



PROGRAM PROJECT APPROACH IN COMPUTER GRAPHIC LEARNING

Abstract: The article presents a training approach in Computer Graphics, based on pre-detailed projects. Projects decomposition into small tasks and number projects planning lead to the programmed achievement of Computer Graphics course objectives. The use of the approach in conventional and e-learning are commented.

Author information:

Rositsa P. Hristova

PhD student

at Faculty of Mathematics and Computer science,
at Konstantin Preslavsky – University of Shumen

✉ r.hristova@shu.bg

🌐 Bulgaria

Keywords:

Computer graphics, active learning,
problem-based, project-based, programming
project-based, OpenGL.

Настоящата статия се реализира с финансовата подкрепа на фонд за научни изследвания при ШУ „Епископ Константин Преславски”, проект РД 08-89/28.01.2020 г.

В последните години педагогическите експерименти доказват огромната полза от активното учене (*active learning*). То включва различни подходи, обединени в едно – поставяне на обучаемите като активна единица в центъра на образователния процес. При него студентът не е пасивен регистратор на лекторските идеи, а активен участник в обучението, което се превръща в обмен на идеи. Подходите включват участие в различни видове образователни игри, експериментиране със симулационни модели, проектно базирано обучение, работа в екип, използване на компютърни системи за електронно обучение и др. Въпреки че се прилага за различни области, най-голям дял, активното учене има в природните науки, математиката и информатиката. Бурното развитие на ИКТ, интернет и социалните мрежи допринесе за разширяване на възможностите за активно обучение.

Компютърната графика се занимава със синтеза, представянето, обработката и изобразяването на графична информация. Компютърната графика е сравнително млада интердисциплинарна наука, която се базира, обединява и ползва множество принципи от други науки като математика, физика, информатика, колориметрия и други. Това интегриране на принципи от различни науки от една страна е призвано да обслужва колкото се може по-голям кръг практически задачи. Но от друга страна изисква от обучаемите знания за понятия, абстрактни структури, алгоритми, методи и подходи от информатиката, математиката и физиката. Това предполага търсенето на иновативни методи и подходи, с които да бъде понижена степента на абстракция и да бъде засилен интереса към учебния материал.

Учебната дисциплина “*Компютърна графика*” се преподава на студенти от направление 4.6 Информатика и компютърни науки в четвърти курс. Целта на курса е обучаемите да получат представа за основните идеи при работа с графична информация и в частност в компютърната графика. Разглеждат се въпросите за: обработка на графична информация, входни и изходни графични устройства и управлението им, интерактивна работа с графични устройства и геометрично моделиране.

Основно внимание се обръща на представянето на специфични графични алгоритми. Застъпено е генерирането и преобразуването на дву- и тримерни графични изображения. Дадени са възможностите за създаване на реалистични цветни изображения. Обсъждат се въпроси за създаване на анимационни модели в реално време. Разглеждат се постановките за синтеза, обработката и съхранението на компютърни изображения за целите на визуализацията

В часовете за упражненията студентите разработват софтуерни проекти. Използва се средата за разработване на приложения CodeBlocks. Работи се с базовата графична библиотека OpenGL и помощната библиотека GLUT.

Затрудненията, които изпитват студентите в курса по Компютърна графика произтичат от пропуските в уводните курсове по програмиране, алгебра и геометрия. “Дупките” по математика и липсата на абстрактно мислене обезкуражава част от студентите. От съществено значение за повишаване на мотивацията на студентите е да разберат, защо им е полезна изучаваната дисциплина, има ли тя реално приложение, били им била полезна при търсене на работа. Мотивацията на обучаемите се увеличава, ако са уверени в приложимостта на усвоените компетентности.

Останалата част от статията се отнася главно за практическата част от курса (упражненията), където студентите разработват софтуерни проекти, изучавайки *графичните библиотеки OpenGL и GLUT*. По същество това е програмиране с изучаване на специализирани библиотеки. Когато се касае за програмиране, „*задачата* е задание, за което се изисква да се напише програма. Обикновено за решаване на задачата е достатъчно директно прилагане на знания от малка по обем тема ... Задачите по програмиране са добре формулирани и имат добре определени крайни резултати“[1]. В зависимост от начина, по който се използват задачите в целия курс, са известни два основни подхода:

- **Problem solving** [2] – решават се задачи с цел практически да се илюстрират свойствата на въведените понятия (вид данни, функция или метод).... Нарича се **обучение чрез решаване на задачи**. Подходът е масово предлаган в началното обучение по програмиране.
- **Problem based** [2] – при този подход задачите определят темата т.е. те са водещи и едновременно с решаването им се въвеждат нови понятия и методи. При този подход въвеждането на всяко ново понятие е предшествано от необходимостта от неговото използване. Прилагането на **проблемно-ориентирания подход** по време на целия курс не винаги е целесъобразно. На практика се прилага комбинация от двата подхода, в която водещ е проблемно-ориентирания подход и това е съображението да се използва същия термин.

„**Програмния проект**[1] е задание, изпълнението на което изисква решаването на няколко свързани помежду си задачи. В този смисъл на проекта може да се гледа като на сложна задача, обединяваща понятия, въведени в няколко раздела. ... Много често крайният резултат от един проект има практическа стойност“. Обикновено проектите се формулират без да се посочват всички детайли. Това означава, че от студентите се очаква сами да вземат решения.

- **Project based** [2] – при този подход водещи в целия курс са проектите. **Проектно-ориентираният подход** изисква специфични грижи от страна на преподавателя преди и по време на курса. ... За успешно прилагане на подхода се изисква от страна на студентите силна мотивация и добра подготовка. Това създава напрежение по време на семестъра.

Подхода използван в апробирания курс е комбинация от трите подхода. Учебният материал се групира в тематични единици, които могат да обхващат повече от една тема, като за всяка тематична единица се изготвя проект. За разлика от класическия проектно-ориентиран подход преподавателят предварително декомпозира проектите на малки задачи, които са

детайлно описани т.е. имаме планиране на стъпките, през които трябва да премине студента, за да завърши проекта. Целия курс се планира като редица от проекти, които частично се надграждат и съдържат варианти на вече решени задачи. За това подхода е наречен **програмирано-проектен**.

Защо **програмирано-проектен**?

Програмираното обучение е широко използвано през последните десетилетия. В специализираната литература липсва единно мнение за това понятие. Началото на програмираното обучение като ново направление в педагогиката е поставено от американския психолог Б. Ф. Скинър през 1954 г. с публикуването на неговата статия в списание „Харвард Едюкейшън Ривю”[3], озаглавена „Науката за ученето и изкуството да се обучава”. Скинър се възприема като родоначалник на линейното програмиране.

По същото време Н. Краудър също развива своите идеи за програмирано обучение, които в определени моменти съществено се отличават от тези на Б. Скинър. Според Краудер не трябва да се изключва възможността от грешки, допуснати от обучаемия в процеса на преподаване. Ако обучаемият отговори грешно, то трябва да му се даде разяснение за допуснатата грешка. Това позволява индивидуализация на обучението всеки обучаем минава по индивидуален път в зависимост от отговорите, които дава. Така той поставя началото на т.нар. **разклонено програмиране**[4].

В своята същност програмираното обучение представлява индивидуална подготовка на учащия с предварително разработена програма за обучение, в процеса на която той придобива знания. Това предоставя възможност за следване на индивидуален темп на работа за всеки учащ в зависимост от неговите знания, умения, психологическа нагласа и др. Характерно за програмираното обучение е разделянето на учебния материал на малки взаимосвързани части.

„Основно средство за реализация на този вид обучение е обучаваща програма. Това може да бъде: програмиран учебник или други печатни материали; програма, предоставена с учебна техника; система за електронно обучение.“[5]. Основната разлика между традиционното и програмираното обучение не е в основните принципи на обучение, а в начина на тяхното прилагане. При програмираното обучение се реализира обмен на информация между обучаващата програма и всеки обучаем поотделно. В традиционното обучение информацията се предава от обучаващия към всички обучаеми едновременно.

Всяко програмирано обучение се изгражда на базата на следните принципи[6]:

- Разделяне на материала на малки, тясносвързани части (кадри, стъпки);
- При усвояването на всяка доза учебен материал от обучаемия се изисква ответна реакция, отговор /Активизация на учащия;
- Незабавна реакция/оценка на отговора на всеки учащ;
- При грешен отговор обучаемият мисли, докато открие правилния или последният му се предоставя в готов вид и тогава преминава към следващия кадър;
- Индивидуализация на ритъма и съдържанието на учебния процес;
- Верификация на програмирания текст.

Тези шест основни принципа са в основата на използваните проекти за програмирано обучение.

В специализираната литература са известни още два вида програмиране, които намират приложение в предлагания подход:

- **смесено програмиране** – прави се съчетание между линейно и разклонено програмиране.
- **полупрограмираното обучение** – една част от материала се програмира, а друга част се разработва по традиционния начин.

Предлаганият **програмирано-проектен** подход естествено се базира на **общо-дидактическите принципи**[7]:

- ✓ Принцип за **научност** – на обучаемите се предоставят строго научни факти, идеи, изводи и закономерности;
- ✓ Принцип за **достъпност** – при декомпозиция на проектите трябва да се интегрират дидактическите правила за достъпност: от лекото към трудното, от простото към сложното, от близкото към далечното т.е. постепенно увеличаваме кръгозора на студентите;
- ✓ Принцип за **нагледност** – при решаване на всяка от задачите в проекта се получава визуален резултат;
- ✓ Принцип на **съзнателност** – организацията на учебния процес е такава, че фактите и закономерностите изискват осмисляне, за да бъдат направени верни изводи, които да бъдат проложени;
- ✓ Принцип на **активност** – подхода изисква всеки студент да работи самостоятелно в средата и създава собствен проект като извлича информация и генерира собствен опит;
- ✓ Принцип на **системност** – организацията на учебния процес е такава, че новите знания се получават в система с получените до момента знания;
- ✓ Принцип на **трайност на знанията и уменията** – психологическата основа на този принцип са процесите на запамятаване, задържане и възпроизвеждане на информация. При повторението и вариативното възпроизвеждане на информацията знанията и уменията се затвърждават;
- ✓ Принцип на **индивидуалния подход** – всеки студент да работи самостоятелно и създава собствен проект със собствена скорост реализирайки собствените си възможности;
- ✓ Принцип на **свобода и стимулиране на дейност** – процеса на получаване на нови знания е такъв, че провокира интереса на обучаемите към нови и по-задълбочени знания;
- ✓ Принцип на **цялостност** – определя способността на обучаемия цялостно да възприема обучението т.е. придобиването на знания да става в съответствие с цялостната концепция за крайния резултат от обучението.

Следващият пример е за начален проект за визуализация на 3D обекти, който се изпълнява самостоятелно от студентите в час. Използва се готов шаблон за главната функция, в който присъстват основните структурни елементи. Затвърждават се знания за цветовете модели и прихващане на събития от клавиатурата. Упражнява се използването на функции за стандартните 3D обекти и трансформации в тримерното пространство.

Пример 1

Задание: ТРАНСФОРМАЦИИ В ПРОСТРАНСТВОТО

Заредете среда за програмиране с инсталирани библиотеки OpenGL и GLUT и главна функция, настроена за работа в 3D.

A) Тествайте функцията:

```
void drawScene( )
{ glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
  glMatrixMode( GL_MODELVIEW );
  glLoadIdentity( ); angle++;
  if ( angle>=360 ) angle=0;
  glTranslatef( 0., 0., -50. );
  glPushMatrix( );
  glRotatef( angle, 1, 1, 0 );
  glutWireCube( 2 );
  glPopMatrix( );
  glTranslatef( 2, 0, 0 );
  glutSolidCube( 2 );
```

```

    glutSwapBuffers( );
    glutPostRedisplay( );
}

```

Б) Поправете функцията, така че

- първия обект, който се изчертава да е с размер 5;
- сменете цвета на първия обект със зелен;
- втория обект, който се изчертава да е с бял цвят;
- първият изчертан обект да е сфера с радиус 4, 12 меридиана и 6 паралела.

В) Поправете функцията, така че

- мащабирайте втория обект по осите y и z с коефициент 0.3;
- преместете гледната точка/камерата на разстояние 100 по оста z .

Г) Нека при натискане на клавиш 'S' вторият обект започва постепенно да се премества по оста x .

Нека при четно натискане на клавиш 'S' вторият обект спира, а при нечетно натискане на клавиша тръгва отново.

Д) Добавете (във функцията за чертане) запълнена сфера, с радиус 6 в горната дясна четвъртинка от прозореца (максимално отдалечена от гледната точка) оцветена в жълто.

Е) Нека при натискане на:

- клавиш 'F' сферата започва да се отдалечава от наблюдателя (по оста z) със стъпка 0.3;
- клавиш 'N' сферата започва да се приближава към наблюдателя по оста z със стъпка

0.3.

Коментар за дейностите, които обучаемите извършват при създаването, тестването и изпълнението на проекта:

- ✓ Действия за настройване на средата за работа с графичните библиотеки за създаване на 3D графика (уводната част);
- ✓ Въвеждане и тестване на готова функция (подточка А);
- ✓ Корекция на функция за промяна характеристиките на обектите (подточка Б);
- ✓ Допълване на програмата с нови операции над обектите, които се визуализират чрез стандартни функции (подточка В);
- ✓ Добавяне от обучаемите на нови обекти в програмата (подточка Д);
- ✓ Добавяне на нови операции над обектите, които изискват проектиране на програмен фрагмент от страна на студентите (подточки Г и Е).

Пример 2

Задание: НАСОЧЕНА СВЕТЛИНА. ЕКСПЕРИМЕНТИРАНЕ С РАЗЛИЧНИ ВИДОВЕ СВЕТЛИНА

Заредете среда за програмиране с инсталирани библиотеки OpenGL и GLUT и главна функция, настроена за работа в 3D.

А) Тествайте функцията:

```

GLfloat position[]={ 0.0, 0.0, -0.2, 1.0 };
void drawScene( )
{ glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
  glMatrixMode( GL_MODELVIEW );
  glLoadIdentity( );
  glPushMatrix( );
  gluLookAt( 0., 0., 10., 0., 0., 0., 0., 1., 0. );
  glRotatef( spin, 1., 0., 0. );
}

```

```

glTranslatef( 0., 0., 2.5 );
glColor3f( 0.0, 1.0, 1.0 );
glutSolidCube( 0.2 );
glColor3f( 0.5, 0.5, 0.5 );
glutSolidTorus( 0.25, 0.85, 10, 16 );
glPopMatrix( );
glutSwapBuffers( );
glutPostRedisplay( );
}

```

Б) Във функцията за инициализация задайте цвят на фона тъмно син;

Включете проверка за разстоянията.

В) Дефинирайте функция Mouse за обработка на събития от мишката така че при натискане на ЛБМ ъгълът spin се променя с 15 градуса (и сцената се пречертава);

Добавете в главната обръщение към функцията за обработка на събития от мишката.

Г) Коригирайте функцията drawScene по следния начин:

Нека торът(Torus) да остане неподвижен, а кубчето да се завърта около него.

Д) Нека при натискане на:

- клавиш 'r' цветът на фона се сменя с червен(0.5);
- клавиш 'g' цветът на фона се сменя със зелен(0.5);
- клавиш 'b' цветът на фона се сменя с тъмно син;
- ДБМ цветът на фона се сменя с черен.

Е) Дефинирайте цвят на AMBIENT излъчване сив(0.2);

Нека при натискане на клавиш 'a' се включва първия светлинен източник(GL_LIGHT0) за AMBIENT излъчване;

Определете начина, по който обектите отразяват AMBIENT светлината;

Нека при натискане на клавиш '1' се включва динамичната светлина;

Нека при натискане на клавиш '0' се изключва динамичната светлина и светлинния източник(GL_LIGHT0).

Ж) Дефинирайте цвят на DIFFUSE излъчване от източника на светлина – жълт(0.8);

Включете светлинния източник GL_LIGHT1 за DIFFUSE излъчване;

Нека източникът на светлина(GL_LIGHT1) се движи заедно с кубчето и е застанал пред него (използвайте position);

Определете начина, по който обектите отразяват DIFFUSE светлината.

З) Дефинирайте цвят на SPECULAR излъчването от източника на светлина – сиво-син(сивия цвят(0.4) се коригира с 0.4 в синия цвят);

Включете светлинния източник GL_LIGHT1 за SPECULAR излъчване;

Определете начина, по който обектите отразяват SPECULAR светлината;

Включете отблясък от всички части от обектите с интензивност 40;

Пробвайте с жълта SPECULAR светлина.

И) Нека ъгълът на конуса, с които се разпространява светлината от светлинния източник GL_LIGHT1 стане 25;

К) Нека при натискане на клавиш 'c' цвета на обекта се взема предвид при осветяването му.

Коригирайте цвета на обекта с 0.3 в червения цвят.

Във връзка с епидемичната обстановка се наложи електронно обучение по дисциплината. Електронното обучение е обучение (Chunhua, 2008), използващо електронни технологии за достъп до учебното съдържание извън традиционната класна стая. В повечето случаи то се отнася до курс, програма или степен, доставени напълно онлайн. Предимството на електронното обучение е, че може да предложи ефективни учебни методи като дейности с

обратна връзка, адаптирани към индивидуалния ритъм на учене. Няма проблем проектите да се задават и получават чрез платформа за електронно обучение, а комуникацията и чрез видеоконферентна връзка.

References:

1. Azalov, P., 2013: Plaven prehod ot zadachi kum proekti v uvodnite kursove po programirane(Плавен преход от задачи към проекти в уводните курсове по програмиране), Mathematics and Informatics, vol. 56, num. 3, pp 224-237.
2. Kuru, S.(Ed.), 2007: Problem Based Learning, TREE Interest Group B5 “Problem based and project oriented learning” Isik University. <http://www3.unifi.it/tree/dl/oc/b5.pdf>
3. Skinner, B. F., 1968: The technology of teaching. In B. F. Skinner, The technology of teaching. New York: AppletonCenturyCrofts
4. Crowder, N. A., 1960: Automatic tutoring by intrinsic programming. In Lumsdaine A.A & Glaser R. (eds) 1960. Teaching machines and programmed learning I: a source book. Washington D.C. National Education Association of the United States
5. Yankov, M., 2018: Interaktiven interfeys na ternarnomodulen model pri izuchavane na pohyatiyata po strukturi ot dannii, disertatsionen trud za prisuzhdane na ONS doctor, Shumen
6. Richmond, U. K., 1968: Uchitelya i mashiny, Moskva, (Учителя и машины, Москва), pp. 58-59
7. Vitanova, N., 2014: Didaktika, CDO of Konstantin Preslavsky – University of Shumen, <http://cdo.shu.bg> (el.modul)
8. Chunhua, Z., 2008: Elearning: The New Approach for Knowledge Management (KM). International Conference on Computer Engineering, CSSE, vol. 5, 291–294