

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №7**

з дисципліни “Основи програмування”

тема “**Масиви структур даних**”

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент(ка) I курсу  групи КП-92  Мовчан Максим  (*прізвище, ім’я, по батькові*)  варіант №21 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

Київ 2019

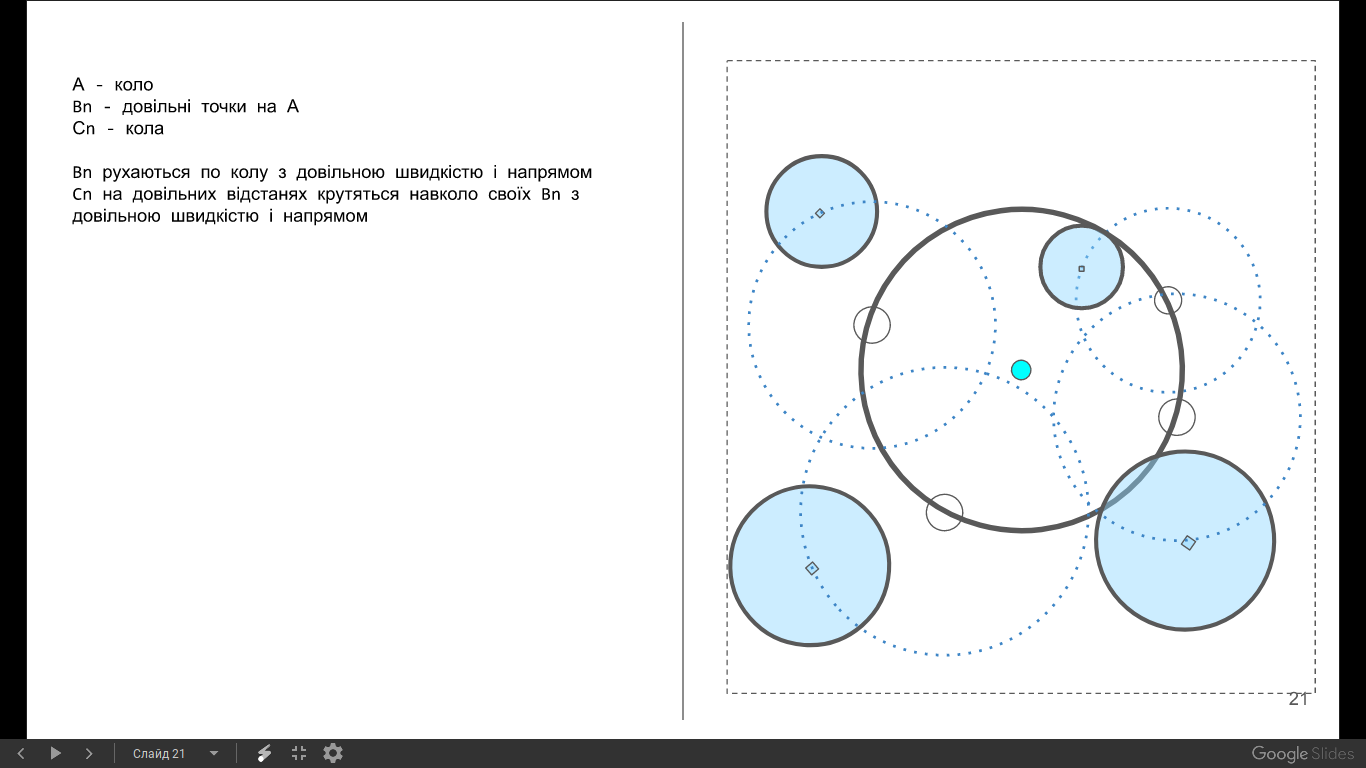
**Мета роботи**

Навчитися використовувати масиви структур даних.  
Навчитись використовувати вказівники на структури та масиви структур даних у користувацьких функціях.

**Постановка завдання**

Намалювати у терміналі динамічну графічну сцену, що складається із набору однотипних графічних об'єктів та піддається заданим правилам взаємодії в залежності від часового параметру t.

Варіант: Slide 21

Частина об'єктів має залежати від часового параметру t, що весь час збільшується у циклі. Інші об'єкти залежать від параметрів перших об'єктів відповідно до завдання.

**Аналіз вимог і проектування**

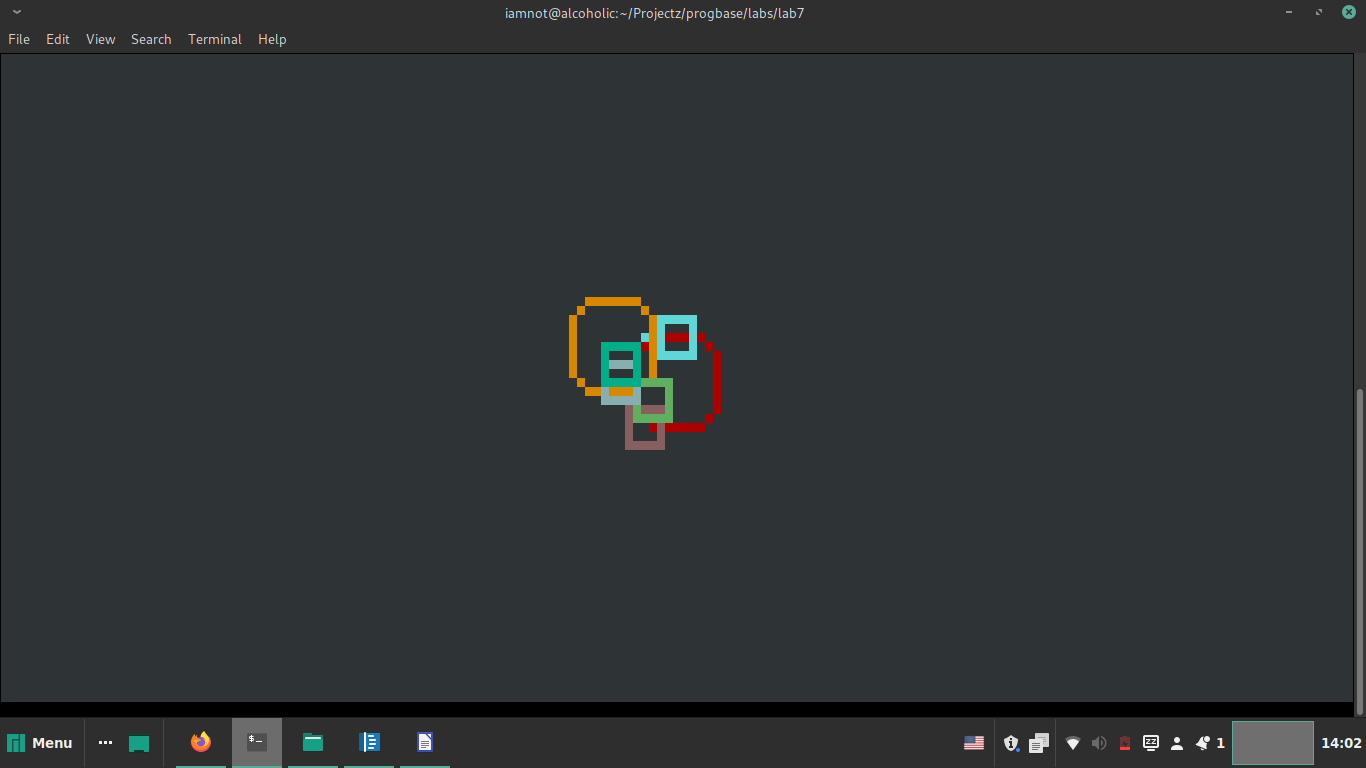
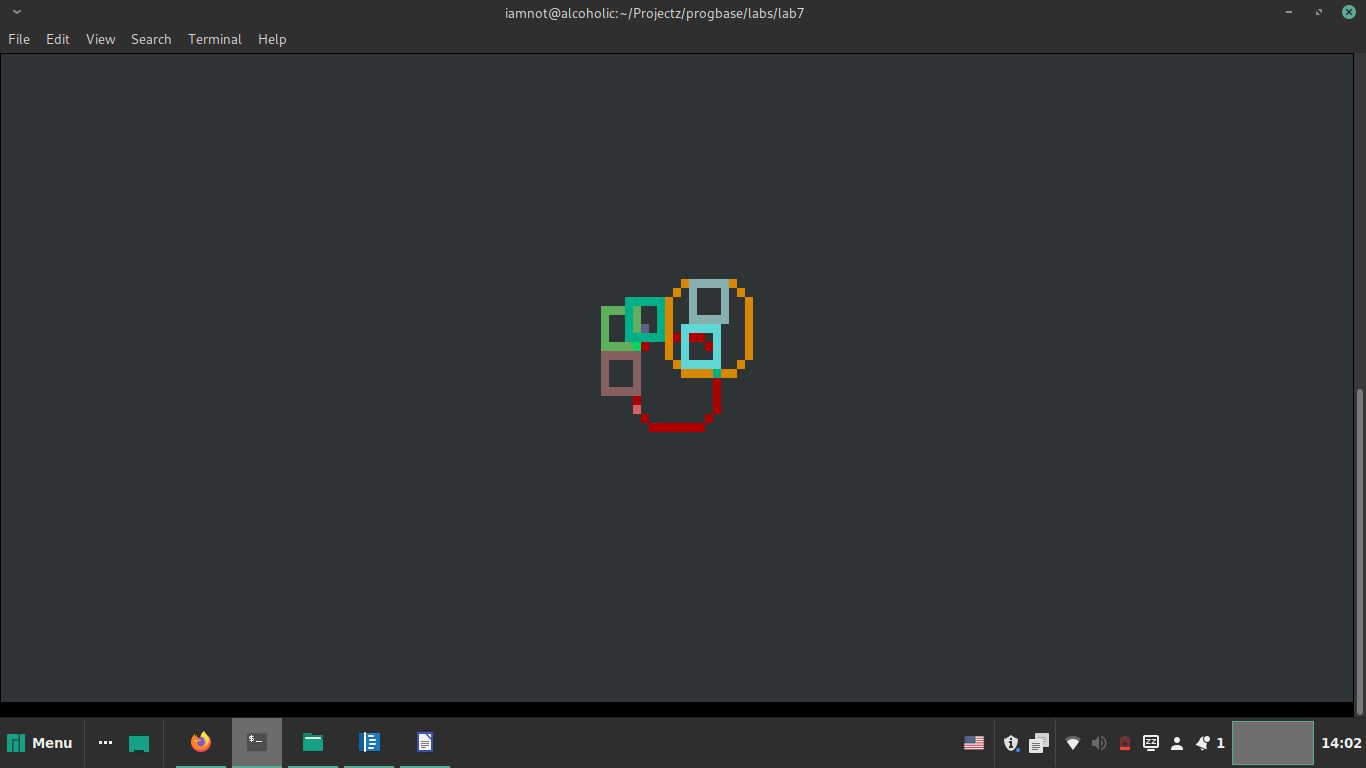
Об’єкти:

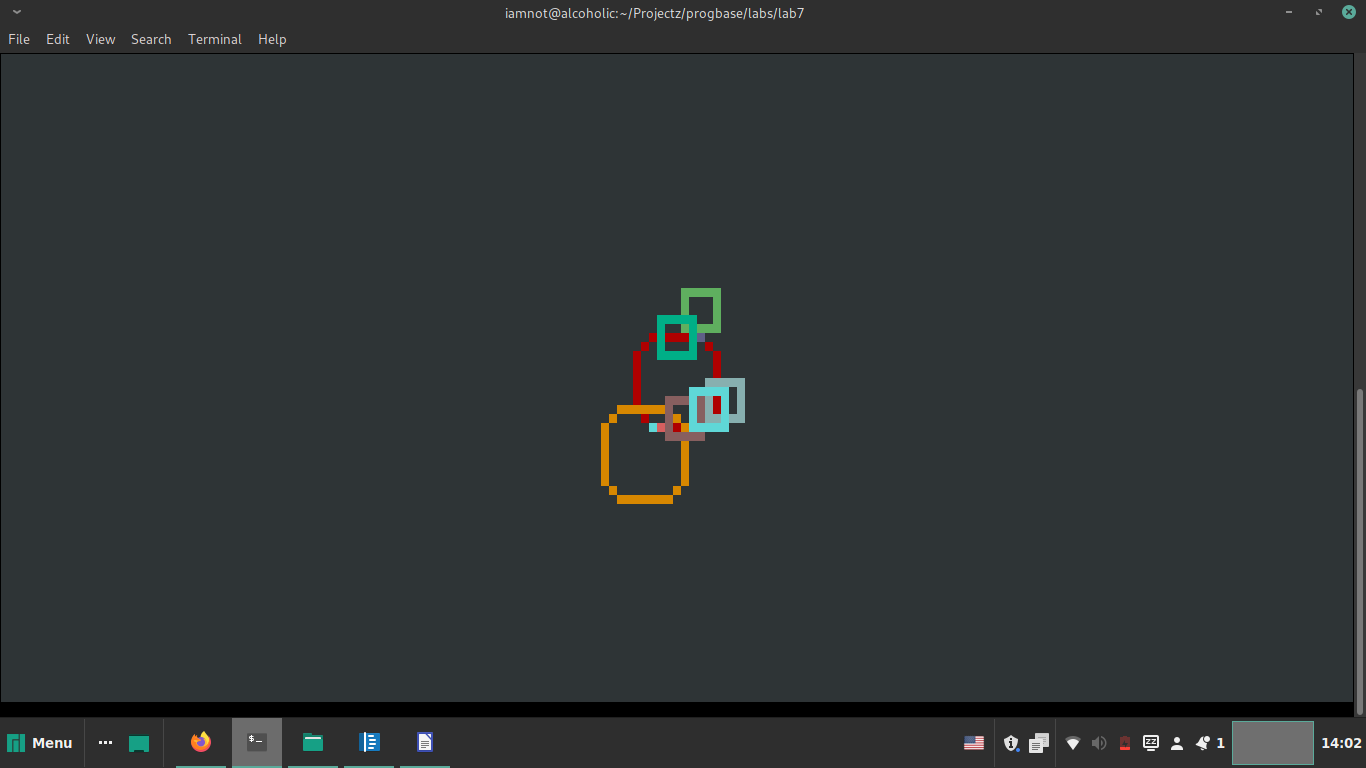
* Ball A
* Ball B
* Ball C

**Тексти коду програм**

|  |
| --- |
| **Lab7.c** |
| #include <math.h>  #include <progbase.h>  #include <progbase/canvas.h>  #include <progbase/console.h>  #include <stdio.h>  #include <stdlib.h>  #include <time.h>  **int** **RandInt**(**int** min, **int** max) { **return** (rand() % (max - min + **1**) + min); };  **struct** Vec2D {  **int** x;  **int** y;  };  **struct** Color {  **int** R;  **int** G;  **int** B;  };  **struct** Ball {  **struct** Vec2D loc;  **int** radius;  **int** speed;  };  **int** **main**() {  **const** **float** pi = **3.1415**;  srand(time(**0**));  **int** N = **5**;  printf("Enter N: ");  scanf("%i", &N);  **struct** Ball A;  **struct** Ball B[N];  **struct** Ball C[N];  **struct** ConsoleSize cs = Console\_size();  **int** w = cs.columns;  **int** h = cs.rows \* **2**;  A.loc.x = w / **2**;  A.loc.y = h / **2**;  **float** angle = **2** \* pi / N;  **float** rotAngle;  **if** (w < h) {  A.radius = w / **4**;  } **else** {  A.radius = h / **4**;  }  A.radius = rand() % A.radius + **5**;  **float** step = **1**;  // colors  **struct** Color colA;  **struct** Color colB[N];  **struct** Color colC[N];  colA.R = rand() % **255**;  colA.G = rand() % **255**;  colA.B = rand() % **255**;  // float move = 0.01;  **int** S = A.radius / N;  **if** (S < **2**) {  S = **2**;  }  **for** (**int** i = **0**; i < N; i++) {  colB[i].R = rand() % **255**;  colB[i].G = rand() % **255**;  colB[i].B = rand() % **255**;  colC[i].R = rand() % **255**;  colC[i].G = rand() % **255**;  colC[i].B = rand() % **255**;  B[i].radius = **3** + rand() % (S);  C[i].radius = **3** + pi \* (rand() % (B[i].radius / **2**));  C[i].speed = rand() % (**11**) - **5**;  **if** (C[i].speed < **0**) {  C[i].speed -= **3**;  } **else** {  C[i].speed += **3**;  }  B[i].speed = rand() % (**11**) - **5**;  **if** (B[i].speed < **0**) {  B[i].speed -= **2**;  } **else** {  B[i].speed += **2**;  }  }  Console\_clear();  **while** (**1**) {  step += **0.01**;  **for** (**int** i = **0**; i < N; i++) {  rotAngle = angle \* i;  B[i].loc.x = (A.loc.x + A.radius \* cos(rotAngle + step \* B[i].speed));  B[i].loc.y = (A.loc.y + A.radius \* sin(rotAngle + step \* B[i].speed));  C[i].loc.x =  (B[i].loc.x + B[i].radius \* cos(rotAngle + step + C[i].speed));  C[i].loc.y =  (B[i].loc.y + B[i].radius \* sin(rotAngle + step + C[i].speed));  sleepMillis(**1**);  }  Canvas\_setSize(w, h);  Canvas\_beginDraw();  Canvas\_setColorRGB(colA.R, colA.G, colA.B);  Canvas\_strokeCircle(A.loc.x, A.loc.y, A.radius);  **for** (**int** i = **0**; i < N; i++) {  Canvas\_setColorRGB(colB[i].R, colB[i].G, colB[i].B);  Canvas\_putPixel(B[i].loc.x, B[i].loc.y);  Canvas\_setColorRGB(colC[i].R, colC[i].G, colC[i].B);  Canvas\_strokeCircle(C[i].loc.x, C[i].loc.y, C[i].radius);  }  Canvas\_endDraw();  }  Console\_setCursorPosition(h + **2**, **2**);  **return** **0**;  } |

**Приклади результатів**

****

****

**Висновки**

Виконавши дану лабораторну роботу я навчився використовувати масиви структур даних, малювати за допомогою модуля canvas з бібліотеки libprogbase,  
Також навчився використовувати вказівники на структури та масиви структур даних у користувацьких функціях

Компіляція всього коду відбувалася за допомогою компілятора gcc.