

Агуулга

Компьютер график ба түүний боломж.....	2
Компьютер график гэж юу вэ?	2
3D дүрслэл 2D дүрслэлээс юугаараа ялгагдах вэ?	3
Графикийн нээлттэй сан OpenGL	3
Толгой файл:	4
OpenGL функцийн формат	4
OpenGL өгөгдлийн төрөл	4
Графикийн анхны жишээ програм	7
Бодит ба дэлгэцийн координатын систем.....	9

Компьютер график ба түүний боломж

Өнөөдөр компьютерээр бүтээсэн зарим зураг фото зурагнаас ялгагдахааргүй бодитой үнэмшилтэй болсон. Үүнийг фото реалистик гэж ярьж хэлцдэг болсныг дуулсан байх. Кино урлаг, видео тоглоомын хөгжилд хүчтэй түлхэлт өгсөн салбарын нэг бол компьютер графикын салбар юм. Үнэндээ, компьютер графикт гарсан ололт, дэвшил нь визуал медиа урлагийг бодит байдлын хүлээснээс чөлөөлж өгсөн бөгөөд өнөөгийн түвшинд компьютерээр бүтээсэн анимейшн нь кино урлаг, телевизийн уламжлалт дүрслэл, жүжиглэлт, жүжигчинг бүрэн орлохоор бололцоотой болсон ч гэж хэлж болно.

Компьютер график гэж юу вэ?

Хүн, хүний уураг тархи тоо үсгэн мэдээллээс илүүтэй зурган мэдээллийг хурдан хүлээж авдаг. Хамгийн товчоор бол компьютерээр бий болгосон зураг дүрслэлийг ерөнхийд нь компьютерийн график гэж ойлгож болно. Хүмүүс компьютер графикыг хэд хэдэн шалтгааны улмаас сонирхон судалдаг. Үүнд: судалгаа, шинжилгээнийхээ өгөгдөл үр дүнг дүрсэлж харуулах сайн хэрэгсэлүүдийг хүсдэг байхад зарим нь компьютерийн тоглоом бичих хүсэлтэй бол өөр бусад нь урлагийг илэрхийлэх шинэ хэрэгслийг эрэлхийлж байна. Эцсийн эцэст хүн бүр бүтээлчээр өөрсдийн санаа бодлоо солилцох, илэрхийлэхийг эрмэлздэг бөгөөд компьютер график энэ бүхэнд сайн туслагч болдог. Компьютер графикын хэрэглээг бид өөрсдийн өдөр тутмын амьдралын олон жишээнээс харж болно. Тухайлбал, ном сэтгүүлийн эх бэлтгэл, телевизийн зар сурталчилгааны шторк, хүүхэлдэйн кино, компьютер тоглоом гэх мэт.

Компьютер графикын програмуудыг үндсэндээ тусгай зориулалтын ба ерөнхий зориулалтын гэж 2 хуваадаг. Тусгай зориулалтын програм гэдэгт бидний сайн мэдэх Microsoft Windows-ийн Paint, барилгын зураг төсөл, инженерийн зураг төсөл хийхэд зориулагдсан CAD, 3dsMax, график дизайнерүүдийн түгээмэл хэрэглэдэг Photoshop, CorelDraw-г дурьдаж болно. Тусгай зориулалтын програм нь хэрэглэгч ашиглах үүрэг бүхий мени, багаж хэрэгслээр тоноглогдсон байдаг. Тэгвэл ерөнхий зориулалтын програмыг C/C++, Java, Python гэх мэт програмчлалын хэл дээр графикын сан ашиглан үүсгэдэг.

Энэ хичээлээр бид графикын нээлттэй сан OpenGL-ийг ашиглан C хэл дээр графикын програм бичиж сурна.

Компьютер графикт хэрэглэгддэг маш олон нэр томъёо, нэршил байдаг. Бүх нэршилд оноосон монгол нэр байхгүй ч бид утгачилан орчуулах, англи үг хэллэгийг шууд хэрэглээд явж байгаа. Тухайлбал,

real time - бодит хугацаа (хоёр хүн хоорондоо утсаар ярих үйлдэл нь бодит хугацаанд хийгдэж байгаа бол захиа бичих нь real time процесс биш юм)

animation – хөдөлгөөн / хөдөлгөөнт зураг (анимейшн)

modeling – загвар/загварчлал

visualization – төсөөлөл

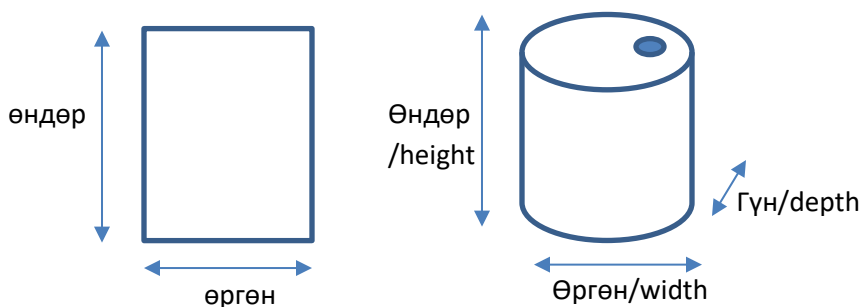
3D – гурван хэмжээст

Perspective - алслалт

3D дүрслэл 2D дүрслэлээс юугаараа ялгагдах вэ?

Зураачид хавтгай (цаасан) дээр хэрхэн юмсыг (биетийг) бодит болгон дүрслэхийг мэддэг. Компьютер графикт, бид ч мөн ялгаагүй хавтгай (дэлгэцэн) дээр 3D бодит бодит дүрслэлийг хэрхэн бий болгохыг үзнэ.

Үнэндээ, 3D компьютерийн график нь хавтгай дэлгэцэн дүрсэлсэн 2 хэмжээст дүрс байх ба 3 дахь хэмжээс буюу гүний хуурмаг дүрслэл юм.



2D + Perspective = 3D

Алслалтын зарчим: ойр орших дүрс хол орших дүрсээс том дүрслэгдэнэ.

Графикын нээлттэй сан OpenGL

OpenGL нь үйлдлийн системүүдээс хамааралгүй өөрөөр хэлбэл бүх төрлийн платформуудад (Linux, Mac OS, Windows ...) ажилладаг графикын програмын стандарт юм. Түүнд 3 хэмжээст график дүрс зурахад зориулагдсан 200 гаруй функцүүд багтдаг бөгөөд гэрэл, сүүдэр, материал / текстур, визуал эффект үүсгэх зориулалт бүхий функцүүд байдаг боловч аливаа үйлдлийн системээс хамааралтай үйлдэл гүйцэтгэх үүрэг бүхий функцүүд байхгүй. Тиймээс тухайн үйлдлийн системээс хамааралтай цонх зурах, цонх үүсгэх, хаах, цонхонд үүсгэх

үзэгдэл боловсруулах гэх мэт үйлдлийг гүйцэтгэхдээ GLUT/freeGLUT санг нэмэлтээр ашиглана.

OpenGL + GLU (OpenGL Utility Library)

GLUT (OpenGL Utility Toolkit)

Графикын сан бүр өөрийн толгой файл, сангуудтай. (Графикын орчинг хэрхэн бүрдүүлэхийг Лаб ажил 1-д заасан байгаа.)

Толгой файл:

```
#include <GL/gl.h>
```

```
#include <GL/glu.h>
```

```
#include <GL/glut.h>
```

```
#include <GL/freeglut.h>
```

OpenGL функцийн формат

OpenGL сангийн функцүүд дараах хэлбэртэй бичигддэг.

```
<Library prefix><Root command><Optional argument count><Optional argument type>
```

Жишээ нь дараах функц нь бодит төрлийн 3 аргумент авах GL сангийн зурах өнгийг тодорхойлох функц юм.

glColor3f(...)

Тэгвэл энэ функцийн өөр хувилбарууд бас байж болно. Тухайлбал,

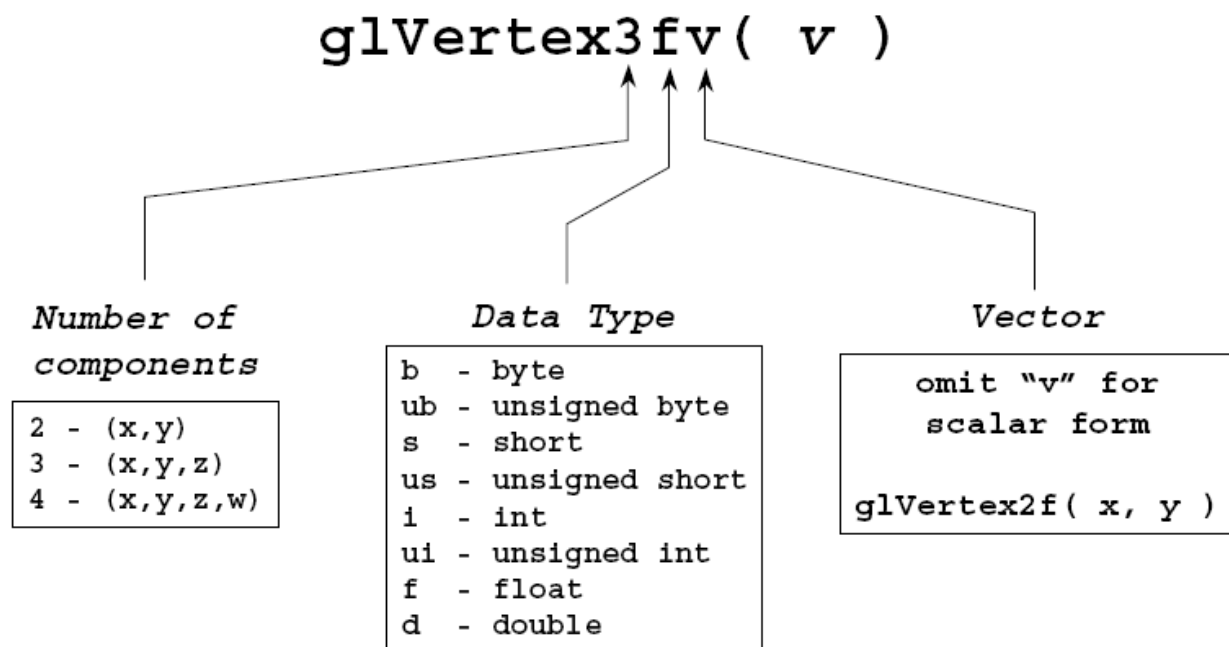
glColor3i, glColor3d гэх мэт.

OpenGL өгөгдлийн төрөл

OpenGL програмын кодыг нэгээс нөгөө платформ руу шилжүүлэхэд хялбар байлгах үүднээс өөрт нь зориулсан тусгай өгөгдлийн төрлүүдтэй. Гэхдээ C хэлний үндсэн өгөгдлийн төрлүүдийг OpenGL-ийн өгөгдлийн төрөлд мөн дүйцүүлэн хэрэглэж болно.

OpenGL Type	Minimum Number of Bits	Command Suffix	Description
GLboolean	1	NA	Boolean
GLbyte	8	b	Signed integer
GLubyte	8	ub	Unsigned integer
GLshort	16	s	Signed integer
GLushort	16	us	Unsigned integer
GLsizei	32	NA	Non-negative integer size
GLsizeiPtr	Number of bits in a pointer	NA	Pointer to a non-negative integer size
GLint	32	i	Signed integer
GLuint	32	ui	Unsigned integer
GLfloat	32	f	Floating point
GLclampf	32	NA	Floating point clamped to the range [0, 1].
GLenum	32	NA	Enumerant
GLbitfield	32	NA	Packed bits
GLdouble	64	d	Floating point
GLvoid*	Number of bits in a pointer	NA	Pointer to any data type; equivalent to "void*" in C/C++.

OpenGL сангийн цэгийн байршлыг тодорхойлох үүрэг бүхий glVertex() функц



Энэ функцийг ч мөн адил хэд хэдэн хувилбараар ашиглаж болно. Жишээ нь хавтгай дээр орших цэгийн байршлыг тодорхойлох үед

```
glVertex2f(1.0, 3.0);
glVertex2i (1, 3);
```

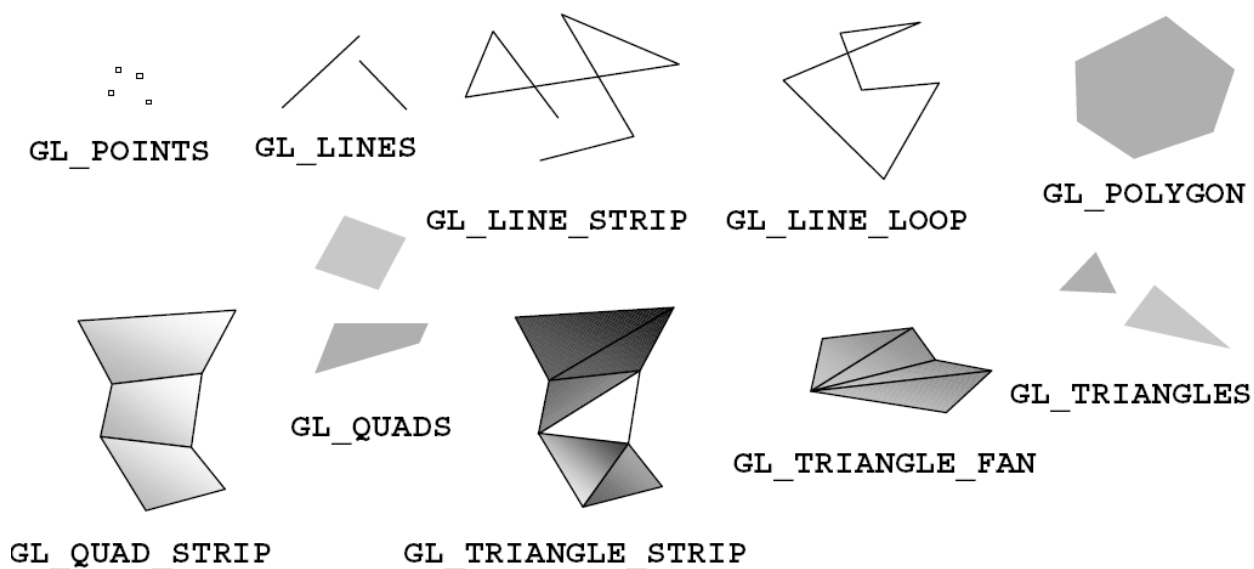
огторгуй дахь цэгийн байршлыг тодорхойлж буй үед

```
glVertex3f(1.0, 2.0, 3.0); //скаляр хэлбэртэй бичиглэл
```

эсвэл

```
GLfloat p[ ]={1.0, 2.0, 3.0};
glVertex3fv(p); //вектор хэлбэртэй бичиглэл
```

ашиглаж болно.



Тэгвэл дээрх тогтмолуудыг хэрхэн сонгон авснаас хамааруулан glVertex3f() функцээр огторгуйн цэгийн байршлыг цэгээр, цэгүүдийг холбон шулуунаар дүрслэх, тэр ч бүү хэл 3 ба түүнээс дээшхи оройнуудыг холбон битүүлж дүрс зурах боломжтой.

Графикын анхны жишээ програм

Бидний анхны програмд ашиглагдаж буй функцүүдийн угтварыг харвал аль сангийн функцүүд ашиглагдаж байгааг төвөггүй харж болно.

```
//simple.c

#include <GL/freeglut.h>

void display(void) {

    glClear(GL_COLOR_BUFFER_BIT);

    glFlush();

}

void setup(void) {

    glClearColor(0.0f, 0.0f, 1.0f, 1.0f);

}

int main(int argc, char** argv) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);

    glutCreateWindow("Simple");

    glutDisplayFunc(display);

    setup();

    glutMainLoop();

    return 0;

}
```

Орчин үеийн үйлдлийн системүүд цонхонд суурилсан системийг ашиглах болсон бөгөөд бидний зурж дүрслэх үйлдэл ч цонх руу хийгдэнэ. Тиймээс бидний хийх анхны үйлдэл шинэ цонх үүсгэх, нээхээс эхэлнэ.

```
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
```

GLUT сангийн энэ функцээр дэлгэцийн горимыг сонгоно. RGB (red, green, blue) 3 өнгөний нийлэмж бүхий өнгөний горимтой, дан буфертэй дэлгэц ашиглахаар тогтмолуудыг тодорхойлсон байна. Графикын командууд дэлгэцийн гадна гүйцэтгэгдэх (ө.х буферлэгдэж байгаад) ба дараа нь цонх руу гаргадаг.

```
glutCreateWindow("Simple");
```

Дэлгэцэнд цонх үүсгэх/зурах үйлдлийг хийнэ. Цонхны нэр буюу гарчгийг тодорхойлох нэг аргументтай функц.

```
glutDisplayFunc(display);
```

Энэ функцэд дуудагдсан хэрэглэгчийн функц цонх анх үүсэх, цонхны хэмжээ өөрчлөгдөх, цонх хураагдах зэрэг үзэгдлүүдийн үед дуудагдан ажилладаг.

```
setup();
```

Програм ажиллахад 1 л удаа хийгдэх анхны тохиргоонуудыг хийх зорилгоор үүсгэсэн хэрэглэгчийн функц.

```
glutMainLoop();
```

Энэ функц дуудагдсанаар GLUT сангийн функцүүд бодит утгаар ажиллаж эхэлдэг гэсэн үг. Өөрөөр хэлбэл, цонхонд үүсэх бүхий л үзэгдлүүд тухайлбал, цонхыг шинээр зурах, цонхыг хураах, цонхны хэмжээг ихэсгэх/багасгах, цонхон дээр хулганаар товчих үзэгдлүүдийг бүртгэн авч боловсруулна. Энэ функцийн дараа бичигдсан ямар ч үйлдэл хийгддэггүй болохыг анхаарах хэрэгтэй. Энэ функц нь програмыг дуусгах хүртэлх системээс ирсэн бүхий л мессеж, товчлууур дахь үйлдлийг боловсруулна.

```
glClearColor(0.0f, 0.0f, 1.0f, 1.0f);
```

функцэд өгөгдсөн өнгөөр цонхыг будна. (Red, green, blue, alpha) гэсэн 4 өнгөний нийлэмжээр өнгийг тодорхойлно. Энэ тохиолдолд цэнхэр өнгөөр цонхыг будна. Эдгээр нь 0.0 - 1.0 завсарт утгууд авна.

```
glClear(GL_COLOR_BUFFER_BIT);
```

Энэ функцээр нэг юмуу хэд хэдэн буферийн комбинацийг цэвэрлэнэ. Энэ тохиолдолд өнгөний буферийг цэвэрлэх үйлдэл хийж байна. Буфер гэдгийг зурган мэдээлэл дүрсэлж хадгалах муж гэж ойлгож болно.


```
glFlush();
```

OpenGL графикын хоолой буюу rendering pipeline ашигладаг бөгөөд энд графикын командуудыг дараалуулан биелүүлдэг. Өөрөөр хэлбэл, OpenGL-ийн командууд энэ хоолойд дараалалд байж байгаад хэд хэдэн хүсэлтийг хамтатган 1 удаа гүйцэтгэх зарчмаар ажилладаг. Ингэснээр графикын hardware рүү хандах хандалтыг цөөрүүлж өгдөг.

Одоо display () функцийг дараах байдлаар дахин тодорхойлж бичье.

```
void display(void) {
    glClear(GL_COLOR_BUFFER_BIT);

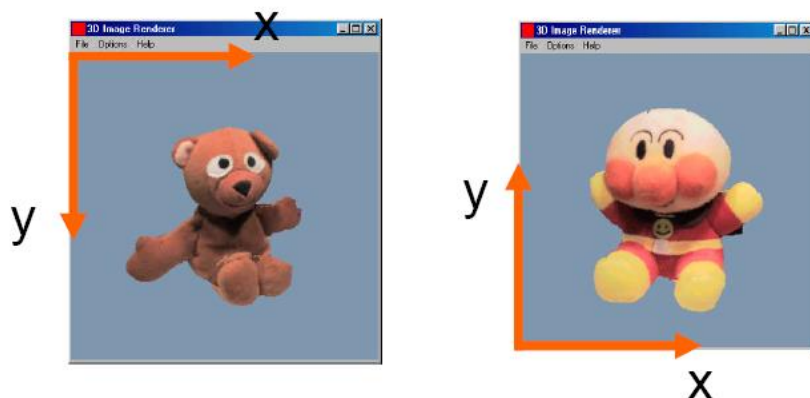
    glColor3f(1.0, 0.0, 0.0); //red color

    glRectf(100.0f, 150.0f, 150.0f, 100.0f); //top left, bottom right

    glFlush();
}
```

Бодит ба дэлгэцийн координатын систем

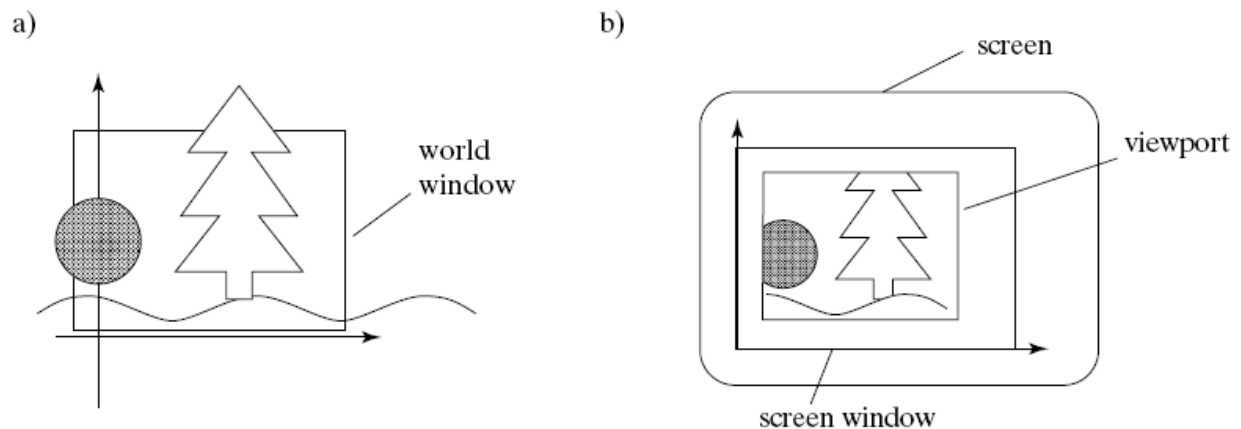
Юмс нэг нь нөгөөгөөсөө ямар зайд хаана хаана байрлахыг тодорхойлоход бидэнд координатын систем хэрэгтэй болно. Бидний мэдэх дэлгэцийн координатын систем пикселээр (pixel) хэмжигдэх бөгөөд х-ийн авах утгын хязгаар $[0 \div \text{screenWidth}-1]$; у-ийн авах утга $[0 \div \text{screenHeight}-1]$.



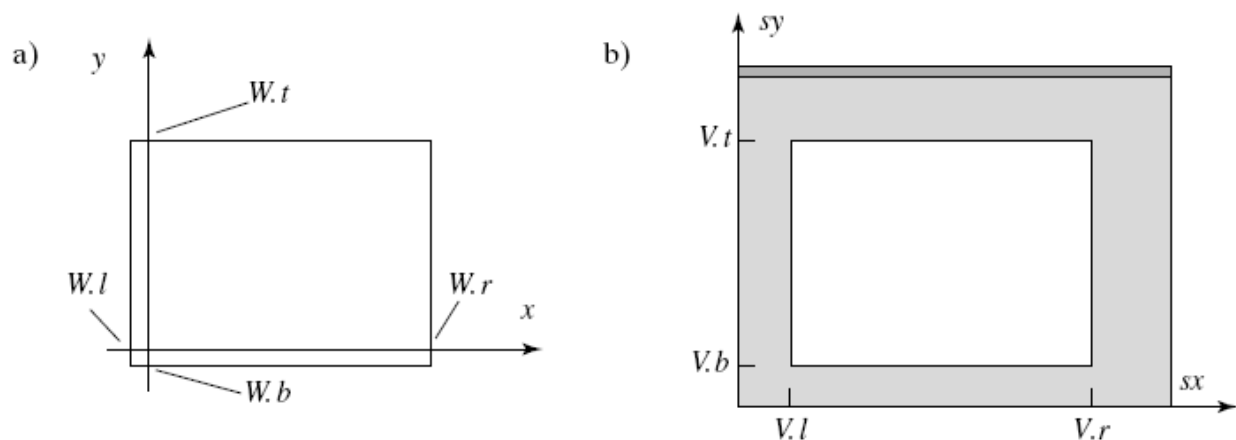
Энэ нь бид зөвхөн x, y-ын эерэг утгыг ашиглаж болно гэсэн үг болохнээ. Гэтэл өгөгдсөн бодлогын хувьд хангалтгүй бөгөөд хэрэв бид х-ийн авах утгын хязгаар $[-1; 1]$ байхад у-ийн авах утга $[-100; 200]$ завсаруудад тодорхойлогддог бол амьдралд

арай илүү нийцнэ. Иймд биетийг ийнхүү тодорхойлох, дүрслэх боломжтой координатын системийг бодит координатын систем (world/true coordinate system) гэж нэрлэе. Тэгш өнцөгт координатын систем бидний хувьд бодит координатын систем болж чадна.

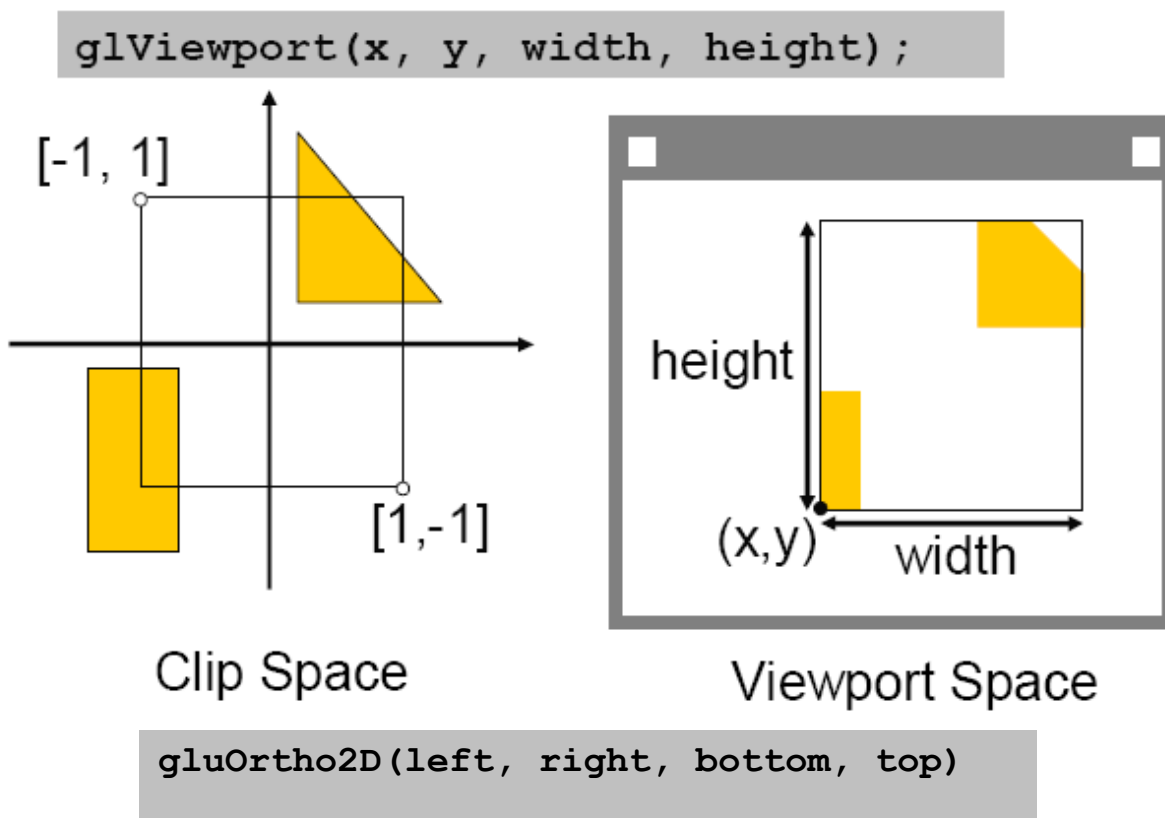
Бодит координатын системд тодорхойлогдсон тэгш өнцөгт мужийг “бодит цонх” гэж нэрлэе. Өөрөөр хэлбэл, бодит цонх бодит ертөнцийн аль хэсэг цааш зурагдахыг тодорхойлно гэсэн үг. Бодит цонхны гадна орших зүйлс тайрагдахгүй буюу зурагдахгүй. Зөвхөн бодит цонхонд багтсан хэсгийг бид дүрсэлж харуулна. Түүнчлэн, дэлгэцийн цонхонд мөн тэгш өнцөгт муж тодорхойлж түүнийг “viewport” буюу “харагдах муж” гэж нэрлэе.



Энэ нь цаашид бодит цонхыг viewport руу буулгах буулгалт болж хувирна.



OpenGL-д бодит цонхыг 2 хэмжээст тохиолдолд `gluOrtho2D()` функцээр, viewport-ыг `glViewport()`-ээр тус тус тодорхойлдог.



```
void gluOrtho2D ( GLdouble left, GLdouble right,
                  GLdouble bottom, GLdouble top);
```

Бодит цонхны хэмжээг тодорхойлох ба **left < x < right; bottom < y < top** байна гэсэн үг.

```
void glViewport ( GLint x, GLint y,
                  GLint width, GLint height);
```

Цонхны view portion буюу харах мужийг тодорхойлно. Цонхыг бүхэлд нь viewport болгон ашиглан тохиолдолд дараах байдлаар дуудаж ашиглана.

```
glViewport(0, 0, winWidth, winHeight);
```