Анимейшн буюу хөдөлгөөнийг үүсгэх

Агуулга

- OpenGL, GLUT тусламжтай хөдөлгөөн хийх
- Давхар буфер
- Урлаг, анимейшн дахь “Tweening” хэрэглээ

Фрейм гэж юу вэ?

- Ямар нэг үйл хөдлөл, явцыг харуулсан дараалсан зургууд маш хурдан солигдоход хүн тэндээс хөдөлгөөнийг олж хардаг.
- Хугацааны тухайн агшин дахь өрнөлийг үзүүлсэн зургийг фрейм/кадр гэж ярьдаг.
- Өргөн нэвтрүүлгийн стандарт нь 1 секундэд 60 фрейм/кадр байдаг. Үүнийг 60 fps буюу frame per second гэдэг.
- Фреймүүдийг уламжлалт фото зураг, компьютер график ашиглан зохиомолоор үүсгэж болно.

Энгийн анимейшн

- Хөдөлгөөний дарааллыг хялбархан үүсгэх боломжтой үзэгдэл боловсруулах функцүүд GLUT санд байдаг.
- Бидний зурсан дүрс цонхонд хэрхэн үсрэх хөдөлгөөнийг жишээгээр үзье.

Анимейшн

- Тодорхой хугацааны интервалтайгаар хэрэглэгчийн функцийг дуудаж ажиллуулна.

```
void glutTimerFunc(
        unsigned int msecs,
        void (*func)(int value),
        int value)
```

  - value параметрээр хэрэглэгчийн тодорхойлсон утгыг мөн дамжуулж болно.

Animated bouncing square

```
#include <windows.h>
#include <gl/glut.h>

//Initial square position and size
GLfloat x1=100.0f;
GLfloat y1=150.0f;
GLsizei rsize=50;

//Step size in x and y directions
GLfloat xstep=1.0f;
GLfloat ystep=1.0f;

GLfloat winWidth, winHeight;
```

//Main program entry point

```
void main (void) {
    glutInitDisplayMode(GLUT_DOUBLE |
        GLUT_RGB);
    glutCreateWindow("Bounce");
    glutDisplayFunc(display);
    glutReshapeFunc(resize);
    glutTimerFunc(33, timer, 1);
    setup();
    glutMainLoop();
}
```

```
void display(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0f, 0.0f, 0.0f);
    glRectf(x1, y1, x1+rsize, y1+rsize);
    //Flush drawing commands and swap
    glutSwapBuffers();
}
void setup(void) {
    glClearColor(0.0f, 0.0f, 1.0f, 1.0f);
}
```

```
void timer(int value) {
    if (x1>winWidth-rsize || x1<0)
        xstep=-xstep;
    if (y1>winHeight-rsize || y1<0)
        ystep=-ystep;
    if (x1>winWidth-rsize)
        x1=winWidth-rsize-1;
     if (y1>winHeight-rsize)
        y1=winHeight-rsize-1;
    x1+=xstep;
    y1+=ystep;
    glutPostRedisplay();
    glutTimerFunc(33, timer, 1);
}
```

```
void resize (GLsizei w, GLsizei h) {
    if (h==0) h=1;
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if (w<=h) {
        winHeight=250.0f*h/w;
        winWidth=250.0f;
    } else {
        winWidth=250.0f*w/h;
        winHeight=250.0f;
    }
    glOrtho(0.0f, winWidth, 0.0f, winHeight, 1.0f, -1.0f);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
```

Давхар буфер

- Графикын ямар ч багцын хамгийн чухал зүйлсийн нэг нь давхар буфер юм.
- GLUT сан давхар буферт цонх (double-buffered window) –ийг дэмждэг.

Давхар буфер

- Үндсэн 2 зорилгоор ашиглагддаг.
  - Зарим төвөгтэй зүйлсийг зурахад хугацаа их зарцуулах бөгөөд зурж буй алхам бүрийг харуулах шаардлагагүй байж болно. Энэ тохиолдолд давхар буфер ашиглан зураг дүрслэлийнхээ нэгтгээд дууссан эцсийн үр дүнг харуулах боломжийг олгодог.
  - Хөдөлгөөнт зураг/анимейшн үүсгэх зорилгоор Фрейм бүрийг идэвхгүй буюу off screen буфер рүү зурж хадгалж яваад бэлэн болмогц нь дэлгэц рүү сольж гаргах

Давхар буфер

```
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
. . .

//Flush drawing commands and swap
glutSwapBuffers();
```

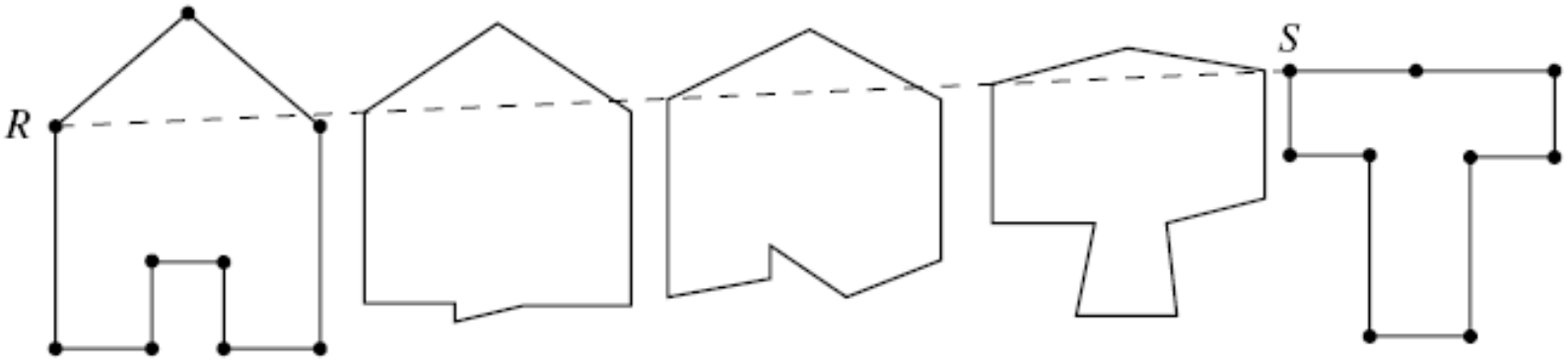
Дээрх өөрчлөлтүүдийн үр дүнд тэгш өнцөгтийн цонхон дээр үсрэх жигд хөдөлгөөн үүснэ.

Урлаг ба анимейшн дэх tweening хэрэглээ

- Нэг дүрсийг нөгөө рүү хувиргах замаар сонирхолтой анимейшн үүсгэж болно.

Tweening

- Хэрэв ижил тооны цэгүүдээс тогтох дараах хоёр дүрс өгөгдсөн гэвэл
  - Байшинг Т болгон хувиргах процесс



Tweening

- Эхний дүрс- **A**: A_i
- Хоёр дахь- **B**: B_i

$i=0, \ldots, n-1$

P(t) муруйг бид дараах томъёогоор олж чадна:

$$P_i(t) = (1-t)A_i+B_it$$

Tweening

- t нь [0, 1] завсарт утгатай гэвэл

  t=0, 0.1, 0.2, …, 0.9, 1.0

  А дүрсээс эхлэн хувирсаар В дүрсэд шилжих бөгөөд завсарын алхамуудад эдгээр 2 дүрсийн дундаас үүссэн холимог дүрс гарна.

Tweening

- t-ийн бага утганд А дүрстэй төстэй, t-ийн утга өсөхийн хэрээр илүү В-тэй төстэй дүрс үүснэ.

Tweening and double buffering

```
//tween back and forth forever
for(t=0.0, delT=0.1; ; t+=delT) {
    clear the buffer
    drawTween (A, B, n, t);
    glutSwapBuffers();
    if (t >=1.0 || t <=0.0) delT = -delT;
}
```

Tweening

- Энэ аргыг ашигласнаар хүүхэлдэйн кино/cartoon бүтээх өртгийг бууруулах бололцоотой болсон.
- Дээхнэ үед 1 секундын хөдөлгөөн хийхийн тулд зураач 24 зураг бэлдэх шаардлагатай болдог байсан.
- Компьютерийн тусламжтайгаар зураач эхний болон төсгөлийн зургуудыг буюу **key frame зургуудыг** бэлдэхэд хангалттай болсон бөгөөд бусад завсарын зургийг компьютерээр автоматаар үүсгэдэг болсон.

Sample

- A = (4, 9)
- B = (3, 7)

- Tween(A, B, t) =

  ((1-t)4 + 3t, (1-t)9 +7t) =

  (4 – t , 9 – 2t)

- Tween(A, B, 0.4) = (3.6, 8.2)

Animation