

Computação Gráfica

Guilherme Henrique de Souza Nakahata

Universidade Estadual do Paraná - Unespar

20 de Abril de 2023

- Podemos modificar e alterar o tamanho das janelas;
- Em execução;
- Conteúdo da janela é redesenhado;
- Levando em conta as novas dimensões;
- Esperado é que a escala do desenho seja alterada de maneira que ele fique dentro da janela;
- Independente do tamanho da janela de visualização ou do desenho;
- Uma janela pequena terá o desenho completo e pequena;
- Uma janela grande terá o desenho completo e maior.

- Quando alteramos uma janela também devemos alterar o mapeamento de coordenadas;
- Aparência do desenho não pode ser alterada (Alteração visual);
- Detecção de alteração no tamanho da janela;
- Função **glutReshapeFunc**;
- Dois parâmetros - Altura e Largura;
- **Viewport** e **gluOrtho2D**.

- gluOrtho**2D**;
- gluOrtho**3D**;
- Define projeção ortográfica (2D);
- Sistema cartesiano;
- 4 Parâmetros;
- **gluOrtho2D**(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top);
- **Left** e **Right** limites mínimo e máximo no eixo X;
- **Bottom** e **Top** limites mínimo e máximo no eixo Y.

- ViewPort;
 - A viewport é uma região retangular na janela de exibição de uma aplicação gráfica;
 - Define a área onde a renderização OpenGL é exibida;
 - Uma janela dentro de outra janela;
 - É uma parte importante do pipeline de renderização do OpenGL;
 - Controlar a área de exibição da cena.
- **glViewport(0, 0, w, h);**
- Parâmetro a nova largura e altura da janela;
- 4 Parâmetros;
- Canto inferior esquerdo (0,0);
- Canto superior direito (w,h);
- É possível usar a viewport para visualizar mais de uma cena em diferentes áreas da janela.

- **glMatrixMode(GL_PROJECTION);**
- Inicializar o sistema de coordenadas;
- Antes da execução de operações de manipulação de matrizes;
- GluOrtho2D;
- Afetar o observador;
- **glLoadIdentity();**
- Carrega uma matriz identidade;
- Iniciando uma nova matriz;
- Substituindo a matriz antiga pela nova;
- Definir uma nova matriz de projeção.

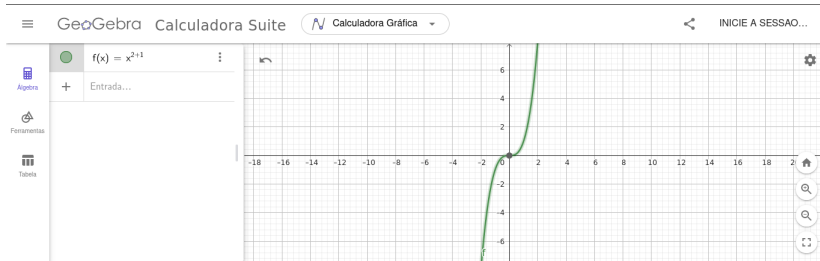
Aspect Ratio

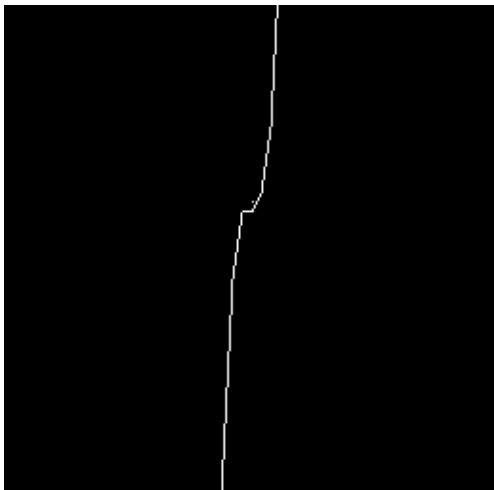
```
1 void resize(GLsizei w, GLsizei h)
2 {
3     if(h == 0) h = 1;
4
5     glViewport(0, 0, w, h);
6
7     glMatrixMode(GL_PROJECTION);
8     glLoadIdentity();
9
10    if (w <= h)
11        gluOrtho2D (0.0f, 180.0f, 0.0f, 180.0f*h/w);
12    else
13        gluOrtho2D (0.0f, 180.0f*w/h, 0.0f, 180.0f);
14 }
```

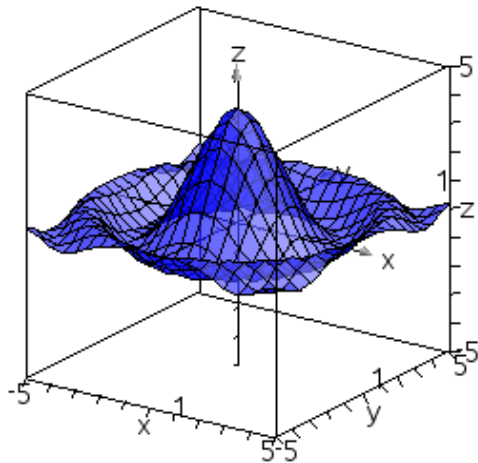
Exemplo

Funções Matemáticas

- Função Bidimensional;
- Função Tridimensional;
- Domínio da função;
- É possível gerar uma formação gráfica;
- Comportamento de curvas;
- Exemplo de Software Geogebra.







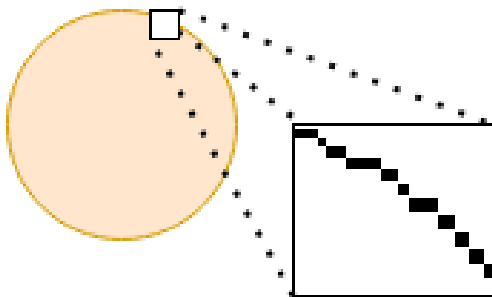
- Como fazer arcos, círculos, esferas?



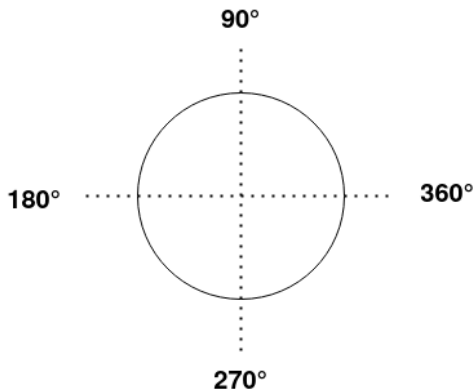
- Como fazer arcos, círculos, esferas?
- $r^2 = x^2 + y^2$;
- $f(x, y) = 0$, os pontos x e y pertencem à circunferência;
- $f(x, y) < 0$, os pontos x e y estão dentro da circunferência;
- $f(x, y) > 0$, os pontos x e y estão fora da circunferência;

Circunferência

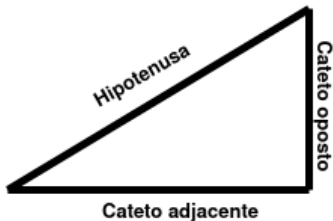
- Como fazer arcos, círculos, esferas?
- Quanto maior a quantidade de linhas na borda;
- Mais suave será a curva;
- Quanto mais linhas;
- Menor o tamanho das linhas;



Circunferência



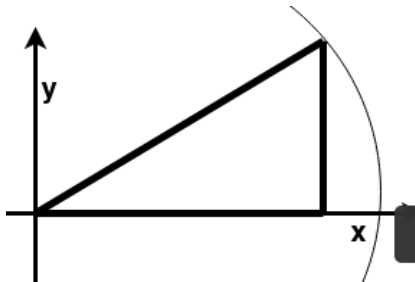
Razões Trigonométricas



$$\text{Cosseno} = \frac{\text{Cateto oposto}}{\text{Hipotenusa}}$$

$$\text{Seno} = \frac{\text{Cateto adjacente}}{\text{Hipotenusa}}$$

$$\text{Tangente} = \frac{\text{Cateto oposto}}{\text{Cateto adjacente}}$$



- E no OpenGL?
- Como fazer?
 - A partir de um ponto (x,y);
 - Raio r;
 - Número de pontos;
- Percorra todos os números de pontos:
 - $X = (X + r \cos(\text{angulo}))$;
 - $Y = (y + r \text{ seno}(\text{angulo}))$;
 - Ângulo recebe o novo Ângulo;

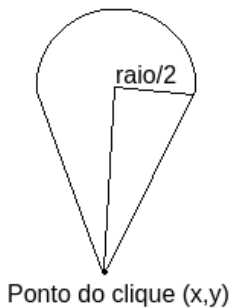
- E no OpenGL?
- Como fazer?
 - A partir de um ponto (x,y);
 - Raio r;
 - Número de pontos;
- Percorra todos os números de pontos:
 - $X = (X + r \cos(\text{angulo}))$;
 - $Y = (y + r \text{ seno}(\text{angulo}))$;
 - Ângulo recebe o novo Ângulo;

Se substituirmos o raio por distancia temos o Algoritmo de Bresenham;

Circunferência

```
1 void Desenha(void) {  
2     glClear(GL_COLOR_BUFFER_BIT);  
3  
4     glBegin(GL_LINE_STRIP);  
5  
6     for (double i = 0; i < 6; i++) {  
7         glVertex2i(OrigemCliquesX + raio * cos(i), OrigemCliquesY + raio * sin(i));  
8     }  
9  
10    glEnd();  
11  
12    glFlush();  
13 }
```

Exemplo



Obrigado! Dúvidas?

Guilherme Henrique de Souza Nakahata

guilhermenakahata@gmail.com

<https://github.com/GuilhermeNakahata/UNESPAR-2023>