

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І АРХІТЕКТУРИ

ГЕОМЕТРИЧНЕ МОДЕЛЮВАННЯ І КОМП'ЮТЕРНА ГРАФІКА

МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
ДЛЯ СТУДЕНТІВ, ЯКІ НАВЧАЮТЬСЯ ЗА НАПРЯМОМ ПІДГОТОВКИ
6.050101 «КОМП'ЮТЕРНІ НАУКИ»

Київ 2013

УДК 004.92
ББК 32.973
Б19

Укладачі: Є.В. Бородавка, кандидат технічних наук, доцент
В.В. Демченко, кандидат технічних наук, доцент

Рецензент: В.М. Міхайленко, доктор технічних наук, професор

Відповідальний за випуск: В.М. Міхайленко, доктор технічних наук,
професор, завідуючий кафедрою ІТППМ

*Затверджено на засіданні кафедри інформаційних технологій
проектування та прикладної математики, протокол №14 від 1
квітня 2013 року.*

Б19 Геометричне моделювання і комп'ютерна графіка: методичні
вказівки до виконання лабораторних робіт / Уклад. Є.В. Бородавка,
В.В. Демченко. – К.: КНУБА, 2013. – 16 с.

Методична розробка містить перелік лабораторних робіт, які
необхідно виконати студентам протягом курсу «Геометричне
моделювання і комп'ютерна графіка».

Призначено для студентів, які навчаються за напрямом підготовки
6.050101 «Комп'ютерні науки» при виконанні лабораторних робіт з
дисципліни «Геометричне моделювання і комп'ютерна графіка».

УДК 004.92
ББК 32.973

© Бородавка Є.В., Демченко В.В. 2013
© КНУБА, 2013

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Лабораторні роботи є логічним продовженням лекційного курсу з дисципліни «Геометричне моделювання і комп'ютерна графіка» і призначення для закріплення теоретичного матеріалу на практиці.

Тематика і зміст лабораторних робіт обумовлені основними розділами робочої навчальної програми з дисципліни «Геометричне моделювання і комп'ютерна графіка». Лабораторні роботи поділені на два модулі. Перший містить чотири лабораторні роботи, що присвячені двовимірним алгоритмам та зображенням. Другий модуль містить п'ять лабораторних робіт які присвячені побудові тривимірних об'єктів та візуальних ефектів.

М1.ЛР1. РОЗРОБКА ПРОГРАМИ ПЕРЕТВОРЕННЯ КАДРУВАННЯ

МЕТА РОБОТИ

Ознайомитись з операцією кадрування та засвоїти практичні прийоми розробки програм перетворення кадрування та їх застосування для формування графічних зображень на екрані дисплею на прикладі генерації фрактального зображення.

ЕТАПИ РОБОТИ

1. Розробити на мові C++ клас для виконання операції проектування.
2. Перекласти програму генерації фракталу папороті з мови BASIC, на мову C++ і оформити її у вигляді функції.
3. Протестувати створений застосунок.

МЕТОДИЧНІ ВКАЗІВКИ

Теоретичні виклади щодо операції кадрування, а також послідовність її виконання та формули можна знайти в конспекті лекцій. Тут ми зосередимося на практичній стороні реалізації класу на мові C++, який реалізує операцію кадрування.

Очевидно, що приватними полями класу будуть координати, що описують світове та приладове вікна, а також їх центри. Тобто для світового вікна це будуть наступні координати: FwX_0 , FwY_0 , FwX_m , FwY_m , FwX_c , FwY_c – всі вони дійсного типу. Відповідно для приладового вікна це будуть координати: FdX_0 , FdY_0 , FdX_m , FdY_m , FdX_c , FdY_c – всі вони цілого типу. Це одним приватним полем дійсного типу буде коефіцієнт масштабування.

Примітка. Координати вікон можна об'єднати в структури «точка з дійсними координатами» та «точка з цілими координатами» відповідно.

Тепер перейдемо до методів. Єдиним приватним методом буде функція розрахунку коефіцієнту масштабування. Публічними методами звичайно будуть конструктор та деструктор, а також три функції: SetWorld, SetDevice та головна наша функція WorldToDevice. Перші два методи на вхід отримують координати вікон – світового і приладового, та заносять ці дані у відповідні приватні поля. Метод SetDevice також викликає розрахунок коефіцієнта масштабування. Головний метод WorldToDevice на вхід отримує точку в світових координатах, а на виході видає точку в приладових координатах. Ця функція викликається для кожної точки зображення.

Теоретичні відомості про фрактали та їх різновиди наведені в конспекті лекцій з дисципліни. Одним з різновидів фракталів є фрактали на основі системи ітеративних функцій (Iterated Function System – IFS). Пропонується ознайомитися з прикладом побудови фрактального зображення папороті, що поданий у вигляді програми на мові BASIC. Основою побудови фракталу папороті є афінне перетворення, що визначається наступними формулами:

$$\begin{aligned}x' &= a \cdot x - b \cdot y + e \\y' &= c \cdot x - d \cdot y + f,\end{aligned}$$

де x, y – координати вхідної точки рівняння;
 x', y' – координати вихідної точки рівняння;
 a, b, c, d, e, f – коефіцієнти рівняння.

Узагальнена схема породження фрактальних зображень приведена на рис. 1. Процес починається з установки точки $P_0(x=0, y=0)$ до якої в залежності від значення випадкового числа застосовується відповідне рівняння системи IFS. Результат перетворень – точка з координатами, виводиться на екран після операції кадрування і знову подається на вхід блоку перетворень рівнянням IFS з вірогідністю датчика випадкових чисел.

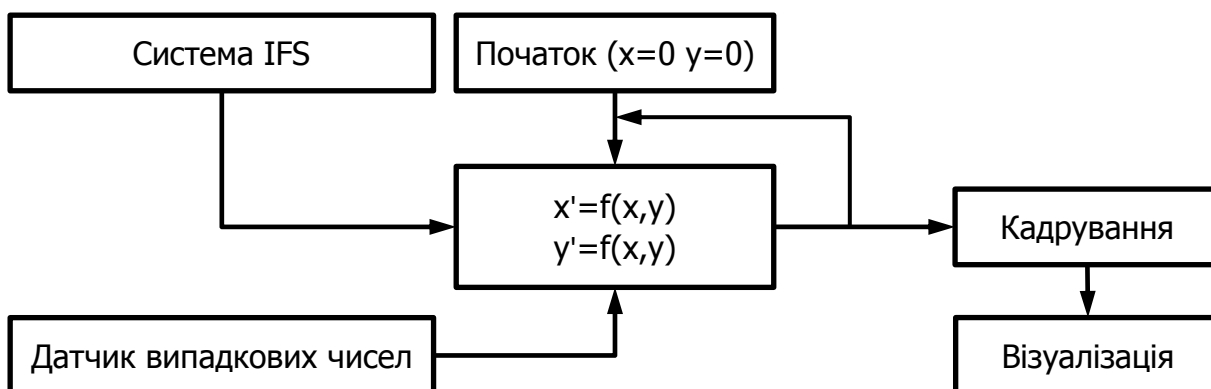


Рис. 1. Узагальнена схема генерації фрактальних зображень

В роботі необхідно виконати апробацію операції кадрування на прикладі виведення зображення фракталу папоротника. При побудові IFS – зображення папоротника насправді використовується не одне, а чотири афінні перетворення. Їх матричні рівняння виглядають однаково та розрізняються лише значеннями a , b , c , d , e та f . Кожному афінному перетворенню надається імовірність (число від 0 до 1), яка визначає його відносну важливість у порівнянні з іншими перетвореннями. Матричні рівняння разом з ймовірностями і утворюють IFS – набір для фрактального зображення папоротника.

Коефіцієнти a , b , c , d , e та f для IFS – папоротника, а також вірогідності афінних перетворень задані в таблиці 1.

Таблиці 1. Коефіцієнти афінних перетворень та їх імовірності

	a	b	c	d	e	f	Імовірність
Рівняння 1	0.00	0.00	0.00	0.16	0.00	0.00	0.01
Рівняння 2	0.85	0.04	-0.04	0.85	0.00	1.60	0.85
Рівняння 3	0.20	-0.26	0.23	0.22	0.00	1.60	0.07
Рівняння 4	-0.15	0.28	0.26	0.24	0.00	0.44	0.07

Далі наводиться програма на BASIC з необхідними коментарями. Оператори VIEW та WINDOW задають відповідно приладове та світлове вікна для операції кадрування в середовищі BASIC, які необхідно замінити на методи розробленого класу в C++ Builder.

```

`Перейти в графічний режим та ініціалізувати екран
SCREEN 12

CLS

`поле індикації
VIEW (0, 0) – (639, 479)

`вікно в світових координатах – область існування фракталу
WINDOW (-4, 0) – (6, 10)

`Ввести початкове число в генератор випадкових чисел
`та ініціалізувати x та y
RANDOMIZE TIMER

X = 0
Y = 0

`Розпочати цикл
WHILE INKEY$ = ""

`Присвоїти значення величинам a, b, c, d, e та f, виходячи з
ймовірностей афінних

```

```

`перетворень, які використовуються для побудови зображення
папоротника
r = RND
IF (r <= 0.01) THEN
a = 0 : b = 0 : c = 0 : d = 0.16 : e = 0 : f = 0
ELSEIF r > 0.01 AND r <= 0.86 THEN
a = 0.85 : b = 0.04 : c = -0.04 : d = 0.85 : e = 0 : f = 1.6
ELSEIF r > 0.86 AND r <= 0.93 THEN
a = 0.2 : b = -0.26 : c = 0.23 : d = 0.23 : e = 0 : f = 1.6
ELSE
a = -0.15 : b = 0.28 : c = 0.26 : d = 0.24 : e = 0 : f = 0.44
ENDIF
`Обчислити нові значення x та y та вивести точку на екран
newx = (a*x) + (b*y) + e
newy = (c*x) + (d*y) + f
x = newx
y = newy
`відображення на екрані пікселя зеленого кольору (код 2)
PSET (x, y), 2
`Продовжувати цикл до натиснення клавіші. Потім повернутися
`в текстовий режим екрану та закінчити
WEND
SCREEN 0

```

Результатом виконання лабораторної роботи повинна бути програма, що використовує об'єкт розробленого класу. Програма повинна відображати фрактал папоротника і реагувати на зміни розмірів вікна.

M1.ЛР2. ЗНАХОДЖЕННЯ ТОЧКИ ПЕРЕТИНУ ДВОХ ВІДРІЗКІВ

МЕТА РОБОТИ

Досконало вивчити та реалізувати на практиці один з базових алгоритмів геометричного моделювання, а саме знаходження точки перетину двох відрізків.

ЕТАПИ РОБОТИ

1. Створити класи, що необхідні для роботи алгоритму знаходження перетину двох відрізків, а саме: «точка» та «відрізок».

2. Створити тип-перелік всіх можливих варіантів розташування відрізків один відносно одного.
3. Написати програму, що дозволяє вводити координати відрізків як графічно так і з клавіатури.
4. Перевірити правильність роботи програми на практиці.

МЕТОДИЧНІ ВКАЗІВКИ

Теоретичні виклади алгоритму для знаходження точки перетину двох відрізків викладені в конспекті лекцій. Основними положеннями алгоритму є знання аналітичної геометрії на площині. Потрібно вміти будувати рівняння прямої по двом точкам в матричному вигляді та розв'язувати систему з двох лінійних рівнянь.

Щодо реалізації класу «точка», то тут все просто – такий клас ми описували на 3-му курсі під час вивчення дисципліни «Діалогові системи та лінгвістичне забезпечення САПР».

Клас «відрізок» повинен містити приватні поля, що описують коефіцієнти рівняння прямої: FA, FB, FC. Також повинні бути два поля класу «точка», що описують початкову та кінцеву точки відрізка. Конструктори повинні отримувати дві точки у вигляді набору з 4-х координат або двох об'єктів класу «точка». Окрім цього в класі повинен бути метод для розрахунку коефіцієнтів рівняння прямої, метод для малювання прямої на канві та метод перевірки перетину з іншим об'єктом класу «відрізок».

Можливі варіанти розташування відрізків наступні: накладаються; лежать на одній прямій; лежать на паралельних прямих; кінець або початок одного з відрізків лежить на іншому відрізку; кінцеві точки обох відрізків співпадають; лежать на прямих, що перетинаються, але самі не перетинаються; перетинаються; не перетинаються. У разі, якщо встановлено, що відрізки перетинаються, або лежать на прямих, що перетинаються, необхідно розрахувати координати точки перетину і видати результат.

Для перевірки правильності роботи програми необхідно підготувати тестові набори координат, що відповідають всім перерахованим вище варіантам. Також варто перевірити вироджені відрізки – паралельні осям координат.

Оскільки в роботі необхідно забезпечити графічне введення координат за допомогою миші, то доцільно в класі «точка» використовувати цілі координати.

М1.ЛР3. ШТРИХУВАННЯ ПОЛІГОНІВ В OPENGL

МЕТА РОБОТИ

Вивчити базові примітиви OpenGL для побудови простих плоских багатокутників та ознайомитися з режимом їх штрихування.

ЕТАПИ РОБОТИ

1. Засобами OpenGL побудувати фрагмент будівлі – фасадну стіну з вікном.
2. Розробити шаблон штрихування багатокутників у вигляді цегляної кладки (або іншої на вибір студента) в Excel.
3. В програмі створити масив розмірністю 128 байт, що описує шаблон штрихування.
4. Повторно побудувати багатокутники, що утворюють стіну з увімкненим режимом штрихування, щоб накласти на них розроблене штрихування.

МЕТОДИЧНІ ВКАЗІВКИ

Фрагмент фасаду будівлі, який необхідно побудувати, подано на рис. 2. Контур стіни треба розбити на декілька багатокутників, щоб по середині залишався отвір для вікна. Вікно побудувати відрізками. Побудову проводити в координатах світу OpenGL, що виставляються за умовчанням: лівий нижній кут $(-1,-1)$, верхній правий кут $(1,1)$.

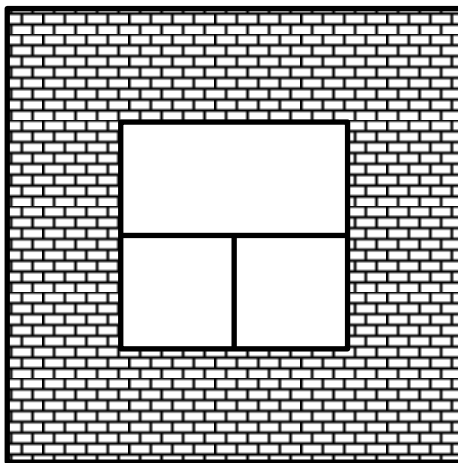


Рис 2. Фрагмент фасаду будівлі

Для візуального створення шаблону штрихування краще використати Excel чи аналогічний табличний процесор. Шаблон створюється для фрагменту зображення розміром 32x32 пікселі. Кожному пікселю відповідає 1 біт – якщо його значення 1, то зображення виводиться, якщо 0 – ні. Всього таких бітів буде 1024, тобто 128 байт. Для наочності побудуємо в Excel сітку розміром 32x32 і в місцях де зображення буде виводитися зафарбуємо комірки чорним кольором. В кожному рядку буде 4 байти по 8 біт кожен. В

залежності від того, які біти в байті мають значення 1, таке число і буде записане в масив. Оскільки масив лінійний, то всі 128 байт в ньому йдуть підряд. Порядок заповнення масиву – від лівого нижнього байту до правого верхнього. Все описане проілюстровано на рис. 3. Значення байтів подано в шістнадцятковому форматі.

	1				2				3				4				1	2	3	4																				
1																00	FF	FF	80																					
2																01	40	40	48																					
3																02	20	20	28																					
4																04	10	10	16																					
5																08	08	08	08																					
6																08	04	04	04																					
7																08	02	02	02																					
8																08	01	01	01																					
9																08	00	80	81																					
10																04	00	40	41																					
11																02	00	20	21																					
12																01	00	10	11																					
13																00	80	08	09																					
14																00	40	04	05																					
15																00	20	02	03																					
16																00	10	01	01																					
17																04	09	00	81																					
18																04	0A	00	41																					
19																04	04	00	21																					
20																04	0B	00	11																					
21																04	10	80	09																					
22																04	20	40	05																					
23																04	40	20	03																					
24																04	80	10	01																					
25																05	00	08	02																					
26																06	00	04	04																					
27																04	00	02	08																					
28																08	00	01	F0																					
29																1F	FF	00	00																					
30																20	00	00	00																					
31																40	00	00	00																					
32																80	00	00	00																					
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0								

Рис. 3. Ілюстрація шаблону штрихування

Щоб вірно розставити значення байтів необхідно знати правила перетворення двійкових чисел в десяткові, а бажано і в шістнадцяткові. Для кого це викликає труднощі, можна скористатися інженерним калькулятором.

Після створення зображення і визначення значень відповідних байтів переходимо до створення масиву в програмі. Для цього треба написати код, що наведено нижче. Якщо вносяться шістнадцяткові значення, то перед числом ставиться 0x.

```
GLubyte Stipple[]= {0x00, 0x00, 0x00, 0x80,  
.....  
0x80, 0xFF, 0xFF, 0x00};
```

Примітка. Ще раз наголошуємо на тому, що порядок внесення байтів знизу-вгору та зліва на право.

Щодо того, як вмикати і вимикати режим штрихування полігонів, а також як встановлювати поточний шаблон штрихування, то ця тема досить детально розглядається в посібнику з використання бібліотеки OpenGL.

М1.ЛР4. СТВОРЕННЯ ПРОГРАМИ З АФІННИМИ ПЕРЕТВОРЕННЯМИ

МЕТА РОБОТИ

Ознайомитися з основними афінними перетвореннями та навчитися їх використовувати на практиці. Навчитися використовувати команди OpenGL для виконання афінних перетворень. Ознайомитися з дисплейними списками OpenGL та навчитися використовувати їх на практиці.

ЕТАПИ РОБОТИ

1. Створити три дисплейні списки для кожного з трьох елементів фасаду будинку.
2. Ознайомитися з командами OpenGL для афінних перетворень та розібратися з їх параметрами.
3. Створити програму генерації фасаду будинку з використанням підготовлених блоків.

МЕТОДИЧНІ ВКАЗІВКИ

Теоретичні виклади щодо дисплейних списків можна знайти в посібнику з використання бібліотеки OpenGL. Елементи фасаду, для яких необхідно створити списки подано на рис. 4.

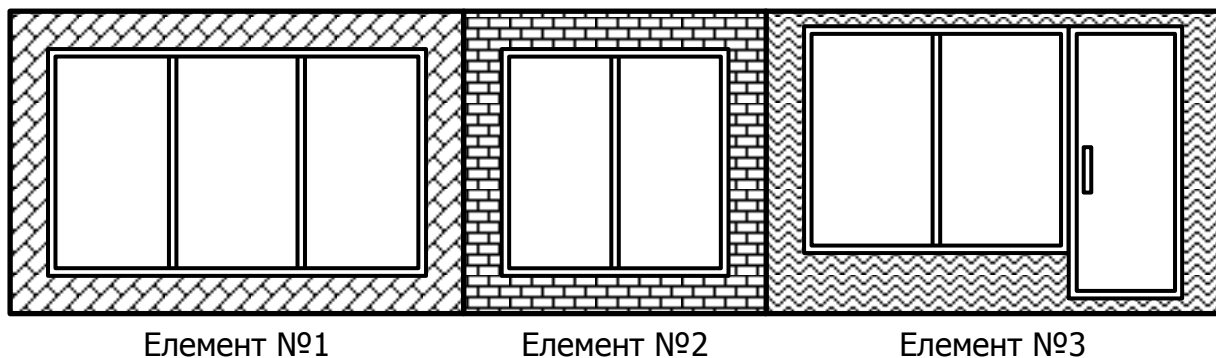


Рис. 4. Елементи фасаду

Примітка. При створенні елементів фасаду обов'язково застосувати шаблон штрихування, що був розроблений в попередній роботі.

Команди OpenGL для афінних перетворень досить прості. Їх всього три: переміщення (glTranslate), поворот (glRotate) і масштабування (glScale). Їх параметри та способи застосування детально описані в посібнику з використання бібліотеки OpenGL. Для повернення в початкову систему координат після виконання чергової групи афінних перетворень доцільно використовувати запам'ятовування поточної модельно-видової матриці в стеку з подальшим її відновленням. Для цього необхідно використовувати команди glPushMatrix та glPopMatrix. Детальніше про принципи їх використання читайте в посібнику.

В результаті роботи програми повинно бути побудоване зображення, на якому фасадні блоки розставлені наступним чином (рис. 5).

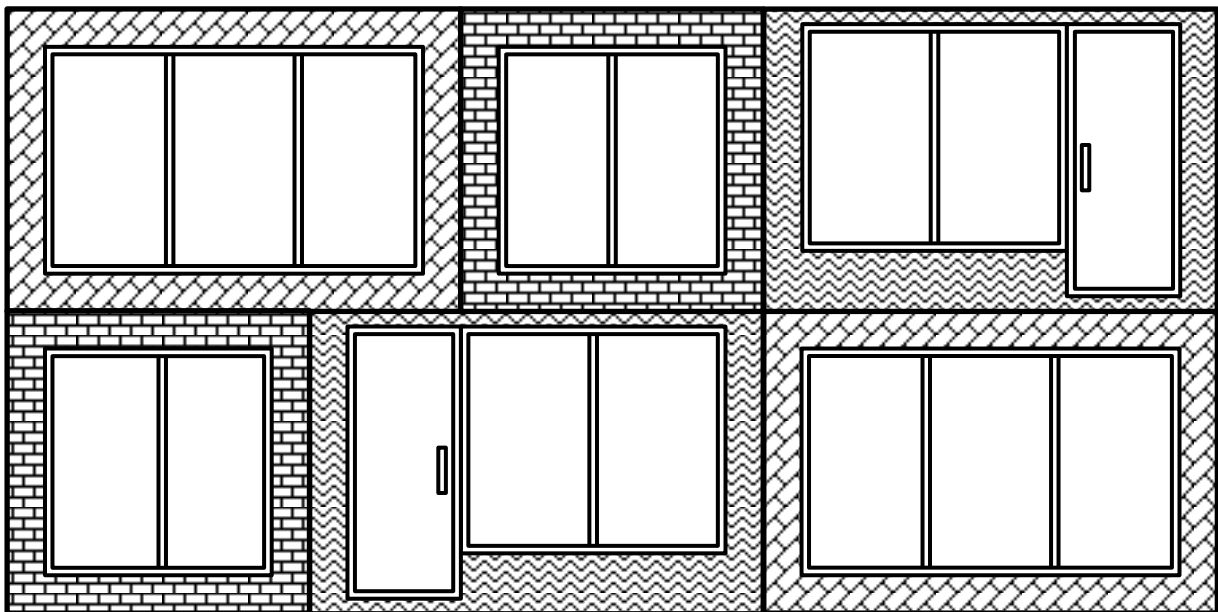


Рис. 5. Розташування елементів фасаду на зображенні

Примітка. Зверніть увагу на те, що елемент №3, який в нижньому ряду знаходиться по середині, є дзеркальним відображенням базового елемента. Це досягається шляхом обертання на 180° навколо осі OX.

Набір компонентів можна розширювати. Також можна створити сітку з 9 елементів, щоб мінімізувати порожнє місце у вікні програми.

М2.ЛР1. РОЗРОБКА ПРОГРАМИ ПРОСТОРОВОГО МОДЕЛЮВАННЯ

МЕТА РОБОТИ

Ознайомитись з основами подання просторових моделей граничних об'єктів та базовими функціями OpenGL відображення просторових моделей.

ЕТАПИ РОБОТИ

1. Ознайомитися з командами OpenGL для створення паралельних та центральних проєкцій.
2. Навчитися створювати плоскі опуклі полігони в тривимірному просторі засобами OpenGL.
3. Розробити програму генерації тривимірної фігури згідно заданого варіанту.

МЕТОДИЧНІ ВКАЗІВКИ

Перед тим, як здійснювати побудову тривимірних зображень, необхідно змінити матрицю проєкцій таким чином, щоб було візуально помітно тривимірність об'єкта. Види та типи проєкцій детально розглянуті в конспекті лекцій, а способи їх завдання в OpenGL описані в посібнику. В даній і подальших роботах ми будемо використовувати центральну перспективну проєкцію.

Для побудови тривимірних граничних моделей будемо використовувати команди OpenGL для побудови трикутників, чотирикутників та опуклих багатокутників. Для цього подану фігуру потрібно розбити на грані, а кожен грань розділити на опуклі багатокутники і побудувати їх за координатами вершин. Варіанти фігур подані на рис. 6.

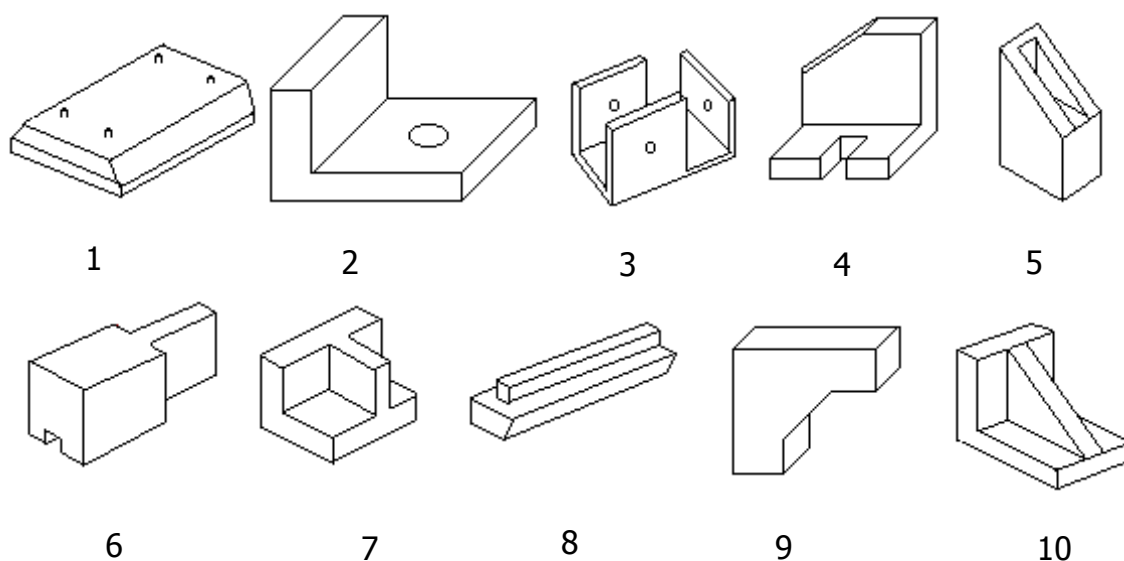


Рис. 6. Варіанти фігур

Примітка. Під час побудови фігур рекомендується використовувати дисплейні списки.

М2.ЛР2. ВИКОРИСТАННЯ TESS-ОБ'ЄКТІВ OPENGL

МЕТА РОБОТИ

Ознайомитися з принципами побудови неопуклих багатокутників в OpenGL із застосуванням Tess-об'єктів та навчитися застосовувати їх на практиці.

ЕТАПИ РОБОТИ

1. Вибрати неопуклий багатокутник, що описує фундамент багатоповерхової будівлі, з наведених варіантів або запропонувати свій.
2. Визначити координати всіх вершин цього багатокутника і створити в програмі масив цих вершин.
3. Побудувати програму, що відображає багатоповерхову будівлю заданого профілю.

МЕТОДИЧНІ ВКАЗІВКИ

Tess-об'єкти використовуються для спрощення побудови неопуклих багатокутників, а також багатокутників з отворами. Теоретичні виклади стосовно їх використання наведені в посібнику з використання OpenGL. Варіанти профілів фундаментів наведено на рис.7.

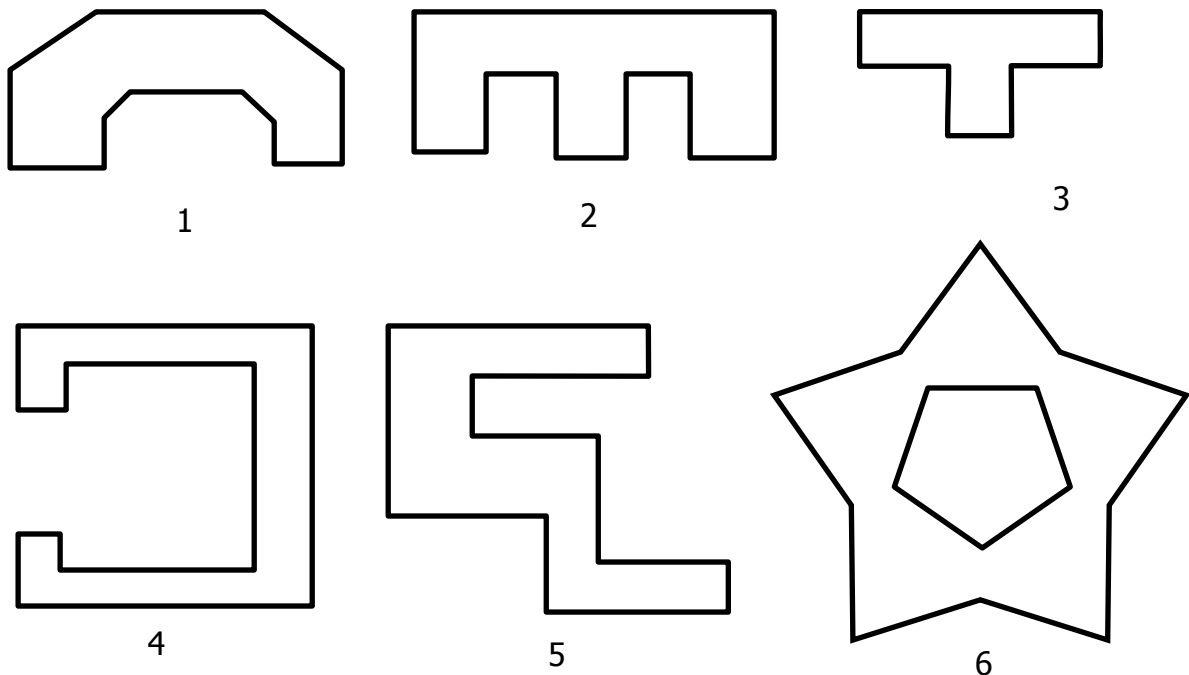


Рис. 7. Варіанти профілів фундаментів

Перед початком створення масивів вершин багатокутника необхідно визначитися з їх координатами. Це найпростіше зробити в будь-якому графічному редакторі. Але можна і просто на папірці в кліточку.

Примітка. Під час побудови потрібно використати дисплейні списки.

М2.ЛР3. РОБОТА З ДЖЕРЕЛОМ ОСВІТЛЕННЯ В OPENGL

МЕТА РОБОТИ

Ознайомитися з механізмами OpenGL для побудови ефекту освітлення. Розібратися з налаштуваннями джерела світла. Навчитися базовим прийомам анімації. Познайтися з quadric-об'єктами OpenGL.

ЕТАПИ РОБОТИ

1. Взявши за основу попередню роботу доповнити сцену джерелом світла.
2. Включити режим освітлення та розрахувати нормалі до всіх полігонів моделі.
3. Візуалізувати джерело світла quadric-об'єктом та створити анімацію руху цього світла навколо моделі будинку.

МЕТОДИЧНІ ВКАЗІВКИ

Теоретичні відомості про роботу з джерелом світла можна знайти в посібнику з використання OpenGL. Там же можна знайти опис роботи з quadric-об'єктами. Окрім включення освітлення не забудьте увімкнути кольори матеріалів та налаштувати їх.

Алгоритм розрахунку нормалі до площини за трьома точками наведено в методичних вказівках до виконання курсової роботи з даної дисципліни.

Для створення анімації необхідно виконувати обертання системи координат перед малюванням джерела світла. Для цього по таймеру треба змінювати кут повороту. Розберемо детальніше, як створити системний таймер і використати його.

Для початку в класі головної форми заведемо поле для ідентифікатора системного таймера:

```
DWORD FuTimerId;
```

Далі в конструкторі форми створюємо таймер:

```
FuTimerId=timeSetEvent(20,0,TimeCallBack,0,TIME_PERIODIC);
```

Не забуваємо знищити його на деструкторі:

```
timeKillEvent(FuTimerId);
```

Тепер залишилося описати функцію виклику таймера, яка буде викликатися з заданою періодичністю:

```
void TimeCallBack(UINT uTimerID, UINT uMessage, DWORD dwUser,
DWORD dw1, DWORD dw2)
{
    Spin += 1;
    InvalidateRect(FormGL->Handle, NULL, false);}

```

Змінна `Spin` в даному прикладі це значення кута повороту системи координат, що задається як параметр команди `glRotatef`.

М2.ЛР4. ПОБУДОВА ТІНІ ОБ'ЄКТА В OPENGL

МЕТА РОБОТИ

Ознайомитися з одним із методів побудови тіней. Ознайомитися з буфером трафарету OpenGL. Ознайомитися з функцією прозорості.

ЕТАПИ РОБОТИ

1. Взнявши за основу попередню роботу доповнити її горизонтальною площиною, що імітує поверхню землі.
2. Створити функцію для розрахунку матриці тіні.
3. Перемножити поточну модельно-видову матрицю на матрицю тіні і вивести зображення фігури у вигляді напівпрозорої тіні.
4. Використати буфер трафарету для запобігання накладання фрагментів тіні від різних граней.

МЕТОДИЧНІ ВКАЗІВКИ

Всі необхідні методичні рекомендації знаходяться в посібнику з використання OpenGL.

М2.ЛР5. ВИКОРИСТАННЯ ТЕКСТУР В OPENGL

МЕТА РОБОТИ

Ознайомитися з принципами накладання двовимірних текстур в OpenGL.

ЕТАПИ РОБОТИ

1. Підготувати файл текстурного зображення в форматі BMP.
2. Взнявши за основу попередню роботу доповнити модель накладанням текстури.

МЕТОДИЧНІ ВКАЗІВКИ

Всі необхідні методичні рекомендації знаходяться в посібнику з використання OpenGL.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. *Лященко А.А.* Геометричне моделювання і комп'ютерна графіка: використання бібліотеки OpenGL. / А.А. Лященко, В.В. Демченко, Є.В. Бородавка, В.В. Смірнов. – К.: КНУБА, 2009. – 90 с.

Навчально-методичне видання

ГЕОМЕТРИЧНЕ МОДЕЛЮВАННЯ І КОМП'ЮТЕРНА ГРАФІКА

МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
ДЛЯ СТУДЕНТІВ, ЯКІ НАВЧАЮТЬСЯ ЗА НАПРЯМОМ ПІДГОТОВКИ
6.050101 «КОМП'ЮТЕРНІ НАУКИ»

Укладачі: **БОРОДАВКА** Євгеній Володимирович
ДЕМЧЕНКО Віктор Вікторович

Редагування та коректура *Г.Є. Голіциної*

Комп'ютерна верстка *О.В. Кириченка*

Підписано до друку Формат 60x84_{1/16}.

Ум. друк. арк. 1,98. Облік.-вид. арк. 1,25.

Тираж 75 прим. Вид. № 21/І-10. Зам. №

КНУБА, Повітрофлотський проспект, 31, Київ, Україна, 03680

E-mail: red-isdat@knuba.edu.ua

Віддруковано в редакційно-видавничому відділі
Київського національного університету будівництва і архітектури

Свідоцтво про внесення до Державного реєстру суб'єктів
видавничої справи ДК №808 від 13.02.2002 р.