

Сумський державний педагогічний університет імені А. С. Макаренка

Фізико-математичний факультет

Кафедра інформатики

Щепотін Ярослав Євгенович

**ОСОБЛИВОСТІ ВИВЧЕННЯ ВЕКТОРНОЇ ГРАФІКИ
В УМОВАХ НЕФОРМАЛЬНОЇ ОСВІТИ**

Спеціальність: 014.09 Середня освіта (Інформатика)

Галузь знань: 01. Освіта

Кваліфікаційна робота

на здобуття освітнього ступеня бакалавра

Науковий керівник:

_____ О.В. Семеніхіна,

доктор педагогічних наук, професор,

«__» _____ 2021 року

Виконавець

_____ Я.Є. Щепотін

«__» _____ 2021 року

Суми 2021

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. АНАЛІЗ НАВЧАЛЬНИХ МАТЕРІАЛІВ ТА ОСОБЛИВОСТЕЙ ВИВЧЕННЯ ВЕКТОРНОЇ ГРАФІКИ В УМОВАХ НЕФОРМАЛЬНОЇ ОСВІТИ.....	7
1.1. Сутнісні характеристики неформальної освіти.....	7
1.2. Неформальна освіта у науково-педагогічних дослідженнях.....	10
1.3. Векторна графіка як вид комп'ютерної графіки.....	15
1.4. Місце векторної графіки в шкільному курсі інформатики.....	20
1.4.1. Структура навчання інформатики в загальноосвітній школі....	20
1.4.2. Програми з інформатики.....	22
1.4.3. Особливості методики вивчення векторної графіки у школі.....	29
1.5. Гурткова робота з комп'ютерної графіки як різновид неформальної освіти.....	31
1.5.1. Особливості організації гурткової роботи в школі.....	31
1.5.2. Робоча програма гуртка з інформатики «Комп'ютерна графіка» для учнів базової середньої школи.....	33
ВИСНОВКИ ДО РОЗДІЛУ 1.....	37
РОЗДІЛ 2. АНАЛІЗ МАТЕРІАЛІВ ДЛЯ ВИВЧЕННЯ ВЕКТОРНОЇ ГРАФІКИ ЗАСОБАМИ OPENGL В СЕРЕДОВИЩІ VISUAL STUDIO 2010.....	38
2.1. Що таке OpenGL, його особливості та значення в сфері комп'ютерної графіки.....	38
2.1.1. Бібліотеки, можливості та функції системи.....	38
2.1.2. Установка та ініціалізація в Windows.....	46
2.1.3. Архітектура та особливості синтаксису.....	47
2.2. Розробка практичних робіт гурткового навчання векторної графіки програмним способом.....	48

2.3. Реалізація власного проєкту як результат засвоєння практичних навичок с комп'ютерної графіки за допомогою OpenGL.....	50
2.3.1. Опис створеної програми.....	50
2.3.2. Структура сцени.....	51
2.3.3. Опис основних функцій.....	51
2.3.4. Реалізація об'єктів.....	52
2.3.5. Реалізація обробки натискань з клавіатури для управління камерою...	53
2.3.6. Демонстрація сцени.....	54
ВИСНОВКИ ДО РОЗДІЛУ 2.....	56
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТОК А.....	62

ВСТУП

Актуальність дослідження. Розвиток сучасного суспільства пов'язано з широкою інформатизацією всіх його сфер. Це вимагає від людей різних спеціальностей високого рівня володіння сучасними інформаційними технологіями.

На сьогоднішній день така область інформатики як комп'ютерна графіка охоплює всі види та форми представлення зображень, доступних для сприйняття людиною або на екрані монітора, або у вигляді копії на зовнішньому носії. Займаючи все більш міцні позиції, вона знаходить застосування не тільки в комп'ютерному світі, але і в різних сферах людської діяльності: наукові дослідження (візуалізація будови речовини, векторних полів і т.д.), медицина (комп'ютерна томографія), дослідно-конструкторські розробки. Тому вміння працювати з комп'ютерною графікою є невід'ємною частиною інформаційної грамотності будь-якої людини [5].

У зв'язку з цим, в даний час в школі необхідний аналіз існуючих програм та підходів до вивчення векторної і растрової графіки. Велике розмаїття професійних програмних засобів обробки об'єктів векторної графіки ставить перед учителем складні завдання, пов'язані в першу чергу з необхідністю обґрунтованого вибору конкретного програмного засобу, а також з розробкою методики навчання векторній графіці [7].

На сучасному етапі розвитку людства така галузь інформатики як комп'ютерна графіка знаходить своє відображення майже у всіх сферах людського життя. Використовуючи засоби комп'ютерної графіки ми можемо побачити ескізи ще не існуючих предметів та пристроїв, ознайомитися з принципами їхньої роботи. Використання комп'ютерної графіки полегшує роботу для інженерів та архітекторів, важливу, а іноді навіть ключову роль вона відіграє в рекламі та індустрії розваг. Засоби комп'ютерної графіки застосовуються для створення більшості сучасних мультфільмів та комп'ютерних ігор. Саме завдяки використанню сучасної комп'ютерної графіки

ми спостерігаємо значне покращення якості зображень комп'ютерних ігор, які іноді нагадують кінофільми

Метою роботи є визначення особливостей вивчення векторної графіки в умовах неформальної освіти.

Об'єкт дослідження: навчання учнів векторної графіки.

Предмет дослідження: особливості вивчення векторної графіки в умовах неформальної освіти.

Завдання до дипломної роботи:

1) Здійснити аналіз стану розробленості проблеми вивчення векторної графіки в умовах неформальної освіти;

2) Схарактеризувати основні дефініції дослідження: комп'ютерна графіка, векторна графіка, неформальна освіта, гурткова робота;

3) Провести аналіз форм неформальної освіти в контексті векторної графіки;

Апробація результатів дослідження:

Практична значущість дослідження полягає у розробці загальних рекомендацій для навчання учнів особливостям вивчення векторної графіки в умовах неформальної освіти та створенні практичних завдань для написання власних програм у середовищі Visual Studio 2010 мовою C++ засобами OpenGL.

Структура роботи: робота складається зі вступу, двох розділів, висновків та списку використаних джерел.

У вступі обґрунтовано актуальність обраної теми, визначені об'єкт, предмет, мета дослідження та сформульовані його основні завдання[8].

Перший розділ містить теоретичні основи дослідження. Розглянуто питання неформальної освіти, її позитивні та негативні аспекти, векторну графіку як різновид комп'ютерної графіки, її недоліки та переваги, а також особливості роботи з нею.

Другий розділ містить теоретичні засади, розробку лабораторних робіт і загальних рекомендацій до створення власних проєктів та написання власних програм у середовищі Visual Studio 2010 мовою C++ засобами OpenGL.

У висновках викладено основні результати кваліфікаційної роботи.

Матеріали допоможуть при опрацюванні багатьох питань, пов'язаних з вивченням векторної комп'ютерної графіки в умовах неформальної освіти.

РОЗДІЛ 1.

АНАЛІЗ НАВЧАЛЬНИХ МАТЕРІАЛІВ ТА ОСОБЛИВОСТЕЙ ВИВЧЕННЯ ВЕКТОРНОЇ ГРАФІКИ В УМОВАХ НЕФОРМАЛЬНОЇ ОСВІТИ

1.1. Сутнісні характеристики неформальної освіти

В даний час у всіх розвинених країнах світу на рівні державної політики досягнуто визнання того, що знання в усій зростаючій мірі стають основою розвитку суспільства, що необхідним засобом соціально-економічного прогресу в ХХІ столітті є трансформація людства в суспільство «довічного навчання».

Положення «від навчання на все життя до навчання через все життя» може служити гаслом сукупної системи освіти і найповніше відображає потенціал та завдання неформальної освіти, яке, в свою чергу, повинно стати одним з механізмів реалізації концепції освіти впродовж життя [15].

При прийнятті концепції безперервної освіти, де основним був закладений принцип «навчання протягом усього життя», почали розглядатися три види освіти - формальна, інформальна та неформальна.

Формальною в загальноприйнятій термінології є та освіта, яка:

- а) набувається учнями в спеціально призначених для навчання установах;
- б) здійснюється професійно підготовленим персоналом;
- в) веде до отримання загальноновизнаного документа про освіту;
- г) сприяє оволодінню учнями систематичними знаннями, вміннями і навичками при їх цілеспрямованій діяльності.

Інформальна освіта характеризується індивідуальною пізнавальною діяльністю, що супроводжує повсякденне життя і не обов'язково носить цілеспрямований характер. Вона є спонтанною і реалізуються за рахунок власної активності учнів в насиченому культурно-освітньому середовищі; спілкуванні, читанні, відвідуванні закладів культури, подорожі, засобах масової інформації, коли дорослий перетворює освітні потенціали суспільства в дієві

фактори свого розвитку, результат повсякденної робочої, сімейної та дозвільної діяльності, не має певної структури [18].

Неформальна освіта найчастіше відбувається за межами освітнього простору, де чітко визначені завдання, методи і результати навчання. В даному випадку мова може йти про освітні установи або громадські організації, про клуби або гуртки, про індивідуальні заняття з репетитором або тренером, про наставництво і дистанційне навчання, про різноманітні тренінги, майстер-класи, конференції, семінари, курси, короткострокові програми, які можуть бути запропоновані на будь-якому етапі освіти або трудової діяльності. Відмінною рисою неформальної освіти є те, що вона, як правило, не супроводжується видачею документа, в більшості випадків носить систематичний і цілеспрямований характер. Що характерно, неформальна освіта надається людям всіх вікових категорій протягом усього життя.

Представники науково-дослідного світу, вивчаючи роботу середніх і вищих навчальних закладів, які застосовують методику неформального навчання, прийшли до висновку, що навчаються один у одного, де викладач не керує процесом отримання знань, а є другом і партнером, набувають найкращу мотивацію до навчання.

Неформальне навчання можна отримувати в будь-якому місці і в будь-який час. У світі проводяться велика кількість різних заходів, де відбувається обмін досвідом та знаннями з потрібних тем.

Відмінна риса таких заходів - це неформальний характер взаємодії. Відбувається обмін досвідом і знаннями, як за допомогою презентацій, доповідей, так і відпочиваючи. Такі заходи є гарною ілюстрацією неформального навчання, тому що освітня активність цілеспрямована, визначена її тривалість і відбувається поза формальної системи освіти.

Основна ознака неформальної освіти - відсутність єдиних, в тій чи іншій мірі стандартизованих вимог до результатів навчальної діяльності. При цьому можуть бути присутніми всі інші ознаки навчання - доцільність, процедура зарахування, лекційно-семінарські заняття, система оцінювання успішності,

сертифікат про закінчення із зазначенням змісту пройденого курсу. Однак останній зазвичай не дає право займатися трудовою діяльністю на професійних засадах або надходити в навчальні заклади, де потрібне документальне підтвердження рівня академічної кваліфікації [14].

Неформальна освіта - це не систематизоване навчання учня знанням та навичкам. Воно так само є важливою складовою частиною його соціалізації, допомагає йому опанувати нові соціальні ролі, сприяє духовному розвитку. По відношенню до системи формальної освіти в сучасному суспільстві воно відіграє допоміжну роль [19].

Неформальна освіта так само є цілеспрямованою, але цілі в ній задані освітніми потребами самих учнів (освіта за вибором добровільна). Вона є доповненням або альтернативою формальної освіти. Здійснюється в приватних і державних освітніх установах, іноді поза спеціальним освітнім простором, передбачає видачу сертифіката недержавної зразка або не веде до сертифікації видачі документа.

До специфічних характеристик неформальної освіти, затребуваних в сучасному суспільстві, відносяться:

- орієнтація на конкретні освітні запити різних груп населення;
- особлива увага до освітніх потреб конкретних категорій осіб (інвалідів, мігрантів);
- високий рівень персональної активності учнів, відсутність примусового характеру, підстави на власній мотивації;
- внутрішня відповідальність зростає за результат освітньої діяльності, високий особистісний сенс навчання;
- розвиток якостей особистості, які забезпечують сприятливі передумови для гідного життя, а також успішної участі в суспільній і трудовій діяльності;
- розвиток мобільності в мінливих умовах сучасного світу;
- гнучкість в організації та методах навчання; організація відносин між учасниками освітнього процесу на взаємній повазі, участі, демократичній культурі [6].

1.2. Неформальна освіта у науково-педагогічних дослідженнях

Початок дискусій в сфері неформальної освіти за кордоном поклав філософ Д. Дьюї і засновник *андрогогіки* М. Ноулес в кінці XIX - середині XX століття. Перші визначення неформальної освіти було дано Ф. Кумбсом і М. Ахмедом, які до неї віднесли будь-яку організовану навчальну діяльність за межами формальної освіти і визначили її як окрему діяльність, спрямовану на служіння суб'єктам навчання, реалізує особистісні цілі навчання [30].

Поняття «*неформальна освіта*» на Заході стало предметом пильної уваги в 60-ті , особливо в 70-ті роки, так як школа більше не була єдиним місцем навчання і не могла монополювати претендувати на просвітницьку роль в суспільстві, освіта та навчання вже не розглядалося як синоніми «навчання в школі», відроджувався інтерес до нових, не традиційних форм і методів навчання.

У зарубіжній педагогіці позначаються основні риси неформальної освіти:

- а) воно являє собою організовану структуровану діяльність;
- б) призначається для ідентифікації цільової групи;
- в) організовується для досягнення певного набору навчальних цілей;
- г) ці види діяльності практикуються поза існуючої системи освіти і розраховані на осіб, офіційно не охоплених шкільною освітою.

Кордон між формальною і неформальною освітою не завжди чітко простежується: в структурах формальної освіти є аспекти неформального, такі як залучення непрофесіоналів в якості вчителів, участь батьків або членів громади в навчальному процесі або в керівництві школою.

Виділяються також і відмінні риси неформальної освіти:

- функціональний характер змісту, його сприйнятливості до місцевого середовища і здатність чуйно реагувати на його потреби;
- специфіка цілей, які часто встановлюються на найближчу перспективу, обмежуються географічно, контекстуально або рамками групи;
- програма складається з урахуванням конкретних, заздалегідь визначених потреб з орієнтацією на учня;

- гнучкість в здійсненні;
- неоднорідність цільових груп;
- діяльність піддається організації і систематизації, є протилежною рутині;
- діяльність, епізодично організована і короткострокова;
- використання викладачів-добровольців або позаштатних викладачів, залучення непрофесіоналів з оплатою або на добровільних засадах;
- самоокупність і широку участь.

Важливу роль в неформальній освіті грає особливе середовище, яке створюється з конкретними педагогічними цілями. Позиція актуалізації створення спеціальних освітніх середовищ спирається в першу чергу на наукові погляди Л.С. Виготського [5], який визнавав соціальне середовище головним виховним, особистісно домінуючим фактором. Він визначав найважливішим, першочерговим завданням педагога організацію такого середовища. Прикладом специфічної реалізації цих ідей в сучасній практиці служить організація літніх шкіл, експедицій, екологічних таборів та деяких інших заходів неформальної освіти.

Сьогодні у вітчизняній педагогіці з'явилося чимало робіт, присвячених неформальній освіті. Предметом дослідження вчених стали як теоретичні, так і практичні питання освіти дітей та молоді (І. Ардабацкая, Е.С. Бабаєва, І.М. Бюрюкова, А.В. Золотарьова, І.В. Іванова, Т.В. Мухлаєва, Т.С. Комісарова, М.Є. Кульпедінова, А.М. Макарьський).

Так, Е.С. Бабаєва, узагальнивши міжнародний досвід неформальної освіти, визначила його суб'єктні характеристики:

- орієнтування на освітні потреби конкретних категорій учнів;
- добровільність і високий рівень мотивації;
- високий особистісний сенс навчання;
- мобільність і високий рівень активності учнів;
- гнучкість в організації та методах навчання;
- вчення, засноване на співпраці [2].

В Україні неформальна освіта трактується по-різному. Ми не можемо погодитися з І.М. Бірюкової, яка вважає, що неформальна освіта позначає будь-яку організацію, що може бути придбаною поза системи формальної базової і додаткової освіти, так як сутнісні ознаки неформальних практик присутні в будь-яких формальних освітніх системах різного рівня.

Автор вважає, що в даний час система неформальної освіти вирішує основні завдання: компенсаторну (надає освіту, яке допомагає надолужити втрачене з деяких причин, незалежно від рівня освіти, статі, віку і т.д.); адаптується (постійне пристосування індивідуума до мінливого світу, новим суспільним і економічним умовам); Такі ж завдання може вирішувати і додаткова освіта [4].

Погоджуючись з І.В. Іванової, можна вважати, що принципова відмінність формальної та неформальної освіти полягає в тому, що для останньої характерні такі риси: добровільність і вільний вибір обсягу і темпу освоєння освітньої програми, її змісту, навчання за індивідуальним освітнім маршрутом; відсутність єдиних стандартів, але підпорядкованість природі дитини, його «нормальному розвитку» (В. І. Слободчиков); неформальні практики реалізуються в дитячих об'єднаннях (гуртках, клубах, секціях, майстерень) і часто визначаються авторськими програмами педагога [11].

Відомо, що неформальна освіта досить «комфортно себе почуває» в системі додаткової освіти дітей, де присутня атмосфера доброзичливості і затишку, та сама «ситуація успіху» (за Л. С. Виготському), партнерські відносини між педагогом і учнями, які, без сумніву, сприяють ефективному навчанню і гуманістичного виховання.

Питанням неформальної освіти в школі присвячені роботи І.А. Ардабацької [1]. Автор вважає, що в сучасному постіндустріальному суспільстві школа втратила монополію на передачу підростаючому поколінню культурних цінностей і необхідних для продуктивного життя відомостей. Згідно з І.А. Ардабацькою, щоб зберегти значення шкільної освіти не тільки в житті держави і суспільства, а й в житті кожної людини, слід говорити про

зміну системи координат, в якій здійснюється сучасний освітній процес. Необхідний відхід від формалізму в освіті, від сприйняття підростаючого покоління, як об'єкта впливу, як економічного, соціального або політичного ресурсу, як маргінальної групи. Тільки прийняття учнів як партнерів по взаємодії і співпраці дозволяє сучасному вчителю зберегти свій авторитет, а освітньої організації реалізувати завдання формування соціально активних і потенційно успішних громадян [1].

Не можна не погодитися з твердженням, що неформальна і додаткова освіта дітей вирішують одні й ті ж завдання, орієнтовані на подібні цільові групи, здійснюють свою діяльність приблизно в одних і тих же напрямках. Разом з тим, в публікаціях останніх років представлена й інша позиція, яка полягає в тому, що, незважаючи на схожі риси, додаткова освіта дітей відрізняється від структур неформальної освіти особливостями функціонування, інституційною інфраструктурою, джерелами фінансування і т.д. У деяких дослідженнях в якості суб'єктів неформальної освіти розглядаються громадські об'єднання [1, 10].

І.А. Ардабацька відстоює точку зору, що неформальна освіта може сприйматися не тільки як додаткове, але і як невіддільне від формальної. При цьому неформальний компонент є історично сформованою частиною шкільного життя.

Таким чином, неформальна освіта не є для шкіл абсолютної чимось особливим. Все, що потрібно від адміністрації освітньої організації - заохочувати випадки трансформації формальних освітніх практик в неформальні. Один із шляхів досягнення цього - створювати середовища спільного навчання, де формальне і неформальне уживається між собою [1].

Питанню неформальної освіти велику увагу в своїх роботах приділяє А.В. Золотарьова [10], яка констатує той факт, що кордони формальної та неформальної освіти активно перетинаються як у рамках шкільної (основної) освіти, так і в рамках додаткової освіти дітей (ДОД). Автор вважає, що додатка (позашкільна) освіти має певний потенціал для розвитку неформальної освіти.

Ролі неформальної освіти в соціальному самовизначенні дітей і молоді присвячені роботи М.Є. Кульпедінової [13]. І для нас дуже важливо, що автор приділяє в своїх роботах багато уваги дитячим громадським об'єднанням, як соціальним інститутам, в яких реалізуються програми неформальної освіти, тому що адаптаційний характер неформальної освіти має першорядне значення.

Важливе значення набуває соціальна орієнтованість неформальної освіти, особливо в рамках дитячих об'єднань. Вона в громадському об'єднанні є за своєю суттю соціальною. Загальною його складовою в громадських об'єднаннях, незалежно від їх провідного профілю (екологічні, патріотичні, миротворчі, економічні, краєзнавчі тощо), є соціально орієтованим навчанням і вихованням. Специфіка соціального навчання в громадському об'єднанні - пріоритетність навчання, а не тільки викладання, як це має місце в формальному освіті, природна інтеграція навчання і практичної діяльності [14].

Його особливі принципи: «вчення через діяльність» («навчання через справу»), «вчення в реальному житті», соціальна спрямованість результату, самоорганізація, діалогічність, співпраця співзвучні положенням продуктивного освіти. Це сприяє розумінню учасниками діяльності об'єднань, для чого потрібні придбані знання і як, з більшою користю, можна ними скористатися в кожному конкретному випадку. Додамо до цього, що продуктивне навчання найчастіше здійснюється на основі проектної діяльності, яка властива сучасним об'єднанням.

Таким чином, аналіз публікацій останніх років показав, що більшість авторів розглядають неформальну освіту дітей через призму додаткової. Додаткова освіта називається невід'ємною частиною неформального, розглядається як «неформальне і безперервне» або як «формалізований інститут неформальної освіти». Багато авторів відстоюють точку зору, що неформальна освіта може сприйматися не тільки як додаткове, але і як невіддільне від формального шкільного. Інші констатують той факт, що кордони формальної та неформальної освіти активно перетинаються як у рамках шкільної (основної) освіти, так і в рамках додаткової освіти дітей.

1.3. Векторна графіка як вид комп'ютерної графіки

Комп'ютерна графіка - це розділ інформатики, який займається проблемами отримання різних зображень (малюнків, креслень, мультиплікації) на комп'ютері.

У комп'ютерній графіці розглядаються наступні завдання:

- представлення зображення в комп'ютерній графіці;
- підготовка зображення до візуалізації;
- створення зображення;
- здійснення дій із зображенням.

Під комп'ютерною графікою зазвичай розуміють автоматизацію процесів підготовки, перетворення, зберігання і відтворення графічної інформації за допомогою комп'ютера. Під графічною інформацією розуміються моделі об'єктів і їх зображення.

У разі, якщо користувач може управляти характеристиками об'єктів, то говорять про інтерактивну комп'ютерну графіку.

Інтерактивна комп'ютерна графіка - це розділ комп'ютерної графіки, в якому передбачається використання комп'ютерів для підготовки і відтворення зображень, але при цьому користувач має можливість оперативно вносити зміни в зображення безпосередньо в процесі його відтворення, тобто передбачається можливість роботи з графікою в режимі діалогу в реальному масштабі часу [11].

Інтерактивна графіка являє собою важливий розділ комп'ютерної графіки, коли користувач має можливість динамічно керувати вмістом зображення, його формою, розміром і кольором на поверхні дисплея за допомогою інтерактивних пристроїв управління.

Історично першими інтерактивними системами вважаються системи автоматизованого проектування, які з'явилися в 60-х роках. Вони являють собою значний етап в еволюції комп'ютерів і програмного забезпечення. В системі інтерактивної комп'ютерної графіки користувач сприймає на дисплеї зображення, що представляє деякий складний об'єкт, і може вносити зміни в

опис (модель) об'єкта. Такими змінами можуть бути як введення і редагування окремих елементів, так і завдання числових значень для будь-яких параметрів, а також інші операції по введенню інформації на основі сприйняття зображень.

Галузі застосування комп'ютерної графіки не обмежується одними художніми ефектами. У всіх галузях науки, техніки, медицини, в комерційній та управлінській діяльності використовуються побудовані за допомогою комп'ютера схеми, графіки, діаграми, призначені для наочного відображення різноманітної інформації. Конструктори, розробляючи нові моделі автомобілів і літаків, використовують тривимірні графічні об'єкти, щоб представити остаточний вигляд виробу. Архітектори створюють на екрані монітора об'ємне зображення будівлі, і це дозволяє їм побачити, як воно впишеться в ландшафт.

Можна розглянути наступні сфери застосування комп'ютерної графіки [23].

Одним з напрямків є наукова графіка. Перші комп'ютери використовувалися лише для вирішення наукових і виробничих завдань. Щоб краще зрозуміти отримані результати, виробляли їх графічну обробку, будували графіки, діаграми, креслення розрахованих конструкцій. Перші графіки на машині отримували в режимі символічної друку. Потім з'явилися спеціальні пристрої - графопостроители (плоттери) для креслення креслень і графіків чорнильним пером на папері. Сучасна наукова комп'ютерна графіка дає можливість проводити обчислювальні експерименти з наочним поданням їх результатів [16].

Ділова графіка - галузь комп'ютерної графіки, призначена для наочного представлення різних показників роботи установ. Планові показники, звітна документація, статистичні зведення - ось об'єкти, для яких за допомогою ділової графіки створюються ілюстративні матеріали. Програмні засоби ділової графіки включаються до складу електронних таблиць.

Конструкторська графіка використовується в роботі інженерів-конструкторів, архітекторів, винахідників нової техніки. Цей вид комп'ютерної графіки є обов'язковим елементом САПР (систем автоматизації проектування).

Засобами конструкторської графіки можна отримувати як плоскі зображення (проекції, перетину), так і просторові тривимірні зображення [22].

Ілюстративна графіка - це довільне малювання і креслення на екрані комп'ютера. Пакети ілюстративній графіки відносяться до прикладного програмного забезпечення загального призначення. Найпростіші програмні засоби ілюстративної графіки називаються графічними редакторами.

Художня і рекламна графіка стала популярною багато в чому завдяки телебаченню. За допомогою комп'ютера створюються рекламні ролики, мультфільми, комп'ютерні ігри, відеоуроки, відеопрезентації. Графічні пакети для цих цілей вимагають великих ресурсів комп'ютера за швидкодією і пам'яті. Відмінною особливістю цих графічних пакетів є можливість створення реалістичних зображень і "рухомих картинок". Отримання малюнків тривимірних об'єктів, їх повороти, наближення, видалення, деформації пов'язано з великим обсягом обчислень. Передача освітленості об'єкта в залежності від положення джерела світла, від розташування тіней, від фактури поверхні, вимагає розрахунків, які враховують закони оптики.

Комп'ютерна анімація - це отримання рухомих зображень на екрані дисплея. Художник створює на екрані малюнки початкового і кінцевого положення рухомих об'єктів, всі проміжні стани розраховує і зображує комп'ютер, виконуючи розрахунки, що спираються на математичний опис даного виду руху. Отримані малюнки, що виводяться послідовно на екран з певною частотою, створюють ілюзію руху.

Поява глобальної мережі Інтернет призвело до того, що комп'ютерна графіка стала займати важливе місце в ній. З'явилося поняття графіка для Інтернету. Все більше вдосконалюються способи передачі візуальної інформації, розробляються досконаліші графічні формати, відчутно бажання використовувати тривимірну графіку, анімацію, весь спектр мультимедіа.

Розрізняють три *види комп'ютерної графіки*. Це растрова графіка, векторна графіка і фрактальна графіка. Вони відрізняються принципами

формування зображення при відображенні на екрані монітора або при друці на папері.

Растровий метод - зображення представляється у вигляді набору забарвлених точок. Растрову графіку застосовують при розробці електронних (мультимедійних) і поліграфічних видань. Ілюстрації, виконані засобами растрової графіки, рідко створюють вручну за допомогою комп'ютерних програм. Найчастіше для цієї мети використовують скановані ілюстрації, підготовлені художниками, або фотографії. Останнім часом для введення растрових зображень в комп'ютер широко використовують цифрові фото- і відеокамери.

Більшість графічних редакторів, призначених для роботи з растровими ілюстраціями, орієнтовані не стільки на створення зображень, скільки на їх обробку. В Інтернеті поки застосовуються тільки растрові ілюстрації.

Векторний метод - це метод представлення зображення у вигляді сукупності відрізків і дуг і т.д. В даному випадку вектор - це набір даних, що характеризують який-небудь об'єкт.

Програмні засоби для роботи з векторною графікою призначені в першу чергу для створення ілюстрацій і в меншій мірі для їх обробки. Такі засоби широко використовують в рекламних агентствах, дизайнерських бюро, редакціях і видавництвах. Оформлювальні роботи, засновані на застосуванні шрифтів і найпростіших геометричних елементів, вирішуються засобами векторної графіки набагато простіше.

Програмні засоби для роботи з фрактальною графікою призначені для автоматичної генерації зображень шляхом математичних розрахунків. Створення фрактальної художньої композиції полягає не в малюванні або оформленні, а в програмуванні.

Фрактальна графіка, як і векторна - обчислюється, але відрізняється від неї тим, що ніякі об'єкти в пам'яті комп'ютера не зберігаються (будується зображення за або по або по системі тому), тому нічого формулювати крім формули, не треба.

коефіцієнти коефіцієнти у рівнянні, отримати отримати зовсім картинукартину. Здатність фрактальної графіки моделювати образи живої природи обчислювальним шляхом часто використовують для автоматичної генерації незвичайних ілюстрацій.

Векторне зображення - це графічний об'єкт, побудований з геометричних примітивів, таких як точки, лінії, сплайни і багатокутники.

Опишемо переваги векторної графіки

Розмір займаної описовою частиною, не залежить від реальної величини об'єкта, що дозволяє, використовуючи інформаціїкількість інформації, описати скільки раз раз великий файлом файлом мінімального розміру.

У зв'язку тимтим, що інформація про об'єкт зберігається в описовій формі, можна нескінченно збільшити графічний примітив, наприклад, дугу кола, і вона залишиться гладкою. З іншого боку, якщо крива представлена у вигляді ламаної лінії, збільшення покаже, що вона насправді не крива.

Параметри об'єктів зберігаються і можуть бути легко змінені. Також це означає що переміщення, масштабування, обертання, заповнення та т. д. Не погіршать якості малюнка. Більш того, зазвичай вказують розміри в апаратно-незалежних одиницях, які ведуть до найкращої можливої растеризації на растрових пристроях.

При збільшенні або зменшенні об'єктів товщина ліній може бути задана постійною величиною, незалежно від реального контуру.

Зазначимо про недоліки векторної графіки

Чи не кожен об'єкт може бути легко зображений у векторному вигляді - для подібного оригінального зображення може знадобитися дуже велика кількість об'єктів і їх складності, що негативно впливає на кількість пам'яті, займаної зображенням, і на час для його відображення (відтворення).

Переклад векторної графіки досить простий. Але дороги назад, як правило, немає - трасування растра, при тому що вимагає значних обчислювальних потужностей і часу, не завжди забезпечує високу якості векторного малюнка.

Формати векторної графіки: .cdr, .ai, .cmx, .eps, .fla, .svg, .swf, .wmf.

До програмних засобів створення та обробки векторної графіки відносяться наступні ГР: CorelDraw, Adobe Illustrator, а також векторизатор (трасувальники) - спеціалізовані пакети перетворення растрових зображень в векторні.

Як в растровій, так і в векторній графіці необхідний спосіб кодування кольору [9].

1.4. Місце векторної графіки в шкільному курсі інформатики

1.4.1. Структура навчання інформатики в загальноосвітній школі

Базовий курс інформатики

Базовий курс інформатики повинен забезпечувати засвоєння основних теоретичних положень інформатики, опанування науковими основами, методами і засобами інформаційних технологій, тобто забезпечувати обов'язковий рівень підготовки учнів з цього предмета [6].

Більшість методистів дотримуються думки, що в основній школі повинен вивчатися базовий курс користувача, який забезпечить необхідні знання, вміння і навички майбутньому користувачу персонального комп'ютера. Учні повинні отримати уявлення про можливості комп'ютера, оволодіти первинними навичками практичного використання комп'ютера для обробки текстів та графічних зображень, для зберігання та пошуку інформації, для обробки великих числових масивів інформації. Базовий курс інформатики може вивчатися за рахунок варіативної складової навчального плану. Вивчення переважно здійснюється за програмою курсу інформатики для 7 - 9 класів [7].

Метою курсу є формування теоретичної бази знань учнів з основ інформатики та практичних навичок використання засобів сучасних інформаційно-комунікаційних технологій у повсякденній практичній, зокрема навчально-пізнавальній, діяльності учнів, започаткування основ їхньої інформаційної культури. Програмою передбачається ознайомлення учнів з основними поняттями інформології, будовою інформаційної системи,

основними видами програмного забезпечення, основами алгоритмізації та програмування на основі використання систем візуального програмування. У 7-11 класах за рахунок варіативної частини навчального плану та за рахунок годин трудового навчання може також вивчатися базовий курс інформатики “Інформатика та технології” [12].

Метою базового курсу інформатики є набуття учнями теоретичних знань з основ інформатики та практичних навичок використання сучасних інформаційних технологій у повсякденній практичній, зокрема навчально-пізнавальній діяльності. У курсі “Технології” у 7-9 класах здійснюється передпрофільне навчання учнів з метою найбільш раннього вибору ними галузі майбутньої професійної діяльності. Програма має на меті розвиток в учнів умінь і навичок виконання операцій зі створення та обробки векторної та растрової графіки, що може використовуватися для: поліграфічних робіт, мультимедійного програмування, створення web-документів [7].

При вивченні базового курсу інформатики передбачено вивчення і розділу алгоритмізації та програмування у 9-му класі з метою надання можливості вчасно здійснювати ґрунтовну та якісну підготовку учнів до участі в олімпіадах, конкурсах, турнірах, науково-практичних конференціях тощо.

Особлива увага на цьому етапі надається формуванню логічного (математичного, алгоритмічного, операційного) мислення учнів, розвитку їх творчого потенціалу. В основній школі (з 8 класу) також розпочинається вивчення інформатики у класах з поглибленим вивченням математики чи Інформатики.

При тематичному плануванні вивчення базового курсу інформатики слід пам’ятати, що в деяких випадках у учнів немає ще достатніх знань з інших предметів (математика, фізика), тому такі теми слід залишати для вивчення в старших класах [3].

1.4.2. Програми з інформатики

Навчальну програму з інформатики підготовлено у 2015–2016 рр. робочою групою у складі: *М. І. Жалдак* (завідувач кафедри теоретичних основ інформатики Національного педагогічного університету ім. М. П. Драгоманова, академік НАПН України, доктор педагогічних наук, професор, голова робочої групи), *Ю. В. Горошко* (завідувач кафедри інформатики і обчислювальної техніки Чернігівського національного педагогічного університету ім. Т. Г. Шевченка, доктор педагогічних наук, доцент), *О. В. Коршунова* (завідувач сектору цифрової грамотності відділу неформальної та інформальної освіти для дорослих Інституту модернізації змісту освіти), *Б. В. Кудренко* (головний спеціаліст Міністерства освіти і науки України), *Н. В. Морзе* (проректор з інформатизації навчально-наукової та управлінської діяльності Київського університету імені Бориса Грінченка, доктор педагогічних наук, професор, член-кореспондент НАПН України), *Є. В. Мотурнак* (завідувач лабораторії навчально-виховного комплексу-ліцею № 100 м. Дніпра, заслужений вчитель України), *Т. В. Нанаєва* (директор з корпоративних справ INTEL в Україні та СНД), *Г. О. Проценко* (перший заступник начальника управління освіти Печерської районної в місті Києві державної адміністрації, вчитель вищої категорії, вчитель-методист, кандидат педагогічних наук), *Й. Я. Ривкінд* (вчитель інформатики ліцею № 38 імені Молчанова м. Києва, заслужений вчитель України), *В. В. Шакоцько* (заступник директора Кременчуцького педагогічного училища імені А. С. Макаренка Полтавської області, викладач-методист), *І. О. Завадський* (доцент кафедри математичної інформатики факультету кібернетики Київського національного університету імені Тараса Шевченка, кандидат фіз.-мат. наук), *В. В. Лапінський* (провідний науковий співробітник відділу математичної та інформативної освіти Інституту педагогіки НАПН України), *А. В. Паньков* (старший науковий співробітник сектору змісту підручників та навчальних матеріалів відділу загальної середньої освіти та підготовки вчителів Інституту модернізації змісту освіти) [14].

Склад робочої групи з оновлення навчальної програми з інформатики для учнів 5–9 класів (2017 рік):

І. О. Завадський (доцент кафедри математичної інформатики факультету кібернетики Київського національного університету імені Тараса Шевченка, голова групи, кандидат фіз.-мат. наук), *О. В. Пасічник* (учитель інформатики НВК «Школа-гімназія «Сихівська» м. Львова, керівник проектів та програм Українського католицького університету), *Н. А. Саражинська* (учитель інформатики Білоцерківської спеціалізованої школи I–III ст. № 12), *О. О. Богатирьов* (доцент кафедри прикладної математики та інформатики Черкаського національного університету ім. Б. Хмельницького, кандидат фіз.-мат. наук), *С. М. Бондаренко* (учитель інформатики Прилуцької ЗОШ I–III ст. № 7), *Л. В. Булигіна* (учитель інформатики Політехнічного ліцею НТУУ «КПІ»), *Г. Ю. Громко* (учитель інформатики Нечаївської загальноосвітньої школи I–III ст. ім. Ю. І. Яновського), *О. Б. Коротка* (учитель інформатики Запорізького колегіуму «Елінт», учитель-методист), *Б. В. Кудренко* (головний спеціаліст департаменту загальної середньої та дошкільної освіти Міністерства освіти і науки України), *В. В. Лапінський* (провідний науковий співробітник відділу математичної та інформатичної освіти Інституту педагогіки НАПН України, кандидат фіз.-мат. наук, доцент), *Л. В. Палюшок* (завідувач кабінету інформаційних технологій Львівського обласного інституту післядипломної педагогічної освіти), *Л. М. Федор* (учитель інформатики Чернівецької гімназії № 5) [10].

Пояснювальна записка

Метою базової загальної середньої освіти є розвиток і соціалізація особистості учнів, формування їхньої національної самосвідомості, загальної культури, світоглядних орієнтирів, екологічного стилю мислення і поведінки, творчих здібностей, дослідницьких і життєзабезпечувальних навичок, здатності до саморозвитку й самонавчання в умовах глобальних змін і викликів.

Випускник основної школи — це патріот України, який знає її історію; носій української культури, який поважає культуру інших народів; компетентний мовець, що вільно спілкується державною мовою, володіє також рідною (у разі відмінності) й однією чи кількома іноземними мовами, має бажання і здатність до самоосвіти, виявляє активність і відповідальність у громадському й особистому житті, здатний до підприємливості й ініціативності, має уявлення про світобудову, бережно ставиться до природи, безпечно й доцільно використовує досягнення науки і техніки, дотримується здорового способу життя [11].

Мета базової загальної середньої освіти досягається шляхом реалізації таких завдань інформатичної освіти:

- визначати й формулювати у різноманітних життєвих ситуаціях задачі, для розв'язання яких можна залучити цифрові пристрої та інформаційні технології;
- знаходити, подавати, перетворювати, аналізувати, узагальнювати та систематизувати дані, необхідні для розв'язання життєвих задач;
- застосовувати алгоритмічний та системний підходи, створювати та аналізувати інформаційні моделі для ефективного розв'язання задач, що постають у житті, навчальній та професійній діяльності;
- вільно, відповідально й безпечно використовувати сучасні інформаційні технології та цифрові пристрої, а також самостійно опанувати нові;
- створювати інформаційні продукти, працюючи індивідуально або в команді;
- критично оцінювати інформацію та її вплив на людину і суспільство, переваги та ризики використання ІТ для себе, суспільства й довкілля;
- усвідомлювати етичні, суспільні, культурні та правові норми й дотримуватися їх під час роботи з інформацією та використання інформаційних технологій.

В основу навчального курсу «Інформатика» для 5–9 класів покладено *розвивально-компетентнісний підхід*, що передбачає формування предметних та ключових компетентностей, а також розвиток певних мисленнєвих навичок. Предметні компетентності формуються завдяки виконанню перелічених вище завдань. Роль курсу інформатики у формуванні ключових компетентностей відображено в табл. 1. [19].

Таблиця 1. Ключові компетентності в курсі інформатики

	Ключові компетентності	Компоненти
	Спілкування державною (і рідною у разі відмінності) мовами	<p><i>Уміння:</i></p> <p>створювати інформаційні продукти та грамотно і безпечно комунікувати з використанням сучасних технологій державною (і рідною у разі відмінності) мовою; висловлюватись та спілкуватися на тему сучасних інформаційних технологій з використанням відповідної термінології.</p> <p><i>Ставлення:</i></p> <p>усвідомлення комунікаційної ролі ІТ; уникнення невнормованих іншомовних запозичень у спілкуванні на ІТ-тематику; надавання переваги використанню програмних засобів та ресурсів з інтерфейсом державною (і рідною у разі відмінності) мовами</p>
	Спілкування іноземними мовами	<p><i>Уміння:</i></p> <p>використовувати програмні засоби та ресурси з інтерфейсом іноземними мовами; використовувати програмні засоби для перекладу текстів та тлумачення іноземних слів; оперувати базовою міжнародною ІТ-термінологією.</p>

		<p><i>Ставлення:</i></p> <p>усвідомлення ролі ІТ в інтерперсональній комунікації у глобальному контексті; розуміння необхідності володіння іноземними мовами для онлайн-навчання й активного залучення до європейської та глобальної спільнот, усвідомлення своєї причетності до них</p>
Математична компетентність		<p><i>Уміння:</i></p> <p>розуміти, використовувати та створювати математичні моделі об'єктів та процесів для розв'язування задач із різних предметних галузей засобами інформаційних технологій.</p> <p><i>Ставлення:</i></p> <p>усвідомлення ролі математики як однієї з основ ІТ</p>
Основні компетентності у природничих науках і технологіях		<p><i>Уміння:</i></p> <p>застосовувати логічне, алгоритмічне, структурне та системне мислення для розв'язування життєвих та проблемних ситуацій; планувати та проводити навчальні дослідження та комп'ютерні експерименти в галузі природничих наук і технологій;</p> <p>послугуватися технологічними пристроями.</p> <p><i>Ставлення:</i></p> <p>усвідомлення міждисциплінарного значення інформатики; усвідомлення ролі наукових ідей в сучасних інформаційних технологіях</p>
Інформаційно-цифрова компетентність		Розкривається у змісті предмета

<p>Уміння вчитися впродовж життя</p>	<p><i>Уміння:</i></p> <p>організувати свою діяльність з використанням програмних засобів для планування та структурування роботи, а також співпраці з членами соціуму; самостійно опанувати нові технології та засоби діяльності.</p> <p><i>Ставлення:</i></p> <p>виявлення допитливості, наполегливості, впевненості, вміння мотивувати себе до навчальної діяльності, долати перешкоди як ключові чинники успіху навчально-пізнавального процесу інформатики; усвідомлення необхідності та принципів навчання протягом усього життя; усвідомлення відповідальності за власне навчання</p>
<p>Ініціативність і підприємливість</p>	<p>Розкривається через наскрізну змістову лінію</p>
<p>Соціальна та громадянська компетентності</p>	<p>Розкривається через наскрізну змістову лінію</p>
<p>Обізнаність та самовираження у сфері культури</p>	<p><i>Уміння:</i></p> <p>грамотно і логічно висловлювати свою думку, аргументувати та вести діалог, враховуючи національні та культурні особливості співрозмовників та дотримуючись етики спілкування і взаємодії у віртуальному просторі; враховувати художньо-естетичну складову при створенні інформаційних продуктів (сайтів, малюнків, текстів тощо).</p> <p><i>Ставлення:</i></p>

		культурна самоідентифікація, повага до культурного розмаїття у глобальному інформаційному суспільстві; усвідомлення впливу інформатики та інформаційних технологій на людську культуру та розвиток суспільства
	Екологічна грамотність і здорове життя	Розкривається через наскрізну змістову лінію

Завдяки розвивальному компоненту курс інформатики має розвивати в учнів аналітичне, синтетичне, логічне й критичне мислення, творчі здібності, естетичний смак, толерантність та повагу до чужого інтелектуального продукту, здатність аналізувати різноманітні процеси та явища й з'ясовувати їхні причинно-наслідкові та структурні зв'язки. Хоча розвиток зазначених здатностей і мисленнєвих навичок не є винятково завданням навчання інформатики, а відбувається не меншою мірою під час вивчення інших навчальних предметів, саме в процесі навчання інформатики закладаються основи таких умінь:

- визначати послідовність дій, які необхідно виконати для розв'язування певних задач, тобто розробляти *алгоритми*;
- подавати алгоритми в певному формальному вигляді та виконувати їх;
- використовувати алгоритмічні структури;
- застосовувати алгоритми для опрацювання різнотипних повідомлень;
- добирати якомога ефективніший алгоритм розв'язування задачі (на зазначених умінях базується *алгоритмічне мислення*);
- визначати параметри об'єктів та їх можливі значення;
- класифікувати явища та об'єкти;

- знаходити структурні зв'язки між класами об'єктів, класифікувати знайдені зв'язки;
- подавати дані в табличному та графічному вигляді, інтерпретувати дані, подані графічно;
- формулювати задачі з опрацювання структур даних і формалізувати їх з метою подальшого автоматизованого розв'язування з використанням ІКТ-засобів (зазначені вміння є основою *структурного мислення*) [22].

1.4.3. Особливості методики вивчення векторної графіки у школі

В першу чергу потрібно враховувати, що вивчення теми «Комп'ютерна графіка» розпочинається ще в початковій школі, в середній школі діти вже більш детально ознайомлюються з даною темою, тому в старшій школі необхідно поглиблювати їхні знання, а не просто повторювати раніше вивчений матеріал.

Оскільки ця тема є однією з вибірових, необхідно правильно розподіляти час на її вивчення, адже згідно з навчальною програмою (рівень стандарту) на вивчення вибірових модулів рекомендується відводити до 35 годин. Потрібно враховувати, що ми не можемо створювати додаткового навантаження поза програмою. Зважаючи на це необхідно більше часу приділяти освоєнню практичних навичок і меншу частину відводити на вивчення та повторення теоретичного матеріалу. Теоретичний матеріал варто подавати частково на початку заняття, перед практичною роботою задля кращого його засвоєння, адже, як відомо, якщо теоретичний матеріал відразу не закріпити практичними завданнями, то потім його досить складно засвоїти.

Власне всю роботу варто будувати на фронтально-індивідуальному виконанні вправ та завдань зі створення та обробки графічних зображень за комп'ютером.

Варто також враховувати, що не всі мають творчий хист, а його розвиток є лише одним із завдань цього курсу. Необхідно підбирати такі завдання, які зможуть виконати всі діти, навіть ті в кого погано розвинені творчі здібності.

Потрібно створювати різнорівневі завдання для диференціації навчання. Слід розробляти завдання, які спрямовані на оволодіння технічними прийомами роботи в графічних редакторах та на пояснення особливостей роботи обраних програм.

Якщо ми спостерігаємо високу зацікавленість учнів в опануванні навичок роботи з різними графічними редакторами можна проводити фронтальні лабораторні роботи, які передбачають роботу школярів в різних графічних редакторах і в різному темпі, такий підхід у деяких випадках є більш продуктивним, але й більш складним у реалізації.

Слід враховувати, що процес навчання в принципі не може бути реалізований тривалий час без інтелектуального контакту між учнем і вчителем. Щоб забезпечити такий контакт вчитель має виконувати загальні методичні вимоги: проводити роз'яснення задач так, щоб вони стали їх особистим завданням; порушувати інтерес учнів, мобілізуючи їхні пізнавальні зусилля і, перш за все, їхню увагу; обговорювати з учнями способи розв'язання завдання, розробляти гіпотези і шляхи їх перевірки; відновити в пам'яті пізнавальний попередній пізнавальний досвід для необхідний для знання нового знання; звертати увагу учнів у потрібних на на головні об'єкти ставити ставити додаткові запитання і, необхідно необхідно, обговорювати їх.

Доцільно проводити самостійні роботи, адже в процесі виконання такого виду роботи в дітей забезпечується активна пізнавальна діяльність, при цьому вони свідомо прагнуть досягти поставленої в завданні мети. Самостійна робота розвиває пізнавальні здібності учнів, сприяє виробленню практичних умінь і навичок, підвищує культуру розумової праці, робить здобуті знання більш осмисленими і глибокими.

Для отримання кращих результатів навчання і поглиблення знань учнів можна застосовувати метод проектів. Застосування цього методу сприяє розвитку пізнавальних навичок учнів, умінь самостійно конструювати свої знання, умінь орієнтуватися в інформаційному розвитку, розвиток критичного мислення [3].

1.5. Гурткова робота графіки як з комп'ютерної графіки як різновид неформальної освіти

1.5.1 Особливості гурткової роботи в школі

Багато дослідників вказують на те, що позакласна робота в сучасній школі, органічно пов'язана з навчальною діяльністю, але на відміну від неї, будується за принципом добровільності, а її зміст має відповідати інтересам школярів. Тому добровільність вибору виду позакласних занять дає можливість всебічно враховувати інтереси і запити школярів, їх загальну підготовку, індивідуальні нахили. У своїй книзі "Методика трудового навчання" Мельникова Л.В. ставить перед позакласною роботою наступні завдання: закріплення, узагальнення і розширення знань і умінь включення учнів у творчу діяльність розширення уявлень про основи сучасного виробництва поглиблена підготовка до вибору професії

Вирішуючи ті ж, що і навчання, позакласна робота значно розширює можливості трудової підготовки школярів. Разом з тим при підборі тематики гурткових занять, тематичних вечорів і т.п. не можна виходити тільки з бажань учнів. Учитель повинен активно впливати на учнів, пробуджувати і виховувати в них інтерес до перспективних видів техніки, нових тенденцій в мистецтві, враховуючи при цьому майбутню професійну підготовку більшості школярів. Обов'язковою умовою успіху позакласної роботи, а також її особливістю є велика, ніж в навчальному процесі самодіяльність і активність учнів.

Кудрявцев Т.В. в книзі «Психологія технічної творчості» підкреслює, що в технічній творчості немає проблеми застосування знань, тому що вона стоїть завжди, само собою зрозуміло. Зазначені завдання злиті воедино в часі, бо технічні поняття, за змістом своїм націлені на практичне виконання і мають прикладний характер. Це можна застосувати до всієї позакласній роботі. Вона

носить, як правило продуктивний характер. При цьому навчальні вправи підбираються з урахуванням трудових умінь [12].

У своїй книзі "Технічна творчість учнів" Столяров Ю.С. зазначає, в процесі позакласної роботи розширюються можливості політехнічної освіти учнів. Оскільки зміст позакласної роботи не пов'язане жорстко з навчальними програмами за технологією, то можуть бути заплановані екскурсії на будь-які підприємства, прийнятні з точки зору безпеки, для ознайомлення школярів з основами сучасного виробництва. Посилення політехнічної спрямованості навчання може стати основною метою позакласної роботи. Приблизно те ж саме можна сказати про можливості профорієнтаційної роботи в процесі позакласної роботи. Цьому сприяє проведення зустрічей з представниками різних професій [24].

Міщерева Є.І. в своїй книзі "Методика трудового навчання" зазначає, що потрібно надавати праці учнів у позакласній роботі виховує характер. Виховання комунікабельності, самостійності, акуратності, працьовитості, почуття критичної самооцінки, дисциплінованості, прагнення до глибокого і всебічного сприйняття краси - одне із завдань позакласної роботи. В процесі виконання трудових завдань учні повинні допомагати один одному, обмінюватися думками, порадами. Перевіряючи виготовлений виріб, вчитель повинен запропонувати учню спробувати самому оцінити його красу і якість. Учитель повинен систематично підкреслювати естетичність добре виготовленого виробу, якість обробки, витонченість форм виробів і чіткість дій учнів [7].

Міщерева Є.І., Мельникова Л.В. виділяють наступні форми позакласної роботи школярів на базі уроків технології: гуртки, клуби за інтересами, екскурсії та інші заходи.

Гурток - випробувана форма позакласної роботи, яка широко застосовується в школах на базі різних навчальних предметів. Незалежно від профілю в роботі гуртків є багато спільного. Тому керівникам гуртків слід

глибоко вивчати досвід гуртків інших профілів і застосовувати в своїй роботі все краще. У практиці робіт шкіл зустрічаються різноманітні гуртки [10, 15].

З точки зору зв'язку з навчальними предметами можна виділити три групи гуртків: предметні, міжпредметні і непередметні.

Предметними називаються такі гуртки, які пов'язані безпосередньо з трудовим навчанням. За своїм змістом вони є як би продовженням того, що робить учень на уроках технології, однак їх діяльність стає більш складною, глибокою. Об'єкти праці в предметних гуртках ні в якому разі не повинні дублювати ті, які учні виконували в навчальний час. Серед міжпредметних гуртків найбільш позитивно зарекомендували себе фізико-технічні гуртки. Непредметні гуртки за змістом діяльності учнів можуть бути найрізноманітнішими. До непередметних відносяться перш за все спортивно-технічні гуртки (авто-, авіа-, судно-, ракетомодельного та ін.). Працюючи в цих гуртках, учні набувають багато нових знань і умінь, які не мають прямого відношення до обробки матеріалів та їх властивостями. До цієї ж групи гуртків відносяться і такі, які знайомлять учнів з деякими новими видами трудової діяльності: карбування металів, художня обробка деревини.

1.5.2. Робоча програма гуртка з інформатики «Комп'ютерна графіка» для учнів базової середньої школи

Область інформатики, що займається методами створення і редагування зображень за допомогою комп'ютерної техніки називається комп'ютерною графікою.

Люди різних професій застосовують комп'ютерну графіку в своїй роботі. Це - дослідники в різних наукових і прикладних областях, художники, конструктори, фахівці з комп'ютерної верстки, дизайнери, розробники рекламної продукції, творці Web-сторінок, автори мультимедіа-презентацій, медики, модельєри тканин і одягу, фотографи та інші.

Зображення на екрані комп'ютера створюються за допомогою графічних програм. Це растрові і векторні редактори, програми створення і обробки

тривимірних об'єктів, системи автоматизації проектування, настільні видавничі системи.

Основна увага в програмі гуртка «Комп'ютерна графіка» приділяється створенню ілюстрацій і редагування зображень, тобто векторних і растрових програм.

Знання та вміння, набуті в результаті освоєння курсу «Комп'ютерна графіка», є фундаментом для подальшого вдосконалення майстерності в області тривимірного моделювання, анімації, відеомонтажу, створення систем віртуальної реальності.

Метою гуртка є навчити учнів створювати і редагувати графічні зображення, використовуючи інструменти спеціальних програм;

Завдання гуртка:

- дати глибоке розуміння принципів побудови і зберігання зображень;
- вивчити формати графічних файлів і доцільність їх використання при роботі з різними графічними програмами;
- розглянути застосування основ комп'ютерної графіки в різних графічних програмах;
- навчити виконувати обмін графічними даними між різними програмами.

У програмі гуртка «Комп'ютерна графіка» розглядаються:

- основні питання створення, редагування і зберігання зображень;
- особливості роботи з зображеннями в растрових програмах;
- методи створення ілюстрацій в векторних програмах.

При вивченні курсу «Комп'ютерна графіка» формуються такі особистісні результати:

- 1) Формування цілісного світогляду, відповідного сучасному рівню розвитку науки і суспільної практики.

2) Формування комунікативної компетентності в спілкуванні та співпраці з однолітками і дорослими в процесі освітньої, суспільно-корисної, навчально-дослідницької, творчої діяльності.

3) Формування цінності здорового і безпечного способу життя.

При вивченні курсу «Комп'ютерна графіка» формуються наступні результати:

Уміння самостійно планувати шляхи досягнення мети, в тому числі альтернативні, свідомо обирати найбільш ефективні способи вирішення навчальних пізнавальних завдань.

Уміння оцінювати правильність виконання навчального завдання, власні можливості її рішення.

Вміння визначати поняття, створювати узагальнення, встановлювати аналогії, класифікувати, встановлювати причинно-наслідкові зв'язки, будувати логічні міркування, умовиводу (індуктивні, дедуктивні і за аналогією) і робити висновки.

Уміння створювати, застосовувати і перетворювати знаки і символи, моделі і схеми для вирішення навчальних та пізнавальних завдань.

Формування і розвиток компетентності в галузі використання інформаційних технологій.

В результаті освоєння теоретичної частини програми гуртка учні повинні знати:

- види комп'ютерної графіки;
- призначення та можливості графічного редактора;
- поняття фрагмента малюнка;
- поняття файлу;
- точні способи побудови геометричних фігур;
- поняття пікселя і піктограми;
- переваги і недоліки графічних редакторів.

В результаті освоєння практичної частини програми гуртка учні повинні вміти:

- виконувати як просте, так і складне виділення об'єктів;
- працювати з шарами;
- робити колаж різної складності;
- володіти можливостями корекції та заливки;
- учні повинні впевнено і легко володіти комп'ютером;
- самостійно складати композиції;
- знати термінологію.

Очікувані результати навчання.

По закінченню навчання учні повинні демонструвати сформовані вміння і навички роботи в графічних редакторах; вміння самостійно застосовувати основні інструменти і прийоми, які використовуються в растрової комп'ютерної графіки при створенні і редагуванні графічних об'єктів; виконувати малюнки і зображення різного ступеня складності.

ВИСНОВКИ ДО РОЗДІЛУ 1

Неформальна освіта – це освіта, яка існує за межами освітнього простору, де чітко визначені завдання, методи і результати навчання. Характеризується наступними рисами:

- орієнтація на запити освітні груп різних груп населення;
- увага до увага потреб освітніх потреб осіб категорій осіб (інвалідів, мігрантів); високий

- високий активності персональної активності учнів примусового відсутність примусового характеру на підстави на власній мотивації; внутрішня

- внутрішня за результат за результат освітньої високий , сенс особистісний сенс навчання;

- якостей розвиток якостей особистості, сприятливі забезпечують для передумови для гідного життя також а також успішної суспільній в суспільній і трудовій діяльності; розвиток

- розвиток мобільності умовах мінливих умовах сучасного світу; гнучкість
- гнучкість та організації та методах організація ; організація відносин освітнього учасниками на процесу на взаємній участі, участі, демократичній культурі По

По учні навчання учні сформовані демонструвати сформовані вміння роботи навички роботи в графічних вміння ; вміння основні застосовувати основні інструменти і прийоми, які використовуються в графіки при графіки при створенні графічних редагуванні графічних об'єктів; виконувати малюнки і ступеня різного ступеня складності. освіта

Неформальна освіта в векторної комп'ютерної векторної графіки цікавою неймовірно може дати може дати на учням наукове блискуче наукове майбутнє.

РОЗДІЛ 2 АНАЛІЗ ДЛЯ МАТЕРІАЛІВ ДЛЯ ГРАФІКИ ВЕКТОРНОЇ ГРАФІКИ ЗАСОБАМИ СЕРЕДОВИЩІ В СЕРЕДОВИЩІ VISUAL STUDIO 2010

2.1. Що таке OpenGL, його особливості та значення в сфері комп'ютерної графіки

2.1.1. Бібліотеки, можливості та можливості та функції

Зараз векторні зображення можна побачити скрізь, починаючи від комп'ютерних ігор і закінчуючи системами моделювання в реальному часі. Раніше, коли векторна графіка існувала тільки на суперкомп'ютерах, не існувало єдиного стандарту в галузі графіки. Всі програми писалися з "нуля" або з використанням накопиченого досвіду, але в кожній програмі реалізовувалися свої методи для відображення графічної інформації. З приходом потужних процесорів і графічних прискорювачів тривимірна графіка стала реальністю для персональних комп'ютерів. Але в той же час виробники програмного забезпечення зіткнулися з серйозною проблемою - це відсутність будь-яких стандартів, які дозволяли писати програми, незалежні від обладнання і операційної системи. Одним з перших таких стандартів, що стоїть і донині є OpenGL [31].

Дана специфікація - це графічний стандарт в області комп'ютерної графіки.

На даний момент він є одним з найпопулярніших графічних стандартів у всьому світі. Ще в 1982 р в Стенфордському університеті була розроблена концепція графічної машини, на основі якої фірма Silicon Graphics в своїй робочій станції Silicon IRIS реалізувала конвеєр рендеринга. Таким чином, була розроблена графічна бібліотека IRIS GL. На основі бібліотеки IRIS GL, в 1992 році був розроблений і затверджений графічний стандарт OpenGL. Розробники цього стандарту - це найбільші фірми розробники як обладнання так і програмного забезпечення: Silicon Graphics, Inc., Microsoft, IBM Corporation, Sun Microsystems, Inc., Digital Equipment Corporation (DEC), Evans & Sutherland, Hewlett-Packard Corporation, Intel Corporation та Intergraph Corporation [25].

OpenGL перекладається як Відкрита Графічна Бібліотека (Open Graphics Library), це означає, що це відкритий і мобільний стандарт. Використовується при створенні комп'ютерних ігор, САПР, віртуальної реальності, візуалізації в наукових дослідженнях.

Програми, написані за допомогою цієї системи можна переносити практично на будь-які платформи, отримуючи при цьому однаковий результат, будь це графічна станція або суперкомп'ютер. Даний продукт звільняє програміста від написання програм для конкретного обладнання. Якщо пристрій підтримує якусь функцію, то ця функція виконується апаратно, якщо немає, то бібліотека виконує її програмно.

Що ж представляє із себе OpenGL? З точки зору програміста це незалежний від мови програмування крос-платформний програмний інтерфейс (API) для графічних пристроїв, таких як графічні прискорювачі. Він включає в себе близько 250 різних команд, за допомогою яких програміст може визначати різні об'єкти і виробляти рендерингАбревіатура [16].

Абревіатура API Programming Application Programming Interface, що як , як програмуванняприкладного програмування. Різні 3D API (Direct3D складова складова частина DirectXOpenGLOpenGL, Glide, Metal і багато інших) призначені в основному для уніфікації та полегшення розробки додатків, що використовують тривимірну графіку і ігри.

API дають нам уніфікацію всіх моделей відеокарт, всіх брендових виробників, апаратну підтримку багатьох речей, зручну (для програміста) емуляцію програмними засобами те, що не підтримується апаратно, і ще багато чого іншого.

Таким чином, у програміста з'являється набагато більше часу, яке він може витратити, скажімо, на програмування геймплея гри, ну або AI (Artificial Intelligence, штучний інтелект).

Для підтвердження незалежності від мови програмування були розроблені різні варіанти прив'язки (binding) функцій OpenGL або повністю перенесені на інші мови. Одним із прикладів може служити бібліотека Java 3D, яка може

використовувати апаратне прискорення OpenGL. Пряма прив'язка функцій реалізована в Lightweight Java Game Library, яка має пряму прив'язку для Java. Sun також випустила версію Java OpenGL (JOGL), яка надає пряму прив'язку до C-функцій, на відміну від Java 3D, яка не має настільки низкорівневої підтримки. Офіційний сайт має посилання на прив'язки для мов Java, Python, і Pascal. Є також варіанти прив'язки OpenGL для мов C++ і C# [21].

Для досягнення апаратної незалежності специфікації, команди для керування вікнами (менеджер вікон), так само як і команди для роботи з пристроями введення (клавіатури, миші), були виключені. Це може здатися серйозним недоліком при використанні OpenGL, але є можливість скомбінувати його з іншими гнучкими бібліотеками, які будуть дбати про завдання управління вікнами і про отримання вхідних даних від користувача. Більш того, цей стандарт не надає ніяких команд для опису складних моделей (молекул, літаків, будівель, птахів і т.д.). У OpenGL знаходяться тільки найбільш примітивні геометричні об'єкти (точки, лінії і багатокутники). Розробнику надається самому сконструювати свої власні моделі, ґрунтуючись на цих кількох простих примітивах. Говорячи більш простою мовою, ви визначаєте об'єкти, задаєте їх розташування в тривимірному просторі, визначаєте інші параметри (поворот, масштаб), задаєте властивості об'єктів (колір, текстура, матеріал), положення спостерігача, а бібліотека подбає про те щоб відобразити все це на екрані.

Є також належні до бібліотеки, які надають більш складні моделі, і кожен може використовувати ці бібліотеки для побудови власних.

Система має добре продуману внутрішню структуру і досить простий процедурний інтерфейс. Незважаючи на це за допомогою OpenGL можна створювати складні і потужні програмні комплекси, витрачаючи при цьому мінімальний час в порівнянні з іншими графічними бібліотеками [15].

У деяких бібліотеках (наприклад під X Windows) є можливість зображати результат не тільки на локальній машині, але також і по мережі. Додаток, який виробляє команди називається клієнтом, а додаток, яке отримує ці команди і

відображає (рендерить) результат - сервером. Спеціальний протокол передає команди між сервером і клієнтом. Завдяки незалежності від ОС, сервер і клієнт можуть працювати на різних платформах. Досить часто сервером є суперкомп'ютер, який виконує складне моделювання, а клієнтом - проста робоча станція, головним чином зайнята візуалізацією графіки. Таким чином можна будувати дуже потужні відтворюють комплекси на основі декількох робочих станцій або серверів, з'єднаних мережею [32].

За наступні 17 років версія бібліотеки досягла позначки 3.1. Але бібліотека не те щоб видозмінювалася - немає. Її творці заклали в неї механізм розширень, завдяки якому виробники апаратного забезпечення (виробники відеокарт) могли випускати розширення OpenGL для підтримки нових специфічних можливостей, які не включені в поточну версію бібліотеки. Завдяки цьому, програмісти могли відразу використовувати ці нові можливості, на відміну від бібліотеки Microsoft Direct3D. У цьому випадку їм би довелося чекати виходу нової версії DirectX.

У сучасному світі, бібліотеки OpenGL і DirectX є конкурентами на платформі MS Windows. Microsoft всіма силами просуває свою бібліотеку, а так само всіма силами прагнуть уповільнити розвиток бібліотеки OpenGL, що послабило б графічну систему конкуруючих ОС, де використовується виключно бібліотека OpenGL для реалізації виведення всієї графіки.на

тена те, що бібліотека OpenGL скорочено скорочено GL - Graphics Library надає практично практично всі для для моделювання і тривимірних сцен, деякі з функцій, які потрібні при роботі з графікою, відсутні в стандартній бібліотеці. Наприклад, щоб задати положення і напрямок камери, з якої буде спостерігатися сцена, потрібно самому розраховувати модельну матрицю, а це далеко не всі вміють. Тому для OpenGL існують так звані допоміжні бібліотеки.

Перша з цих бібліотек *GLU* - головна додаткова бібліотека (Graphics Library Utilities). Ця бібліотека вже стала стандартом і поставляється разом з головною бібліотекою OpenGL. До складу цієї бібліотеки увійшли більш

складні функції, наприклад для того щоб визначити циліндр або диск буде потрібно всього одна команда. Також в бібліотеку увійшли функції для роботи зі сплайнами, реалізовані додаткові операції над матрицями та додаткові види проєкцій.

Бібліотека *GLUT* (Graphics Library Utilities Toolkit) – це також незалежна від платформи бібліотека. Вона реалізує не тільки додаткові функції OpenGL, але і надає функції для роботи з вікнами, клавіатурою і мишкою. Для того щоб працювати із стандартом в конкретній операційній системі (наприклад Windows або X Windows), треба провести деяке попереднє налаштування, яке залежить від конкретної операційної системи. З бібліотекою GLUT все набагато спрощується, буквально декількома командами можна визначити вікно, в якому буде працювати OpenGL, визначити переривання від клавіатури або мишки і все це не буде залежати від операційної системи. Бібліотека надає також деякі функції, за допомогою яких можна визначати деякі складні фігури, такі як конуси, тетраедри, і навіть можна за допомогою однієї команди визначити чайник! Але одним з її недоліків є те, що її не можна використовувати в комерційних проєктах.

Free GLUT - Free Graphics Library Utilities Toolkit, є безкоштовною Open Source альтернативою GLUT. Вона повністю повторює функціональність GLUT, але автори і вихідні коди у них абсолютно різні.

Є ще одна бібліотека схожа на GLUT, називається вона GLAUX (Auxiliary Graphics Library). Це бібліотека розроблена фірмою Microsoft для операційної системи Windows, тому вона є (і може використовуватися) тільки в Visual Studio будь-якої версії. Вона багато в чому схожа з бібліотекою GLUT, але трохи відстає від неї за своїми можливостями. І ще один недолік полягає в тому, що бібліотека GLAUX призначена тільки для Windows, в той час як GLUT підтримує дуже багато операційних систем.

Існують і інші додаткові бібліотеки для OpenGL. Всі вони додають щось своє або орієнтовані на якусь платформу. Наприклад існує така бібліотека як GLX - це розширення для X Windows, що дозволяє використовувати в X

Windows OpenGL. GLX надає не тільки локальний рендеринг, а й рендеринг по мережі.

Бібліотеки, як GLEW (The OpenGL Extension Wrangler Library) і GLEE (The OpenGL Easy Extension library) створені для полегшення роботи з розширеннями і різними версіями OpenGL. Це особливо актуально для програмістів в Windows, так як, заголовні і бібліотечні файли, що поставляються з Visual Studio, знаходяться на рівні версії OpenGL 1.1.

OpenGL має тільки набір геометричних примітивів (точки, лінії, багатокутники) з яких створюються всі тривимірні об'єкти. Часом подібний рівень деталізації не завжди зручний при створенні сцен. Тому поверх OpenGL були створені більш високорівневі бібліотеки, такі як Open Inventor і VTK. Дані бібліотеки дозволяють оперувати більш складними тривимірними об'єктами, що полегшує і прискорює створення тривимірної сцени [9].

GLM (OpenGL Mathematics) - допоміжна бібліотека, що надає програмістам на C ++ класи і функції для виконання математичних операцій. Бібліотека може використовуватися при створенні 3D-програм з використанням OpenGL. [23] Однією з характеристик GLM є те, що реалізація заснована на специфікації GLSL. Вихідний код GLM використовує ліцензію MIT[8].

Не вдаючись глибоко в подробиці, нижче наведені деякі особливості, реалізовані даної системи:

1. Геометричні примітиви дозволяють конструювати математичний опис об'єкта. В даний час примітивами є: точки, лінії, багатокутники, зображення і бітмапи (побітові карти).
2. Кольорове кодування в форматі RGBA (Red-Green-Blue-Alpha) або в режимі індексованих кольорів.
3. Перегляд і моделювання дозволяють розміщувати об'єкти на тривимірній сцені, пересувати камери навколо цього об'єкта і вибирати потрібну точку перегляду при рендеринге сцени.
4. Текстура допомагають привнести реалізм в моделі за допомогою рендеринга зображень реально виглядають поверхонь поверх багатокутників моделі.

5. Освітлення матеріалів - невід'ємна частина всієї тривимірної графіки. OpenGL надає в розпорядження команди для підрахунку кольору будь-якої точки при заданих властивостях матеріалу і джерел світла в приміщенні.
6. Подвійна буферизація дозволяє усунути миготіння при анімації. Кожне наступне зображення в анімаційному ряду будується в окремому буфері пам'яті і по тільки по завершенні рендеринга.
7. Anti-aliasing згладжує нерівні краї ліній, що на відображаються на комп'ютерному дисплеї. Ламані лінії часто з'являються при низькому графічному вирішенні. Anti-aliasing є стандартною технікою в комп'ютерній графіці і полягає в зміні кольору та інтенсивності точок поблизу лінії для зменшення штучних зигзагів.
8. Gouraud тіні - це техніка застосування згладжених тіней тривимірного об'єкту для відмінностей тонких на квітів на деякій поверхні.
9. Z-буферизація відстежує тривимірною координату тривимірного об'єкту. Буфер для використовується для до наближення до об'єкта. Він є так само визначальним при поверхоньзаслоненних поверхонь.
10. Атмосферні ефекти такі як тумантуман, дим, серпанок роблять зображення, створені комп'ютером, більш реалістичними. Без атмосферних ефектів зображення часто виглядають нереально різкими і окресленими. Fog - це термін, який насправді описує алгоритм моделювання туману, диму, забруднень або просто ефект присутності повітря, надаючи глибину зображенню.
11. Alpha змішування використовує значення Alpha (значення дифузії матеріалу) як складову колірною коду RGBA, дозволяючи комбінувати колір оброблюваного фрагмента з кольором точок, які вже зберігаються в буфері. Уявіть, наприклад, отрисовку прозорого блакитного скла перед червоною коробкою. Alpha-змішування дозволяє моделювати прозорість скла таким чином, що частина коробки, яку видно крізь скло матиме бузковий відтінок.
12. Трафаретні плани обмежують отрисовку певної областю екрану.

13. Списки відображення дозволяють зберігати команди відтворення в деякому списку для подальшого рендеринга. При правильному застосуванні, списки відображення можуть помітно підвищити продуктивність рендеринга.

14. Поліномніе обчислення дають можливість використовувати нерівномірні В-сплайни. Вони допомагають малювати плавні криві через кілька контрольних точок, крім необхідності зберігати всі проміжні точки.

15. Зворотній зв'язок, виділення і вибір дають вам можливість створювати додатки, які дозволяють користувачеві вибирати область екрану або окремий об'єкт, зображений на екрані. Режим зворотного зв'язку дозволяє розробнику отримувати результати розрахунків при рендеринге.

16. Растрові примітиви (бітмапи і прямокутники з точок)

17. Точкові операції

18. Перетворення: повороти, масштабування, трансляція, тривимірні перспективи і так далі.

Багато функцій OpenGL - варіації один одного, розрізняючи лише типами даних і їх аргументами. Звичайно ж, цього не було б, якби він спочатку був зроблений для мов, що підтримують перевантаження функцій, таких як C++. Щоб не було плутанини в іменах функцій, ввели кілька домовленостей (правил), за якими будується ім'я функції стандарту.

По-перше, всі імена функцій починаються з приставки `gl`. наприклад,

```
glBegin ();
```

```
glEng ();
```

По-друге, якщо набір функцій мають однаковий сенс і розрізняються лише кількістю і типами параметрів, то ім'я таких функцій записують у вигляді:

```
glЗагальна частина_імені_функції [n] [type],
```

де *n* - кількість параметрів, *type* - тип параметрів.

Наприклад:

```
glVertex2d (1.0, 0.5); // 2d означає: 2 параметра типу GLdouble
```

```
glVertex3f (1.0f, 0.5f, 0.0f); // 3f означає: 3 параметра типу GLfloat
```

```
glColor3ub (127, 0, 255); // 3ub означає: 3 параметра типу GLubyte.
```

2.1.2. Установка та ініціалізація в Windows

Установка.

Останню версію бібліотеки OpenGL можна завжди знайти на офіційному сайті <http://www.opengl.org>. Там розміщені також інші додаткові бібліотеки для полегшення програмування додатків і величезна кількість документації.

У середовищі програмування Microsoft Visual C ++ 6.0 всі основні заголовки йдуть в комплекті, так само як і всі файли lib. DLL файли бібліотеки містяться в будь-якої версії Windows. Потрібно встановити додаткові бібліотеки, які розширюють можливості програмування.

Бібліотека GLUT не йде за замовчуванням в системі і в середовищі розробки. Вам буде потрібно знайти і завантажити останню версію за посиланням: GLUT Downloads. Скопіюйте * .dll файли в папку system, що знаходиться в директорії с Windows. Файли * .lib покладіть в папку lib Visual C ++, файл glut.h в папку Include \ gl.

Наприклад, файли можуть розташовуватися таким чином:

glut.dll і glut32.dll в папці C: \ Windows \ system

glut.lib і glut32.lib в папці C: \ Program Files \ Microsoft Visual Studio \ VC98 \ lib

glut.h в папці C: \ Program Files \ Microsoft Visual Studio \ VC98 \ include \ GL

Ініціалізація.

Для початку потрібно створити проект в Visual C ++. Якщо це для Вас важко, то Вам варто для початку вивчити C ++, а вже потім переходити на OpenGL.

Після створення нового Win32 додатка (не консольного) в Visual C ++, потрібно додати для збірки проекту бібліотеки. У меню Project / setting, вибираємо закладку LINK. У рядку "Object / Library Modules" додаємо "OpenGL32.lib GLu32.lib GLaux.lib". Потім нажмать ОК. Тепер все готово для створення програми на OpenGL

2.1.3. Архітектура та особливості синтаксису

З точки зору архітектури, графічна система OpenGL є конвеєром, що складається з декількох етапів обробки даних:

- Апроксимація кривих і поверхонь
- Обробка вершин і збірка примітивів
- Растеризация і обробка фрагментів
- Операції над пікселями
- Підготовка текстури
- Передача даних в буфер кадру

Взагалі, цей стандарт можна порівняти з кінцевим автоматом, стан якого визначається безліччю значень спеціальних змінних (їх імена зазвичай починаються з символів `GL_`) і значеннями поточної нормалі, кольору і координат текстури. Вся ця інформація буде використана при надходженні в систему координат вершини для побудови фігури, в яку вона входить. Зміна станів відбувається за допомогою команд, які оформляються як виклики функцій.

Для забезпечення інтуїтивно зрозумілих назв в OpenGL повне ім'я команди має вигляд:

type glCommand_name [1 2 3 4] [b s i f d ub us ui] [v] (type1 arg1, ..., typeN argN)

Таким чином, ім'я складається з декількох частин:

- *gl* це ім'я бібліотеки, в якій описана ця функція: для базових функцій OpenGL, функцій з бібліотек GLU, GLUT, GLAUX це *gl*, *glu*, *glut*, *aux* відповідно [30].

- *Command_name* ім'я команди
- [1 2 3 4] число аргументів команди
- [b s i f d ub us ui] тип аргументу:
 - символ *b* означає тип `GLbyte` (аналог `char` в `C \ C ++`),
 - символ *f* - тип `GLfloat` (аналог `float`),
 - символ *i* - тип `GLint` (аналог `int`) і так далі.

Повний список типів і їх опис можна подивитися у файлі `gl.h`

[V] наявність цього символу показує, що в якості параметрів функції використовується покажчик на масив значень

Символи в квадратних дужках в деяких назвах не використовуються. Наприклад, команда `glVertex2i ()` описана як базова в бібліотеці OpenGL, і використовує в якості параметрів два цілих числа, а команда `glColor3fv ()` використовує в якості параметра покажчик на масив з трьох дійсних чисел [33, 34].

2.2. Розробка практичних робіт гурткового навчання векторної графіки програмним способом

Лабораторна робота №1

Побудова примітивів за допомогою бібліотеки OpenGL

Мета роботи: вивчити основні процедури і функції бібліотеки OpenGL, набути практичних навичок побудови об'ємних геометричних фігур.

Завдання

Побудувати за допомогою бібліотеки куб в перспективній проекції, що обертається навколо свого центру. Напрямок обертання вибирає користувач.

Внизу екрана повинен виводитися текстовий рядок виду «Прізвище учня»

Контрольні питання

1. Які особливості роботи з OpenGL під операційною системою Windows?
2. Що таке видова матриця?
3. Як задати матрицю проектування?
4. Список зображень.
5. Як виводиться текст в OpenGL?

Лабораторна робота №2

Видалення невидимих граней

Мета роботи: вивчити метод не лицьових граней, освоїти на практиці видалення не лицьових граней за допомогою піраміди відсікання в будь-якій сцені.

Завдання

Написати програму, що реалізовує побудова частини сцени, що потрапляє в піраміду відсікання, пов'язаною з рухомою камерою. Відобразитися повинні тільки лицьові грані. Сцена повинна складатися як мінімум з трьох різних фігур, розташованих торкаючись один одного поруч.

Інтерфейс повинен передбачати зміну положення і напрямки камери перегляду.

Контрольні питання

1. Що таке не лицьова грань?
2. Що необхідно для обчислення вектора зовнішньої нормалі?
3. Опишіть метод побудови тіла за допомогою піраміди відсікання.
4. Опишіть метод побудови тіла за допомогою площини відсікання.

Лабораторна робота №3

Текстурування тривимірних зображень з використанням бібліотеки OpenGL

Мета роботи: вивчити основні процедури і функції при текстуруванні зображень, набути практичних навичок накладення текстури на об'ємну фігуру.

Завдання

Написати програму обертання куба, кожна грань якого текстурована, на тлі фотографічного зображення. Напрямку обертання задається користувачем. При цьому кожна грань повинна бути текстурована різними зображеннями. Потрібно продемонструвати текстурування з повторенням і без нього.

Контрольні питання

1. Дайте визначення текстури.
2. В якому форматі зберігаються зображення в OpenGL?

3. Для чого необхідно заповнювати структуру BITMAPINFO?
4. Чи можна в зображенні використовувати кілька текстур?
5. Як реалізується прозорість в OpenGL?

Лабораторна робота №4

Джерела світла і матеріали в бібліотеці «OpenGL»

Мета роботи: вивчити методи установки джерел світла і побудови тіней, набути практичних навичок отримання тіні від будь-якого об'єкта, а загальною управлінням загальною освітленістю сцени

Завдання

Написати функції, які керують напрямком світла, керують освітленістю сцени, і побудова тіні об'єкта на площині

Контрольні питання

1. Від чого залежить розрахунок загальної освітленістю об'єкту?
2. Чому джерело світла можна розглядати як три джерела, розташованих в одній точці?
3. Назвіть характеристики матеріалу.
4. Скільки джерел світла підтримує OpenGL?
5. Якою командою задаються параметри джерела світла?
6. Як побудувати тінь за допомогою OpenGL?

2.3. Реалізація власного проєкту як результат засвоєння практичних навичок з комп'ютерної графіки за допомогою OpenGL

2.3.1. Опис створеної програми

У моєму проєкті розроблено тематичну сцену **класної кімнати**, яка в свою чергу містить:

- 9 унікальних об'єктів(предметів);
- два джерела кольорового світла (не білого кольору);
- зміна положення спостерігача (поворот, переміщення) та масштабування

сцени;

-в заголовку вікна прізвище і групу.

Весь програмний код міститься в додатку А.

2.3.2. Структура сцени

Кожен об'єкт був зроблений окремою функцією для кращого розуміння та зручності у розташуванні. Для реалізації ванної кімнати було зроблено такі об'єкти/функції:

- Кімната(стіни та підлога) - room();
- Двері - door();
- Дошка - board();
- Вікно - window();
- Шафи -shelf();
- Іграшки - toys();
- Килим у вигляді шахматної дошки - carpet();
- Вчительський стіл та парті - sitting();
- Настінні лампи - lamp();

2.3.Опис . Опис основних функційglPushMatrix

- glPushMatrix() та glPopMatrix() використовуються для збереження і відновлення поточних координат.
- glTranslated(Dx, Dy, Dz) використовується для зміни положення об'єкти відносно 3-х мірної системи координат.
- glRotated(j, x₀, y₀, z₀) використовується для обертання фігури відносно осей x, y, z.
- glColor3ub(0-255, 0-255, 0-255) та glColor4ub(0-255, 0-255, 0-255, 0-255) використовуються для встановлення кольору в просторі RGB.
- auxSolidBox(width, height, depth), auxSolidSphere(r) та auxSolidOctahedron(width) використовуються для створення коробки, сфери та октаедра відповідно.

- `gluCylinder()` та `gluDisc()` використовуються для зображення 2-х вимірних циліндра та кола відповідно.
- `glBegin(GL_QUADS)` та `glVertex3d(x, y, z)` використовуються для зображення прямокутника за допомогою точок.
- `glEnable(GL_ALPHA_TEST)`, `glEnable(GL_BLEND)` та `glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)` використовуються для отримання дозволу обробляти 4-ий компонент кольору – прозорість.
- `glutIgnoreKeyRepeat(1)`, `glutSpecialUpFunc(releaseKey)` та `glutSpecialFunc(pressKey)` використовуються для обробки натискань з клавіатури.

2.3.4. Реалізація об'єктів

```

void board()
{
    glPushMatrix(); //door
    glTranslated(3.95, 0.5, -3.5);
    glRotated(90, 0, 1, 0);
    glColor3ub(139, 69, 19);
    auxSolidBox(4.5, 2.25, 0.18);
    glPopMatrix();

    glPushMatrix(); //door box up
    glTranslated(3.9, -0.6, -3.5);
    glRotated(90, 0, 1, 0);
    glColor3ub(0, 0, 0);
    auxSolidBox(4.5, 0.1, 0.3);
    glPopMatrix();
}

```

```

void display(void) {
    if (deltaMove) computePos(deltaMove);
    if (deltaAngle) computeDir(deltaAngle);
    if (deltaMoveUp) computePosUp(deltaMoveUp);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    gluLookAt(x, y, z, x + lx, y, z + lz, 0.0f, 1.0f, 0.0f);
    glEnable(GL_ALPHA_TEST);
    glEnable(GL_BLEND);
    glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
    glPushMatrix();
    glRotated(45, 0, 1, 0);
    room();
    glPopMatrix();
    glDisable(GL_ALPHA_TEST);
    glDisable(GL_BLEND);
    glutSwapBuffers();
}

```

2.3.5. Реалізація обробки натискань з клавіатури для управління камерою

```

void pressKey(int key, int xx, int yy) {
    switch (key) {
        case GLUT_KEY_LEFT:
            deltaAngle = -0.03f;
            break;
        case GLUT_KEY_RIGHT:
            deltaAngle = 0.03f;
            break;
        case GLUT_KEY_UP:
            deltaMove = 0.9f;
            break;
        case GLUT_KEY_DOWN:
            deltaMove = -0.9f;
            break;
        case GLUT_KEY_PAGE_UP:
            deltaMoveUp = 0.9f;
            break;
        case GLUT_KEY_PAGE_DOWN:
            deltaMoveUp = -0.5f;
            break;
    }
}

```

```
void releaseKey(int key, int x, int y) {  
    switch (key) {  
        case GLUT_KEY_LEFT:  
        case GLUT_KEY_RIGHT:  
            deltaAngle = 0.0f;  
            break;  
        case GLUT_KEY_UP:  
        case GLUT_KEY_DOWN:  
            deltaMove = 0;  
            break;  
        case GLUT_KEY_PAGE_UP:  
        case GLUT_KEY_PAGE_DOWN:  
            deltaMoveUp = 0.0f;  
            break;  
    }  
}
```

2.3.6. Демонстрація сцени



Рис. 1. Візуалізація сцени



Рис. 2. Вигляд сцени зліва

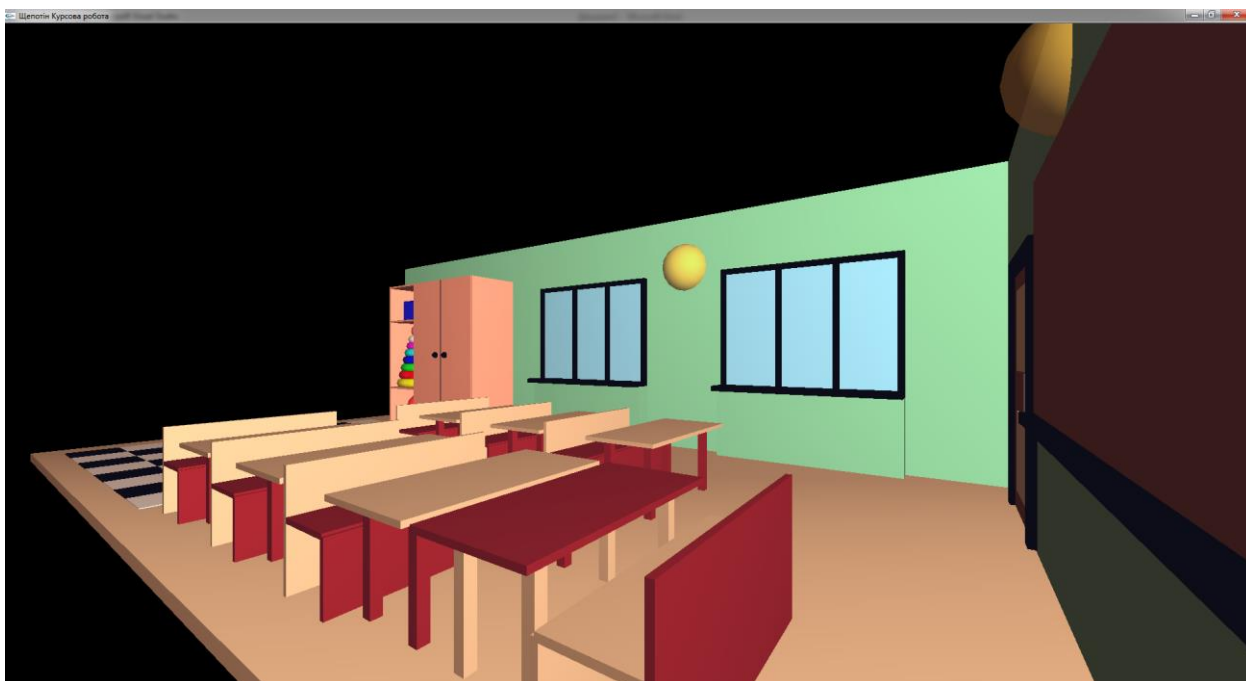


Рис. 3. Вигляд сцени справа

ВИСНОВКИ ДО РОЗДІЛУ 2

В ході роботи на розділом були розроблені лабораторні роботи та отримані практичні навички з написання програм для операційних систем сімейства Windows із застосуванням технологій векторної графіки з використанням бібліотеки OpenGL.

Таким чином, можна виділити наступні вирішені в рамках даної роботи завдання:

- вивчення принципів роботи OpenGL в віконній середовищі Windows;
- отримання практичних навичок використання даного засобу;
- отримання навичок програмування динамічних анімаційних сцен;
- отримання навичок програмування інтерактивних додатків.

Також була проведена робота з такими можливостями бібліотеки як:

- використання графічних примітивів;
- застосування освітлення;
- застосування проекції.

Розроблений додаток є повнофункціональною моделлю класної кімнати, містить в собі велику кількість вбудованих налаштувань, за допомогою яких відображаються основні можливості та переваги графічної бібліотеки OpenGL, а також інтуїтивно зрозумілий інтерфейс.

У роботу було зроблено сцену ванної кімнати за допомогою засобів OpenGL. Сцена містить 9-ть унікальних об'єктів, наявна зміни положення камери/спостерігача, наявні 2-а не білого кольору джерела світла. Я використав більшість навичок, які здобув під час виконання практичних робіт.

Сподіваюсь, що ці знання стануть фундаментом у вивченні векторної графіки та допоможуть учням створенні майбутніх проектів.

ВИСНОВКИ

На сьогоднішній день створення, розробка застосування інформаційних технологій має важливе значення для підвищення ефективності навчального процесу.

Неформальне навчання може сприяти безперервній освіті, особистісному розвитку і професійному росту людини будь-якої професії. Для його реалізації потрібно вибрати середовище, яке буде підтримано більшістю електронних носіїв і буде зручним в використанні.

Кращим програмним продуктом для вивчення векторної графіки в умовах неформальної освіти є графічний стандарт OpenGL, тому що він безкоштовний в використанні, поширюється у відкритому вихідному коді, легкий при встановленні та оновленні. Практично будь-який викладач, що володіє базовими знаннями програмної мови C++ може створити власний позашкільний курс, вести заняття і контролювати діяльність своїх учнів очно або дистанційно, в будь-який зручний для нього час.

Навчання комп'ютерній графіці в даний час популярно і поширене, так як застосовується в різних сферах життя людини, дозволяє виконувати точні геометричні побудови, креслення та іншу конструкторську документацію; допомагає в розробці рекламної продукції, а також використовується в науці та медицині.

Процес навчання векторної графіки в умовах неформальної освіти має свої особливості: викладач повинен забезпечити студента найбільшою кількістю інформації, яка буде зрозуміла і проста в засвоєнні; в процесі самостійного вивчення векторного графічного редактора в учнів виникають складнощі, які необхідно вирішити за допомогою зворотного зв'язку і підтримки індивідуального підходу; також обов'язкове розробка зрозумілого і структурованого пояснювального візуального матеріалу для найкращого засвоєння нової інформації; повинна відбуватися перевірка і доробка практичної частини завдань .

Матеріал, розроблений в середовищі Visual Studio 2010 засобами OpenGL, дозволить студенту виконувати лабораторні та практичні завдання і перевірити рівень засвоєння матеріалу. Застосування таких курсів формує незалежну роботу в оволодінні новими знаннями, просуває того, хто навчається на новий щабель усвідомлення власних можливостей і свого місця в новому інформаційному суспільстві. учні отримують широку можливість засвоєння і закріплення набутих знань, і перевірки своїх результатів, а також уміння користуватися сучасними технологіями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ардабацкая И. А. Неформальное образование в школе: проблемы и перспективы. *Вестник Костромского государственного университета. Педагогика. Психология. Социокинетика*. 2016. № 1.
2. Бабаева Э. С. История неформального образования за рубежом. *Гуманизация образования*. 2015. № 2
3. Барболіна Т. М. Шкільний курс інформатики та методика його викладання: навч. посіб. Ч.1. Загальна методика. Полтава: Полтав. держ. пед. університет ім. В. Г. Короленка, 2007. 124 с.
4. Бирюкова И. К. Неформальное образование: понятие и сущность. *Известия ВГПУ*. 2012. №10
5. Выготский Л. С. Педагогическая психология. Москва, 1996.
6. Гаврилова И. В., Запруднова Л. А. Формальная, неформальная и информальная модели образования. *Молодой ученый*. 2016. №10. С. 1197-1200. — URL: <https://moluch.ru/archive/114/29876/>
7. Гукасова А. М., Мишарева Е. И., Романина В. И. Методика трудового обучения. Частные вопросы учебно-методическое пособие для студентов-заочников II-III курсов факультета педагогики и методики начального обучения педагогических институтов. Москва: Просвещение, 1990. 175 с.
8. Дегтярев В. М. Инженерная и компьютерная графика: учебник / за ред. В. М. Дегтярева, В. П. Затыльников. Москва, 2010. 238 с.
9. Залогова Л. А. Компьютерная графика. Москва, 2005. 319 с.
10. Золотарева А. В. Дополнительное образование детей в аспекте формальных и неформальных характеристик. *Ярославский педагогический вестник*. 2015. № 4. С. 46–53.
11. Иванова И. В. Неформальное образование инвестиции в человеческий капитал. *Вестник Томского государственного университета*. 2015. № 390.
12. Кудрявцев Т. В. Психология технического творчества: процесс и способы решения технических задач. Москва: Педагогика, 1975. 304 с.

13. Кульпединова М. Е. Неформальное образование в общественном объединении как фактор социального самоопределения детей и молодежи. *Вестник Костромского государственного университета им. Н.А. Некрасова. Педагогика. Психология. Социальная работа. Ювенология. Социокинетика*. 2012. Т. 18. № 1. Ч. 2. С. 198–202.
14. Макарский А. М. Формирование экологической культуры учащихся в условиях неформального экологического образования. 2007.
15. Мельникова Л. В., Осипова Л. В., Фридман Т. Б. Методика трудового обучения. Учебное пособие для учащихся пед. училищ по спец. преподавание труда и черчения в 4-8 кл. сред. общеобразовательной школы / под ред. Л. В. Мельниковой. Москва: Просвещение, 1985, 224 с.
16. Методичні рекомендації щодо викладання навчальних предметів у 2020/2021 навчальному році від 11.08.2020 № 1/9 430 с. — URL: <https://mon.gov.ua/storage/app/uploads/public/5f4/cae/d10/5f4caed10f675968632995.pdf>
17. Мухлаева Т. В. Международный опыт неформального образования. Человек и образование. 2010. № 4. С. 158–162.
18. Песковский Е. А. Неформальное образование как средство формирования готовности старшеклассников к высшему образованию: автореф. дис. ... канд. пед. наук. Красноярск, 1997.
19. Полковникова А. Ю. Компьютерная графика. Виды компьютерной графики. — URL: <http://xn--c1ajtfjx.xn--p1ai/kompyuternaya-grafika-vidy-kompyuternoj-grafiki>
20. Порев В. Н. Компьютерная графика, 2005. 428 с.
21. Поревит В. Н. Комп'ютерна графіка. 2002.
22. Програма для загальноосвітніх навчальних закладів. — URL: <https://imzo.gov.ua/osvita/zagalno-serednya-osvita-2/navchalni-prohramy-5-9-klasy-naskrizni-zmistovi-liniji/informatyka-naskrizni-zmistovi-liniji/>
23. Програмування GLUT: вікна і анімація. Miguel Angel Sepulveda, LinuxFocus.

24. Техническое творчество учащихся книга ,учебное пособие Столяров Ю. С., Комский Д. М., Гетта В. Г., Плуток А. М., Колотилов В. В. Москва: Просвещение, 1989
25. Тихомиров Ю. Програмування тривимірної графіки, 1998..
26. Хамадаш А. Неформальное образование: концепции, состояние, перспективы. *ЮНЕСКО*. 1992. № 1/2.
27. Херн Бейкер. *Комп'ютерна графіка і стандарт OpenGL*, 3-е видання: пер. з англ. Москва: Видавництво будинок «Вільямс», 2005. 1 168 с.
28. Хілл, Ф. *OpenGL. Програмування комп'ютерної графіки. Для професіоналів: «Пітер»*, 2004. 1088 с.
29. Шикин А. В., Боресков А. В. *Комп'ютерна графіка. Полігональні моделі.* Москва: ДИАЛОГ-МІФІ, 2001
30. Coombs P. and Ahmed M. *Attacking Rural Poverty. How non-formal education can help*, Baltimore: John Hopkins Press., 1974.
31. *OpenGL performance optimization, Siggraph'97 course.*
32. *The OpenGL graphics system: a specification (version 1.1).*
33. *The OpenGL Utility Toolkit (GLUT) Programming Interface, API version 3, specification.*
34. *Visual Introduction in OpenGL, SIGGRAPH'98.*

ДОДАТОК А

Лістинг програми

```
#include <windows.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glaux.h>
#include <GL/glut.h>
#include <math.h>
#include <stdlib.h>

int xOrigin = -1;
float angle = 0.0;
float lx = 0.0f, lz = -1.0f;
//float x = -0.36f, z = 7.0f;
float x = -3.5f, z = 7.0f;
float deltaAngle = 0.0f;
float deltaMove = 0;
float ly = 0.7f, y = 0.0f, deltaMoveUp = 0.0f;

void changeSize(int w, int h) {
    float ratio;
    if (h == 0) h = 1;
    ratio = w * 1.0 / h;
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(70.0f, ratio, 0.1f, 100.0f);
    glMatrixMode(GL_MODELVIEW);
}

void door()
{
    glPushMatrix();
    glTranslated(4, -0.45, -7);
    glRotated(90, 0, 1, 0);
    glColor3ub(139, 69, 19);
    auxSolidBox(1.5, 3, 0.18);
    glPopMatrix();

    glPushMatrix();
    glTranslated(4, 0.3, -7);
    glRotated(90, 0, 1, 0);
    glColor3ub(250, 235, 215);
    auxSolidBox(1, 1, 0.2);
}
```

```

glPopMatrix();

glPushMatrix();
glTranslated(4, -1.2, -7);
glRotated(90, 0, 1, 0);
glColor3ub(250, 235, 215);
auxSolidBox(1, 1, 0.2);
glPopMatrix();

glPushMatrix();
glTranslated(4, -0.45, -7.85);
glRotated(90, 0, 1, 0);
glColor3ub(0, 0, 0);
auxSolidBox(0.2, 3, 0.3);
glPopMatrix();

glPushMatrix();
glTranslated(4, -0.45, -6.15);
glRotated(90, 0, 1, 0);
glColor3ub(0, 0, 0);
auxSolidBox(0.2, 3, 0.3);
glPopMatrix();

glPushMatrix();
glTranslated(4, 1.1, -7);
glRotated(90, 0, 1, 0);
glColor3ub(0, 0, 0);
auxSolidBox(1.9, 0.2, 0.3);
glPopMatrix();

glPushMatrix();//дверная ручка
GLUquadricObj* quadObj;
quadObj = gluNewQuadric();
glColor3ub(200, 200, 200);
glRotated(90, 1, 0, 0);
gluQuadricDrawStyle(quadObj, GLU_FILL);
glTranslated(3.89, -6.37, 0.2);
gluCylinder(quadObj, 0.04, 0.04, 0.43, 60, 60);
glPopMatrix();

glPushMatrix();
glColor3ub(200, 200, 200);
glRotated(90, 1, 0, 0);
gluQuadricDrawStyle(quadObj, GLU_FILL);

```

```
        glTranslated(4.11, -6.37, 0.2);
        gluCylinder(quadObj, 0.04, 0.04, 0.43, 60, 60);
        glPopMatrix();
    }

    void board()
    {

        glPushMatrix();
        glTranslated(3.95, 0.5, -3.5);
        glRotated(90, 0, 1, 0);
        glColor3ub(139, 69, 19);
        auxSolidBox(4.5, 2.25, 0.18);
        glPopMatrix();

        glPushMatrix();
        glTranslated(3.9, -0.6, -3.5);
        glRotated(90, 0, 1, 0);
        glColor3ub(0, 0, 0);
        auxSolidBox(4.5, 0.1, 0.3);
        glPopMatrix();
    }

    void window()
    {

        glPushMatrix();
        glTranslated(1, 0.5, -8.92);
        glRotated(90, 0, 1, 0);
        glColor3ub(135, 206, 235);
        auxSolidBox(0.18, 2.25, 3);
        glPopMatrix();

        glPushMatrix();
        glTranslated(1, 1.65, -8.92);
        glRotated(90, 0, 1, 0);
        glColor3ub(0, 0, 0);
        auxSolidBox(0.34, 0.1, 3.2);
        glPopMatrix();

        glPushMatrix();
        glTranslated(-0.55, 0.5, -8.92);
        glRotated(90, 0, 1, 0);
        glColor3ub(0, 0, 0);
        auxSolidBox(0.34, 2.25, 0.1);
    }
}
```

```

    glPopMatrix();

    glPushMatrix();
    glTranslated(0.55, 0.5, -8.92);
    glRotated(90, 0, 1, 0);
    glColor3ub(0, 0, 0);
    auxSolidBox(0.34, 2.25, 0.1);
    glPopMatrix();

    glPushMatrix();
    glTranslated(1.55, 0.5, -8.92);
    glRotated(90, 0, 1, 0);
    glColor3ub(0, 0, 0);
    auxSolidBox(0.34, 2.25, 0.1);
    glPopMatrix();

    glPushMatrix();
    glTranslated(2.55, 0.5, -8.92);
    glRotated(90, 0, 1, 0);
    glColor3ub(0, 0, 0);
    auxSolidBox(0.34, 2.25, 0.1);
    glPopMatrix();

    glPushMatrix();
    glTranslated(1, -0.65, -8.70);
    glRotated(90, 0, 1, 0);
    glColor3ub(0, 0, 0);
    auxSolidBox(0.8, 0.1, 3.2);
    glPopMatrix();

    glPushMatrix();
    glTranslated(1, -1.375, -8.7);
    glBegin(GL_QUADS);
    glColor3ub(121, 195, 143);
    auxSolidBox(3.2, 1.4, 0.3);
    glPopMatrix();
}
void window2()
{
    glPushMatrix();//window
    glTranslated(-5, 0, 0);
    window();
    glPopMatrix();
}

void shelf()

```

```
{  
    glPushMatrix();  
    glTranslated(-7.5, -1.97, -8.2);  
    glBegin(GL_QUADS);  
    glColor3ub(238, 149, 114);  
    auxSolidBox(2.5,0.05 ,1.4 );  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslated(-7.5, 1.97, -8.2);  
    glBegin(GL_QUADS);  
    glColor3ub(238, 149, 114);  
    auxSolidBox(2.5,0.05 ,1.4 );  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslated(-8.75, -0.01, -8.2);  
    glBegin(GL_QUADS);  
    glColor3ub(238, 149, 114);  
    auxSolidBox(0.05,4,1.4 );  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslated(-6.25, -0.01, -8.2);  
    glBegin(GL_QUADS);  
    glColor3ub(238, 149, 114);  
    auxSolidBox(0.05,4,1.4 );  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslated(-7.5, -0.01, -7.5);  
    glBegin(GL_QUADS);  
    glColor3ub(238, 149, 114);  
    auxSolidBox(2.5,4,0.05);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslated(-7.5, -0.01, -7.47);  
    glBegin(GL_QUADS);  
    glColor3ub(0, 0, 0);  
    auxSolidBox(0.01,4,0.01);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslated(-7.7, -0.01, -7.45);  
    glColor3ub(0, 0, 0);
```

```
auxSolidSphere(0.08);
glPopMatrix();

glPushMatrix();
glTranslated(-7.3, -0.01, -7.45);
glColor3ub(0, 0, 0);
auxSolidSphere(0.08);
glPopMatrix();

glPushMatrix();
glTranslated(-10, -0.01, -8.2);
glBegin(GL_QUADS);
glColor3ub(238, 149, 114);
auxSolidBox(0.05,4,1.4 );
glPopMatrix();

glPushMatrix();
glTranslated(-9.4, 1.97, -8.2);
glBegin(GL_QUADS);
glColor3ub(238, 149, 114);
auxSolidBox(1.25,0.05 ,1.4 );
glPopMatrix();

glPushMatrix();
glTranslated(-9.4, -1.97, -8.2);
glBegin(GL_QUADS);
glColor3ub(238, 149, 114);
auxSolidBox(1.25,0.05 ,1.4 );
glPopMatrix();

glPushMatrix();
glTranslated(-9.4, -0.97, -8.2);
glBegin(GL_QUADS);
glColor3ub(238, 149, 114);
auxSolidBox(1.25,0.05 ,1.4 );
glPopMatrix();

glPushMatrix();
glTranslated(-9.4, 0.97, -8.2);
glBegin(GL_QUADS);
glColor3ub(238, 149, 114);
auxSolidBox(1.25,0.05 ,1.4 );
glPopMatrix();

glPushMatrix();
glTranslated(-8.78, -0.01, -8.87);
```

```

    glBegin(GL_QUADS);
    glColor3ub(238, 149, 114);
    auxSolidBox(2.5,4,0.05);
    glPopMatrix();
}

void toys()
{
    glPushMatrix();
    glTranslated(-10, -1.55, -8.2);
    glColor4ub(245, 0, 0, 210);
    auxSolidSphere(0.4);
    glPopMatrix();

    glPushMatrix();
    glTranslated(-9.6, -1.8, -7.9);
    glColor4ub(245, 245, 0, 210);
    auxSolidSphere(0.2);
    glPopMatrix();

    glPushMatrix();
    glTranslated(-9.85, -0.82, -8);
    glColor4ub(245, 245, 0, 210);
    glRotated(90, 1, 0, 0);
    auxSolidTorus(0.13,0.45);
    glPopMatrix();

    glPushMatrix();
    glTranslated(-9.85, -0.59, -8);
    glColor4ub(245, 0, 0, 210);
    glRotated(90, 1, 0, 0);
    auxSolidTorus(0.13,0.40);
    glPopMatrix();

    glPushMatrix();
    glTranslated(-9.85, -0.38, -8);
    glColor4ub(0, 245, 0, 210);
    glRotated(90, 1, 0, 0);
    auxSolidTorus(0.13,0.35);
    glPopMatrix();

    glPushMatrix();
    glTranslated(-9.85, -0.16, -8);
    glColor4ub(0, 0, 245, 210);

```

```
glRotated(90, 1, 0, 0);
auxSolidTorus(0.13,0.3);
glPopMatrix();

glPushMatrix();
glTranslated(-9.85, 0.05, -8);
glColor4ub(0, 245, 245, 210);
glRotated(90, 1, 0, 0);
auxSolidTorus(0.13,0.25);
glPopMatrix();

glPushMatrix();
glTranslated(-9.85, 0.25, -8);
glColor4ub(245, 0, 245, 210);
glRotated(90, 1, 0, 0);
auxSolidTorus(0.13,0.2);
glPopMatrix();

glPushMatrix();
glTranslated(-9.85, 0.45, -8);
glColor4ub(245, 245, 245, 210);
glRotated(90, 1, 0, 0);
auxSolidTorus(0.13,0.15);
glPopMatrix();

glPushMatrix();
glTranslated(-9.85, 0.7, -8);
glColor4ub(205, 91, 69, 240);
auxSolidSphere(0.2);
glPopMatrix();

glPushMatrix();
glTranslated(-10.2, 1.3, -8.2);
glColor4ub(0, 0, 245, 240);
auxSolidCube(0.60);
glPopMatrix();

glPushMatrix();
glTranslated(-9.6, 1.25, -8.2);
glColor4ub(245, 245, 0, 240);
auxSolidCube(0.5);
glPopMatrix();

glPushMatrix();
```

```
        glTranslated(-9.6, 1.65, -8.2);
        glColor4ub(245, 0, 0, 240);
        auxSolidCube(0.3);
        glPopMatrix();
    }

void lamp() {
    glPushMatrix();
    glTranslated(-1.45, 1.85, -8.9);
    glColor4ub(245, 202, 56, 200);
    auxSolidSphere(0.5);
    glPopMatrix();
}

void carpet1()
{
    glPushMatrix();
    glColor3ub(255, 255, 255);
    glTranslated(-11, -1.97, -7);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(0, 0, 0);
    glTranslated(-11, -1.97, -6.25);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(255, 255, 255);
    glTranslated(-11, -1.97, -5.5);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(0, 0, 0);
    glTranslated(-11, -1.97, -4.75);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(255, 255, 255);
    glTranslated(-11, -1.97, -4);
```

```

    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(0, 0, 0);
    glTranslated(-11, -1.97, -3.25);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(255, 255, 255);
    glTranslated(-11, -1.97, -2.5);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75,0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(0, 0, 0);
    glTranslated(-11, -1.97, -1.75);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();
}

void carpet2()
{
    glPushMatrix();
    glColor3ub(0, 0, 0);
    glTranslated(-10.25, -1.97, -7);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(255, 255, 255);
    glTranslated(-10.25, -1.97, -6.25);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(0, 0, 0);
    glTranslated(-10.25, -1.97, -5.5);

```

```

    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(255,255, 255);
    glTranslated(-10.25, -1.97, -4.75);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(0, 0, 0);
    glTranslated(-10.25, -1.97, -4);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(255, 255,255);
    glTranslated(-10.25, -1.97, -3.25);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(0, 0, 0);
    glTranslated(-10.25, -1.97, -2.5);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(255, 255,255);
    glTranslated(-10.25, -1.97, -1.75);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.75, 0.75, 0.01);
    glPopMatrix();
}

void carpet()
{
    glPushMatrix();
    carpet1();
    glPopMatrix();
}

```

```
    glPushMatrix();
    carpet2();
    glPopMatrix();

    glPushMatrix();
    glTranslated(1.5, 0, 0);
    carpet1();
    glPopMatrix();

    glPushMatrix();
    glTranslated(1.5, 0, 0);
    carpet2();
    glPopMatrix();

    glPushMatrix();
    glTranslated(3, 0, 0);
    carpet1();
    glPopMatrix();

    glPushMatrix();
    glTranslated(3, 0, 0);
    carpet2();
    glPopMatrix();

    glPushMatrix();
    glTranslated(4.5, 0, 0);
    carpet1();
    glPopMatrix();

    glPushMatrix();
    glTranslated(4.5, 0, 0);
    carpet2();
    glPopMatrix();
}

void teacher()
{
    glPushMatrix();
    glColor3ub(139, 26, 26);
    glTranslated(0.5, -1.05, -2.5);
    glRotated(90, 1, 0, 0);
    auxSolidBox(1, 2, 0.05);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(222, 184, 135);
```

```
glTranslated(0.2, -1.65, -1.75);
glRotated(90, 1, 0, 0);
auxSolidBox(0.1, 0.1, 1.2);
glPopMatrix();

glPushMatrix();
glColor3ub(222, 184, 135);
glTranslated(0.2, -1.65, -3.22);
glRotated(90, 1, 0, 0);
auxSolidBox(0.1, 0.1, 1.2);
glPopMatrix();

glPushMatrix();
glColor3ub(222, 184, 135);
glTranslated(0.8, -1.65, -1.75);
glRotated(90, 1, 0, 0);
auxSolidBox(0.1, 0.1, 1.2);
glPopMatrix();

glPushMatrix();
glColor3ub(222, 184, 135);
glTranslated(0.8, -1.65, -3.22);
glRotated(90, 1, 0, 0);
auxSolidBox(0.1, 0.1, 1.2);
glPopMatrix();

glPushMatrix();
glColor3ub(139, 26, 26);
glTranslated(1.7, -1.45, -2.5);
glRotated(90, 1, 0, 0);
auxSolidBox(0.05, 2, 1.1 );
glPopMatrix();

glPushMatrix();
glColor3ub(222, 184, 135);
glTranslated(1.375, -1.35, -2.5);
glRotated(90, 1, 0, 0);
auxSolidBox(0.65, 2, 0.05 );
glPopMatrix();

glPushMatrix();
glColor3ub(222, 184, 135);
glTranslated(1.05, -1.675, -2.5);
glRotated(90, 1, 0, 0);
auxSolidBox(0.05, 2, 0.65 );
glPopMatrix();
```

```
}  
  
void parta()  
{  
    glPushMatrix();  
    glColor3ub(222, 184, 135);  
    glTranslated(-0.6, -1.05, -2.5);  
    glRotated(90, 1, 0, 0);  
    auxSolidBox(1, 2, 0.05);  
    glPopMatrix();  
  
    glPushMatrix();  
    glColor3ub(139, 26, 26);  
    glTranslated(-0.9, -1.65, -1.75);  
    glRotated(90, 1, 0, 0);  
    auxSolidBox(0.1, 0.1, 1.2);  
    glPopMatrix();  
  
    glPushMatrix();  
    glColor3ub(139, 26, 26);  
    glTranslated(-0.9, -1.65, -3.22);  
    glRotated(90, 1, 0, 0);  
    auxSolidBox(0.1, 0.1, 1.2);  
    glPopMatrix();  
  
    glPushMatrix();  
    glColor3ub(139, 26, 26);  
    glTranslated(-0.3, -1.65, -1.75);  
    glRotated(90, 1, 0, 0);  
    auxSolidBox(0.1, 0.1, 1.2);  
    glPopMatrix();  
  
    glPushMatrix();  
    glColor3ub(139, 26, 26);  
    glTranslated(-0.3, -1.65, -3.22);  
    glRotated(90, 1, 0, 0);  
    auxSolidBox(0.1, 0.1, 1.2);  
    glPopMatrix();  
  
    glPushMatrix();  
    glColor3ub(222, 184, 135);  
    glTranslated(-1.8, -1.45, -2.5);  
    glRotated(90, 1, 0, 0);  
    auxSolidBox(0.05, 2, 1.1 );  
    glPopMatrix();  
}
```

```

    glPushMatrix();
    glColor3ub(139, 26, 26);
    glTranslated(-1.475, -1.35, -2.5);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.65, 2,0.05 );
    glPopMatrix();

    glPushMatrix();
    glColor3ub(139, 26, 26);
    glTranslated(-1.15, -1.675, -2.5);
    glRotated(90, 1, 0, 0);
    auxSolidBox(0.05, 2,0.65 );
    glPopMatrix();

}
void sitting()
{
    glPushMatrix();
        glTranslated(0.5,0 ,0 );
        teacher();
        glPopMatrix();

        glPushMatrix();
        glTranslated(0.5,0 ,0 );
        parta();
        glPopMatrix();

        glPushMatrix();
        glTranslated(-1.3,0 ,0 );
        parta();
        glPopMatrix();

        glPushMatrix();
        glTranslated(-3.1,0 ,0 );
        parta();
        glPopMatrix();

        glPushMatrix();
        glTranslated(0.5,0 ,-3 );
        parta();
        glPopMatrix();

        glPushMatrix();
        glTranslated(-1.3,0 ,-3);

```

```

    parta();
    glPopMatrix();

    glPushMatrix();
    glTranslated(-3.1,0 ,-3 );
    parta();
    glPopMatrix();
}

void room() {
    glPushMatrix();// floor
    glColor3ub(222, 184, 135);
    glTranslated(-4, -2.05, -5);
    glRotated(90, 1, 0, 0);
    auxSolidBox(16, 8, 0.15);
    glPopMatrix();

    glPushMatrix(); //left wall
    glTranslated(-4, 0.375, -8.925);
    glBegin(GL_QUADS);
    glColor3ub(121, 195, 143);
    auxSolidBox(16, 5, 0.15);
    glPopMatrix();

    glPushMatrix(); //right wall
    glTranslated(4, 0.375, -5);
    glRotated(90, 0, 1, 0);
    glColor3ub(121, 195, 143);
    auxSolidBox(8, 5, 0.15);
    glPopMatrix();

    glPushMatrix(); //door
    door();
    glPopMatrix();

    glPushMatrix(); //board
    board();
    glPopMatrix();

    glPushMatrix();//window
    window();
    glPopMatrix();
    glPushMatrix();
    window2();
    glPopMatrix();
}

```

```

    glPushMatrix();
    glTranslated(-0.5,0 ,0 );
    shelf();
    glPopMatrix();

    glPushMatrix();
    toys();
    glPopMatrix();

    glPushMatrix();
    carpet();
    glPopMatrix();

    glPushMatrix();
    sitting();
    glPopMatrix();

    glPushMatrix();
    lamp();
    glPopMatrix();
    glPushMatrix();
    glTranslated(5.5,0.4 ,4);
    lamp();
    glPopMatrix();
}

void computePosUp(float deltaMoveUp) {
    y += deltaMoveUp * 0.1f;
}

void computePos(float deltaMove) {
    x += deltaMove * lx * 0.1f;
    z += deltaMove * lz * 0.1f;
}

void computeDir(float deltaAngle) {
    angle += deltaAngle;
    lx = sin(angle);
    lz = -cos(angle);
}

void display(void) {

    if (deltaMove) computePos(deltaMove);
    if (deltaAngle) computeDir(deltaAngle);
    if (deltaMoveUp) computePosUp(deltaMoveUp);
}

```

```

glClearColor(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
glLoadIdentity();
gluLookAt(x, y, z, x + lx, y, z + lz, 0.0f, 1.0f, 0.0f);
glEnable(GL_ALPHA_TEST);
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
glPushMatrix();
glRotated(45, 0, 1, 0);
room();
glPopMatrix();
glDisable(GL_ALPHA_TEST);
glDisable(GL_BLEND);
glutSwapBuffers();
}

```

```

void pressKey(int key, int xx, int yy) {
    switch (key) {
        case GLUT_KEY_LEFT:
            deltaAngle = -0.03f;
            break;
        case GLUT_KEY_RIGHT:
            deltaAngle = 0.03f;
            break;
        case GLUT_KEY_UP:
            deltaMove = 0.9f;
            break;
        case GLUT_KEY_DOWN:
            deltaMove = -0.9f;
            break;
        case GLUT_KEY_PAGE_UP:
            deltaMoveUp = 0.9f;
            break;
        case GLUT_KEY_PAGE_DOWN:
            deltaMoveUp = -0.5f;
            break;
    }
}

```

```

void releaseKey(int key, int x, int y) {
    switch (key) {
        case GLUT_KEY_LEFT:
        case GLUT_KEY_RIGHT:
            deltaAngle = 0.0f;
            break;
        case GLUT_KEY_UP:
        case GLUT_KEY_DOWN:

```

```

        deltaMove = 0;
        break;
    case GLUT_KEY_PAGE_UP:
    case GLUT_KEY_PAGE_DOWN:
        deltaMoveUp = 0.0f;
        break;
    }
}

int main(int argc, char** argv) {
    float pos[4] = { 3, 3, 3, 1 }, dir[3] = { 2, 2, -2 }, color[4] = { 0.45, 0.25, 0.6, 1 },
    pos1[4] = { -2.5, 5, -2, 1 },
    color1[4] = { 0.3, 0.2, 0.1, 1 }, color2[4] = { 0.3, 0.2, 0.1, 1 }, pos2[4] = { -1, 3, -4,
    1 }, ambient[4] = { 0.001, 0.001, 0.001, 1 };
    GLfloat mat_specular[] = { 0.05, 0.05, 0.1, 1 };
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
    glutInitWindowPosition(500, 300);
    glutInitWindowSize(1920, 1080);
    glutCreateWindow("Щепотін Кирсова робота");
    glutDisplayFunc(display);
    glutReshapeFunc(changeSize);
    glutIdleFunc(display);
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_COLOR_MATERIAL);
    glEnable(GL_LIGHTING);
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT, ambient);
    glEnable(GL_LIGHT0);
    glLightfv(GL_LIGHT0, GL_POSITION, pos);
    glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, dir);
    glEnable(GL_LIGHT1);
    glEnable(GL_LIGHT2);
    glLightfv(GL_LIGHT1, GL_POSITION, pos1);
    glLightfv(GL_LIGHT2, GL_AMBIENT, color1);
    glLightfv(GL_LIGHT2, GL_POSITION, pos2);
    glMaterialfv(GL_FRONT, GL_EMISSION, mat_specular);
    glutIgnoreKeyRepeat(1);
    glutSpecialUpFunc(releaseKey);
    glutSpecialFunc(pressKey);
    glutMainLoop();
    return 1;}

```