід до задачі, спробуйте додати до своєї програми додаткові можливості та функції (можливо, дати можливість зміни певних правил).

В. Напишіть штучний інтелект, який за заданим початковим розміщенням виведе Індіану Джонса з папки, приготованої Майя.

Технічні умови

Передбачити, що інформація про печеру зберігається у текстовому файлі в такому форматі:

- перший рядок містить два числа N, M — розміри печери. У наступних N рядках розміщено по M чисел, розділених пропусками, що відповідають клітинкам печери;
- 0 — вільне місце;
- 1 — камінь;
- 2 — тотем;
- 3 — відмітка;
- 4 — початкове положення Індіани.

Для перевірки надати папку з усіма необхідними для роботи програми файлами і документацією. Ім'я папки має відповідати назві команди.

Розбір завдання II туру

Як бачимо, завдання є ні чим іншим, як фабульним переформулюванням класичної комп'ютерної гри Сокобан. Якщо в першому турі перевірялася здатність команд-учасниць створити складну фізичну модель, то завдання другого туру вимагає лише акуратності програмування і чіткого планування роботи в умовах обмеженого часу.

Саме в цьому турі найяскравіше проявляється робота команди. Ролі, які можна запропонувати, як основні — це координатор, програміст, дизайнер, алгоритміст та секретар.

Координатор — слідкує за часом, розподіляє завдання, визначає пріоритети в роботі і приймає остаточне рішення. Це роль капітана, якщо він не є, наприклад, основним програмістом у команді. У завдання

програміста входить реалізація ідей команди у вигляді програмного коду й інтеграція в нього продуктів інших учасників — графіки, текстів, звуку тощо. Дизайнер займається розробкою графічного контенту і системи навігації. Секретар займається створенням документації, тестує створені фрагменти програми і розробляє довідкову систему. Алгоритміст має розробити ефективні алгоритми, які запрограмує програміст. Найбільш складною частиною його роботи є розробка алгоритму штучного інтелекту.

Наведемо детальні критерії, за якими оцінувалися повнота і досконалість реалізації програмного продукту (табл. 2).

Таблиця 2

Повнота реалізації	
1	Наявність програми
2	Наявність початкової заставки
3	Наявність довідки в програмі
4	Можливість зчитування конфігурації печери з файлу
5	Можливість керування головним героєм
6	Герой не проходить крізь камені, межі печери та тотеми
7	Тотем пересуваються за умовою
8	Розміщення всіх тотемів на відмітках відкриває вихід з печери
9	Додані додаткові рівні гри
10	Наявність редактора печер
11	Можливість запису печери в файл
12	Існує можливість зберігання/завантаження поточного стану гри
13	Графічне оформлення відповідає фабулі
14	Програма містить екран переможця
15	Наявна система штучного інтелекту для головного героя
16	Створене головне меню програми
	Творчий доробок
Досконалість реалізації	
1	Система штучного інтелекту розв'язує тривіальну задачу
2	Система штучного інтелекту розв'язує нетривіальну задачу
3	Програма працює без критичних збоїв
4	Програма перевіряє коректність введення початкових даних
5	Програма працює з комфортною швидкістю
6	Відсутні помилки відображення графічної інформації
7	Відсутні інші помилки

Завдання заочного туру Всеукраїнського турніру юних інформатиків наступного року буде доступне у вересні-жовтні. Слідкуйте за інформаційними листами.

Бажаємо успіхів у підготовці до ТЮІ-2014!

