

# Гаригийн хөдөлгөөнийг симуляц хийх (Лабораторийн Ажил №12)

Багийн нэр: Шадар гурван цэрэг

(Багийн ахлагч)

МТЭС, МКУТ-ийн Программ Хангамжийн III түвшний оюутан, М. Өсөхбаяр  
[21b1num0380@stud.num.edu.mn](mailto:21b1num0380@stud.num.edu.mn)

(Гишүүд)

МТЭС, МКУТ-ийн Программ Хангамжийн III түвшний оюутан, Э. Уранчимэг  
[21b1num0609@stud.num.edu.mn](mailto:21b1num0609@stud.num.edu.mn)

МТЭС, МКУТ-ийн Мэдээллийн системийн III түвшний оюутан, Т. Жавхлантөгс  
[21b1num2200@stud.num.edu.mn](mailto:21b1num2200@stud.num.edu.mn)

## 1. ОРШИЛ

Энэхүү лабораторийн ажлаар хэд хэдэн биетүүдийн хоорондох таталцлыг симуляц хийх программыг C++ хэл дээр өгөгдсөн зааврын дагуу Body, Universe гэх классуудыг үүсгэн хэрхэн хэрэгжүүлсэн талаар унших боломжтой. Мөн онолын судалгаа хэсэгт хэрэгжүүлэлтэд ашиглагдсан томьёо, шинэ ойлголтуудыг тайлбарласан.

## 2. ЗОРИЛГО

Энэ удаагийн тайлангийн зорилго нь өгөгдсөн зааврын дагуу гараг эрхэсийн хөдөлгөөнийг таталцлын хууль ашиглан бичсэн кодын хэрэгжүүлэлтийг тайлбарлах явдал юм. Үүний тулд дараах зорилтуудыг тавьж ажилласан:

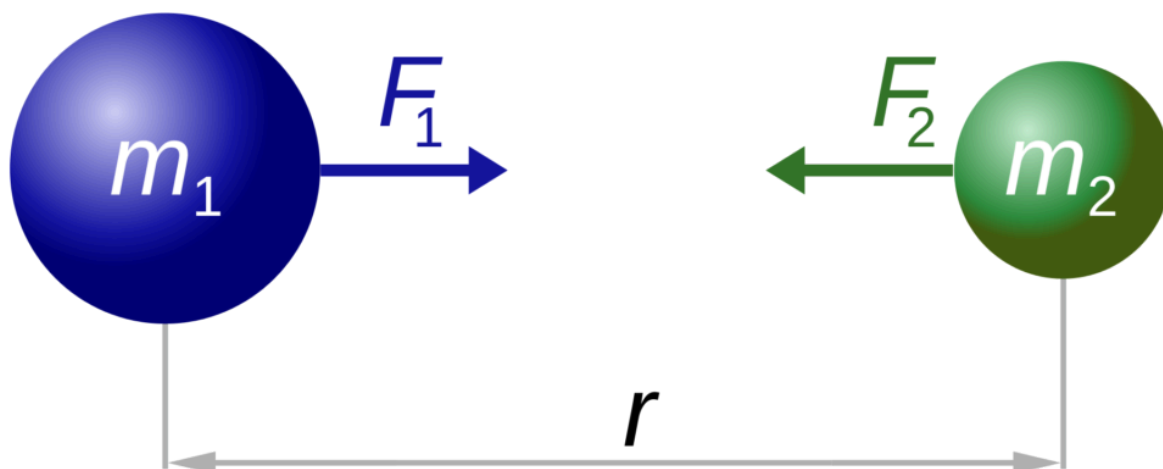
1. Бүх ертөнцийн таталцлын хуулийн талаар судалгаа хийх
2. Хоёр цэгийн хоорондох зай олох томьёог судлах
3. Файлын зориулалттай функцын талаар судалгаа хийх
4. Body, universe класс байгуулах
5. Нарны аймгийн гарагуудын объектыг байгуулах
6. Кодыг цэгцтэй, зөв форматтай бичих

## 3. ОНОЛЫН СУДАЛГАА

### 3.1. Бүх ертөнцийн таталцлын хууль

Байгаль дээр байгаа аливаа масстай биеүүдын хоорондох харилцан үйлчлэлийг таталцлын харилцан үйлчлэл гэнэ. Энэ харилцан үйлчлэлийг илэрхийлж байгаа хэмжигдэхүүнийг *бүх ертөнцийн таталцлын хууль* гэнэ. Түүнийг ажиглаж буй туршилт, тохиолдол доош унах үзэгдэл юм. Бүх ертөнцийн таталцлын харилцан үйлчлэлийг судалж хууль ба томъёоллыг

нээж гаргасан эрдэмтэн бол Исаак Ньютон юм. Бүх ертөнцийн таталцлын хууль нь харилцан таталцаж байгаа 2 биеийн харилцан үйлчлэлийн хүч нь 2 биеийн массын үржвэрээс шууд, хоорондох зайнаас урвуу хамааралтай.



$$F_1 = F_2 = G \frac{m_1 \times m_2}{r^2}$$

Зураг 1.Таталцлын хуулийн томъёо

### 3.2. Хоёр цэгийн хоорондох зай олох

Хоёр цэгийн хоорондох зайг тооцоолох томъёо нь хоёр хэмжээст эсвэл гурван хэмжээст орон зайд байгаа хоёр цэгийн координатыг ашигладаг. Хоёр цэгийн хоорондох зай нь харгалзах координатуудын ялгаврын квадратуудын нийлбэрээс квадрат язгуур авсантай тэнцүү байдаг ба Пифагорын теоремийн нэг үндсэн хэрэглээ юм.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

### 3.3. Файлаас тэмдэгт унших

C++ хэл дээр өгөгдлийн урсгал (stream) гэдэг нь компьютерын ой болон оролт/гаралтын төхөөрөмжүүдийн хооронд шилжсэн тэмдэгтүүдийн дарааллыг хэлнэ. Өгөгдлийн урсгалтай хамаатай бүх класс fstream.h толгой файл дотор тодорхойлогддог. Энэ нь файл үүсгэх, бичих, унших боломжийг олгодог ба лабораторын ажлын хүрээнд файлаас тэмдэгт мөрийг унших функцыг хэрэгжүүлэн судлав.

Жишээ :

```
#include<fstream>//файлтай ажиллах толгой файлаа дуудах
```

```
int main(){
    const int max = 40;
    char str[max]; //текст уншиж авах саваа бэлдэнэ
    Ifstream ipfile("nbody.txt"); //файлаас мөр тэмдэгт унших ifstream классын объектыг
    while (ipfile){                                     байгуулах
        ipfile.getline(str,max); //getline функцээр тэмдэгт мөрийг уншиж авах
    }
    getch();
}
```

## 4. ХЭРЭГЖҮҮЛЭЛТ

- **Хийх ажил:** Гаригийн хөдөлгөөнийг симуляц хийх

### 1. Body класс үүсгэх

- name, x, y, vx, vy, mass гэсэн гишүүн өгөгдөл болон body хоорондын зайг олох, force олох, байрлал шинэчлэх гишүүн функцуудыг агуулсан классыг үүсгэнэ.

### 2. Universe класс үүсгэх

- body хадгалах зориулалт бүхий bodies сав үүсгэх ба тэрхүү савандаа body-г нэмэх функц, тэрхүү саванд байрлах body-нуудын force-г олон байрлалыг шинэчлэх simulate функц, тэрхүү simulate-г дэлгэцлэх display гэсэн гишүүн функцүүдийг агуулсан классыг үүсгэнэ.

### 3. Text file унших

- Text file-с мэдээллийг уншин Body-нуудаа үүсгэнэ.

```
Lab 12 > C++ > ≡ lab12.txt
1  S 500 500 0 0 100000000
2  P 255 255 2 0 1
3  A 745 745 -2 0 2
4  M 270 740 -1.5 -3 12000
5  Q 253 450 0 -2 50000
6  R 500 0 3 0 1000000
7  V 504 0 4 4 0.000001
```

Lab12.txt файл

## 5. ДҮГНЭЛТ

Энэхүү лабораторийн ажлаар өмнөх үзсэн мэдлэгүүдээ ашиглан хэд хэдэн биетүүдийн хоорондох таталцлыг симуляц хийж C++ хэл дээр хэрэгжүүллээ. Даалгаврыг гүйцэтгэх явцад Body болон Universe гэсэн хоёр класс үүсгэж, Body класст тухайн биетийн байршил, масс, хурдатгал зэрэг мэдээллүүдийг авч, Universe класст тэдгээрийн векторуудыг хадгалж, симуляцийг удирдсан. Мөн шаардлагын дагуу нэмэлт функцүүдийг хэрэгжүүлсэн.

### 5.1. Багийн гишүүн бүрийн оролцоо

**М. Өсөхбаяр** - Онолын судалгааг хийж, багийн ерөнхий зохион байгуулалт хийж, уулзах тов зарлан, ажлуудыг хуваарилсан, код бичсэн ба дүгнэлт дээр ажилласан.

**Э. Уранчимэг** - Онолын судалгааг хийж, тайлангийн оршил, зорилго, судалгаан дээр ажилласан, багийн гишүүдийн кодыг хянасан бөгөөд тайланг хянан засварласан.

**Т. Жавхлантөгс** - Онолын судалгаа хийж, гол функцууд дээр ажилласан, багийн гишүүдийн кодыг хянасан ба тайланг хянан засварласан.

### 5.2. Багаар ажиллахад тулгарсан асуудал

**Асуудал:**

- Ойлголтын асуудал
- Цагийн менежментийн асуудал
- Ажлын хуваарилалт

**Шийдвэрлэсэн арга зам:**

- Гишүүн бүр тус тусдаа онолын судалгаа хийж дараа нь уулзан, ярилцаж нэгэн жигд ойлголттой болсон.
- Өмнө нь уулзсан цагтаа уулзаж, биеэр уулзах боломжгүй үед онлайнаар ярилцаж нэг doc файл дээр хамт тайлангаа бичин ажилласан.
- Өмнөх лабораторийн ажил дээр багийн ахлагч голлон үүрэг гүйцэтгэсэн гэж шүүмжлүүлсэн тул энэхүү ажил дээр бүгд жигд, чадах ажлуудаа хуваарилан авч, нэг нь нөгөөгийнхөө дутууг нөхөх замаар ажилласан.

### 5.3. Багаар ажиллаад сурсан зүйлс

- Багаар ажилласнаар бусдаасаа суралцаж, өөрийн алдаан дээр ажиллаж боломжийг илүү олгосон. Багийн гишүүдийн оролцоотойгоор илүү боловсронгуй, код тайлан бичигдсэн.
- Мөн багаар ажиллах үед нэг нэгнийхээ сул талыг нөхөж нөгөө нь бусдыхаа давуу тал нь болж ажиллахыг гол зорилтоо болгосноор даалгаврыг хурдан чанартай биелэгдэх боломжтой болсон.

- Нэгэндээ хэтэрхий найдах нь бас хор уршигтай ба өөрт ноогдсон үүргийг хариуцлагатайгаар гүйцэтгэж, нэг нь алдсан бол алдааг нь засалцаж, сайн хийсэн бол нэгнээ дэмжих чухал болохыг ойлгосон.
- Багийн ажил дээр мэдээлэл цуглуулах болоод мэдээлэл дээр дүн шинжилгээ хийх , дүгнэлт гаргах зэрэг асуудлууд дээр хамтын шийдлийг олж асуудлыг шийдэх нь илүү зөв зохистой хийгээд үнэн зөв хариултыг гаргадаг юм байна.

## 6. АШИГЛАСАН МАТЕРИАЛ

1. Объект хандлагат программчлал хичээлийн лекцийн материал, М.Золжаргал
2. Объект хандлагат технологийн C++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар.
3. <https://www.geeksforgeeks.org/data-structures/linked-list/>

## 7. ХАВСРАЛТ

```
1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4  #include <cmath>
5  #include <cstring>
6  #include <unistd.h>
7  #include <stdlib.h>
8
9  using namespace std;
10
11  const double GRAVITATIONAL_CONSTANT = 0.00005;
12
13  class Body
14  {
15  public:
16      string name;
17      double x, y;    // Байршил
18      double vx, vy; // Velocity
19      double mass;
20
21      //Байгуулагч
22      Body()
23      {
24          name = "null";
25          x = 0;
26          y = 0;
27          vx = 0;
28          vy = 0;
29          mass = 0;
30      }
31      //Параметртэй байгуулагч
32      Body(string n, double init_x, double init_y, double init_vx, double init_vy, double m)
33      {
34          name = n;
35          x = init_x;
36          y = init_y;
37          vx = init_vx;
38          vy = init_vy;
39          mass = m;
40      }
41  }
```

```

42 //Зай олох функц
43 double distance(Body a, Body other)
44 {
45     return sqrt((a.x - other.x) * (a.x - other.x) + (a.y - other.y) * (a.y - other.y));
46 }
47
48 //Force олох функц
49 void calculateForce(Body a, Body other)
50 {
51     double d = distance(a, other);
52     double force = GRAVITATIONAL_CONSTANT * mass * other.mass / (d * d);
53
54     double fx = force * (other.x - x) / d;
55     double fy = force * (other.y - y) / d;
56
57     double ax = fx / mass;
58     double ay = fy / mass;
59
60     vx += ax;
61     vy += ay;
62 }
63
64 //Байрлал шинэчлэлт
65 void updatePosition()
66 {
67     x += vx;
68     y += vy;
69 }
70 };
71
72 class Universe
73 {
74     public:
75     vector<Body> bodies;
76
77     //bodies-д body нэмэх функц
78     void addBody(Body body)
79     {
80         bodies.push_back(body);
81     }
82

```

```

83 //simulate хийх функц
84 void simulate()
85 {
86     //body хоорондын force
87     for (size_t i=0; i<bodies.size(); i++)
88     {
89         for (size_t j=0; j<bodies.size(); j++)
90         {
91             if (i != j)
92             {
93                 bodies[i].calculateForce(bodies[i],bodies[j]);
94             }
95         }
96     }
97
98     for (size_t i = 0; i < bodies.size(); i++)
99     {
100         bodies[i].updatePosition();
101     }
102 }
103
104 //Дэлгэц рүү хэвлэх функц
105 void display()
106 {
107     // body-ны name хадгалах болоод дэлгэцлэхэд хэрэглэгдэх сав
108     string A[100][150];
109
110     for(int i=0; i<100; i++)
111     {
112         for(int j=0; j<150; j++)
113         {
114             A[i][j]=" ";
115         }
116     }
117
118     //савандаа body-г байрлуулах давталт
119     for(size_t i = 0; i < bodies.size(); i++)
120     {
121         int xx,yy;
122         yy = int(bodies[i].x);
123         xx = int(bodies[i].y);
124         A[xx/20][yy/20]=bodies[i].name;
125     }

```



```

127         //дэлгэцлэх давталт
128         for(int i=0; i<100; i++)
129         {
130             for(int j=0; j<150; j++)
131             {
132                 cout<<A[i][j];
133             }
134             cout<<endl;
135         }
136     }
137 };
138 int main()
139 {
140     Universe universe;
141     ifstream file("/Users/javkhlantugst/Desktop/NUM/Semester 05/Объект хандлагат програмчлал/Laboratory/Lab 12/C++/lab12.txt");
142
143     string name;
144     double x, y, vx, vy, mass;
145     Body a;
146     while (file >> name >> x >> y >> vx >> vy >> mass)
147     {
148         a.name = name;
149         a.x = x;
150         a.y = y;
151         a.vx = vx;
152         a.vy = vy;
153         a.mass = mass;
154
155         universe.addBody(a);
156     }
157     file.close();
158
159     // Дэлгэцлэх
160     while(true)
161     {
162         //delay
163         usleep(10000);
164         //Дэлгэц цэвэрлэх байрлалыг шинээр байрлуулах
165         system("clear");
166         universe.simulate();
167         universe.display();
168     }
169 }

```