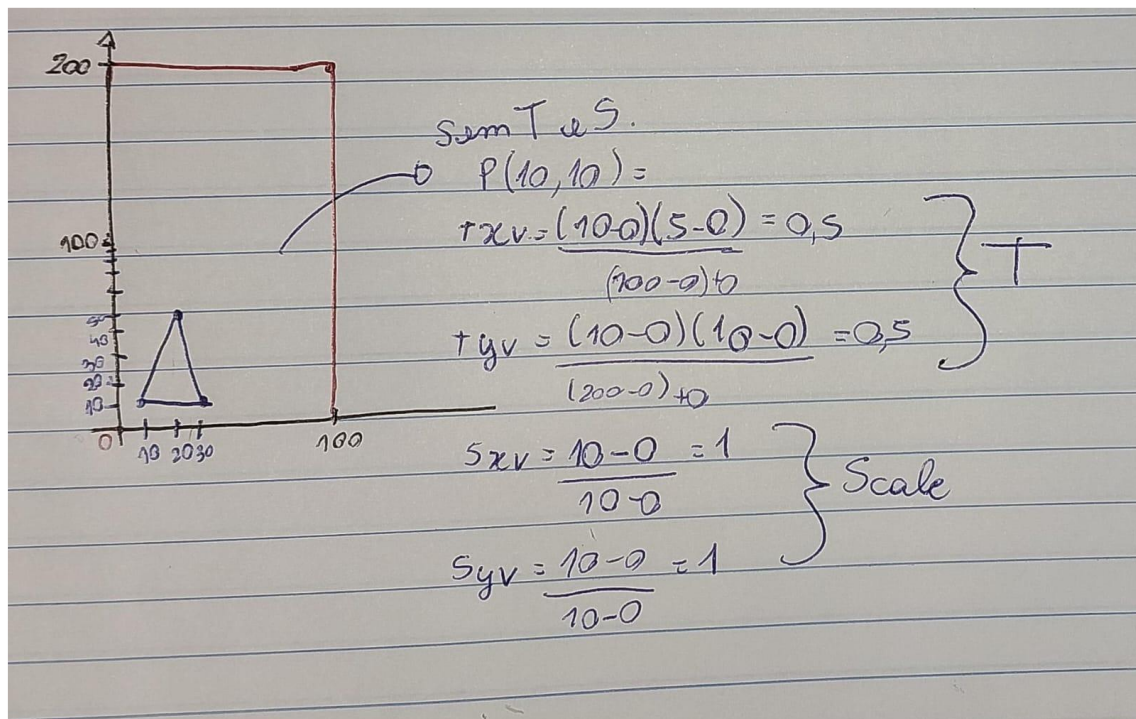


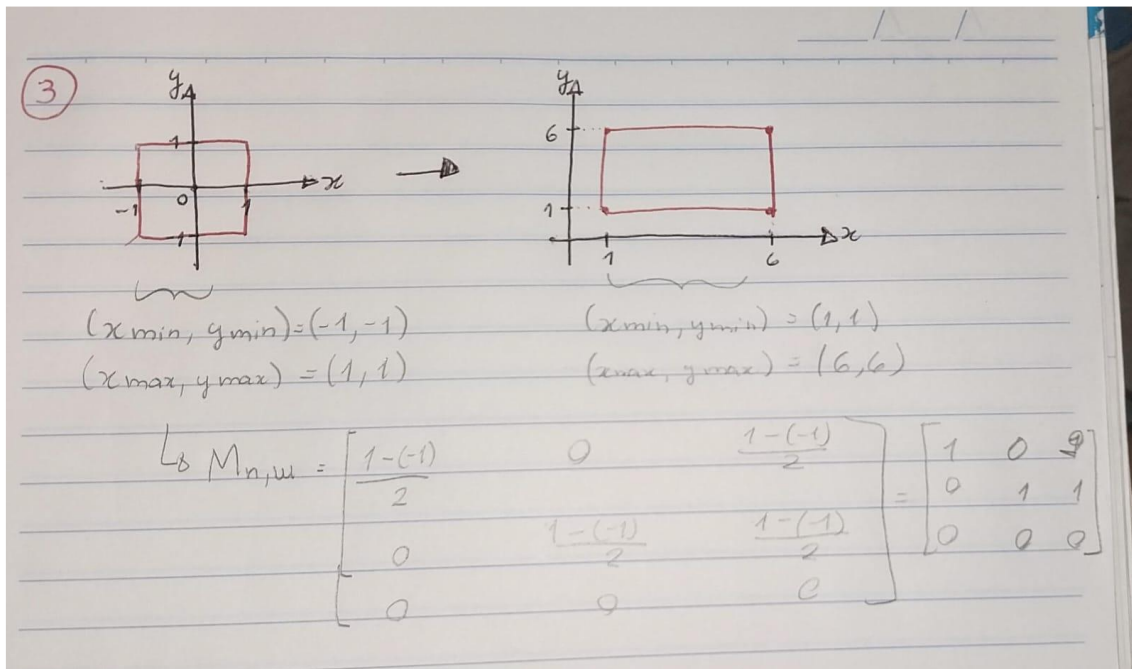
1 –

Janela de recorte é uma janela que mostra a parte do mundo que vai ser vista, e o que está fora não é visto, estando no sistema de coordenadas do mundo. Já a viewport está posicionada dentro da janela do sistema, e na viewport é possível ver o que está na janela de recorte, podendo termos várias viewport com diferentes posições, e tudo isso mapeado no sistema de coordenadas do dispositivo. Basicamente a janela recorta e a viewport mostra esse recorte mapeado.

2 –



3 –



4 –

Para um efeito de pan, precisamos fazer as translações com a viewport, assim movendo horizontalmente a “câmera”. Já para o efeito de zoom in e zoom out precisamos fazer uma mudança de escala e translação na matriz.

5 –

```
// Função para calcular o código de região de um ponto
int calculateRegionCode(int x, int y, int xmin, int ymin, int xmax, int ymax) {
    int code = INSIDE;
    if (x < xmin) code |= LEFT;
    if (x > xmax) code |= RIGHT;
    if (y < ymin) code |= BOTTOM;
    if (y > ymax) code |= TOP;
    return code;
}

// Função para recortar uma linha usando o algoritmo Cohen-Sutherland
void cohenSutherland(int x1, int y1, int x2, int y2, int xmin, int ymin, int xmax, int ymax) {
    int code1 = calculateRegionCode(x1, y1, xmin, ymin, xmax, ymax);
    int code2 = calculateRegionCode(x2, y2, xmin, ymin, xmax, ymax);
    int accept = 0;

    while (1) {
```

```

if (!(code1 | code2)) {
// Ambos os pontos estão dentro da janela
accept = 1;
break;
} else if (code1 & code2) {
// Ambos os pontos estão fora da janela em uma mesma região
break;
} else {
// Pelo menos um ponto está fora da janela, recortar a linha
int x, y;
int codeOut = code1 ? code1 : code2;
// Encontrar a interseção com a borda da janela
if (codeOut & TOP) {
x = x1 + (x2 - x1) * (ymax - y1) / (y2 - y1);
y = ymax;
} else if (codeOut & BOTTOM) {
x = x1 + (x2 - x1) * (ymin - y1) / (y2 - y1);
y = ymin;
} else if (codeOut & RIGHT) {
y = y1 + (y2 - y1) * (xmax - x1) / (x2 - x1);
x = xmax;
} else if (codeOut & LEFT) {
y = y1 + (y2 - y1) * (xmin - x1) / (x2 - x1);
x = xmin;
}
if (codeOut == code1) {
x1 = x;
y1 = y;
code1 = calculateRegionCode(x1, y1, xmin, ymin, xmax, ymax);
} else {
x2 = x;
y2 = y;
code2 = calculateRegionCode(x2, y2, xmin, ymin, xmax, ymax);
}
if (accept) {
printf("Linha recortada: (%d,%d) -> (%d,%d)\n", x1, y1, x2, y2);
} else {
printf("Linha completamente fora da janela.\n");
}
}

```

6 –

Precisamos fazer testes nas linhas para ver se a linha está totalmente fora da janela, está totalmente dentro, ou se tem algum tipo de interseção com a janela (parcialmente dentro). O algoritmo de Cohen-Sutherland faz isso comparando os valores dos bits com as fronteiras de recorte, podendo usar operações binárias. Após fazer os testes que dirão a resposta para a linha, vamos verificando as que não passaram pelo teste de linhas e encontrando suas

interseções a partir da ordem de bits, e cortando nas fronteiras, até termos apenas linhas dentro da janela de recorte.