

PŘÍLOHA P1: BRESENHAMŮV ALGORITMUS RASTERIZACE ÚSEČKY

```
/*-----  
Bresenhamuv celociselny linearni interpolacni algoritmus  
-----*/  
double CLine::Bresenham(int x0, int y0, int x1, int y1)  
{  
    Barva(red,green,blue);  
    Casovac->Unpause();  
  
    int dy = y1 - y0;  
    int dx = x1 - x0;  
  
    int krokx, kroky;  
    if (dy < 0) { dy = -dy; kroky = -1; } else kroky = 1;  
    if (dx < 0) { dx = -dx; krokx = -1; } else krokx = 1;  
  
    dy >>= 1; // dy je 2*dy  
    dx >>= 1; // dx je 2*dx  
  
    Bod(x0,y0);  
    if (dx > dy)  
    {  
        int predikace = dy - (dx >> 1); // dy - 2*dx  
        while (x0 != x1)  
        {  
            if (predikace >= 0)  
            {  
                y0 += kroky;  
                predikace -= dx;  
            }  
            x0 += krokx;  
            predikace += dy;  
            Bod(x0,y0);  
        }  
    } else  
    {  
        int predikace = dx - (dy >> 1);  
        while (y0 != y1)  
        {  
            if (predikace >= 0)  
            {  
                x0 += krokx;  

```

```
        predikace -= dy;
    }
    y0 += kroky;
    predikace += dx;
    Bod(x0,y0);
}

Casovac->Pause();
return Casovac->GetTime();
}
```

PŘÍLOHA P2: VÝPOČET BEZIÉROVY KUBICKÉ KŘIVKY

```
/*-----  
Vypocet bodu Bezierovy kubiky  
-----*/  
  
bod CCurve::Souradnice_Bezierova_Kubika_Iterativne(bod p1,bod p2,bod p3,bod p4,  
float t)  
{  
    float pom1, pom2, pom3;  
    float B0,B1,B2,B3;  
    bod point;  
    // pomocne promene pro zprehledneni kodu  
    pom1 = 1 - t;  
    pom2 = pom1 * pom1 * pom1;  
    pom3 = t * t * t;  
  
    // Bernsteinovy polynomy  
    B0 = pom2;  
    B1 = 3*t*pom1*pom1;  
    B2 = 3*t*t*pom1;  
    B3 = pom3;  
  
    // vypocet bodu krivky pro t = <0,1>  
    point.x = B0*p1.x + B1*p2.x + B2*p3.x + B3*p4.x;  
    point.y = B0*p1.y + B1*p2.y + B2*p3.y + B3*p4.y;  
    point.z = B0*p1.z + B1*p2.z + B2*p3.z + B3*p4.z;  
  
    // vrat novou souradnici  
    return point;  
}  
  
/*-----  
Bezierova kubika urcena iterativnim postupem  
-----*/  
  
double CCurve::Bezierova_Kubika_Iterativne(bod A, bod B, bod C, bod D)  
{  
    Barva(red,green,blue);  
    Casovac->Unpause();  
  
    // pocatek krivky  
    bod point0 = A;  
    bod point1;
```

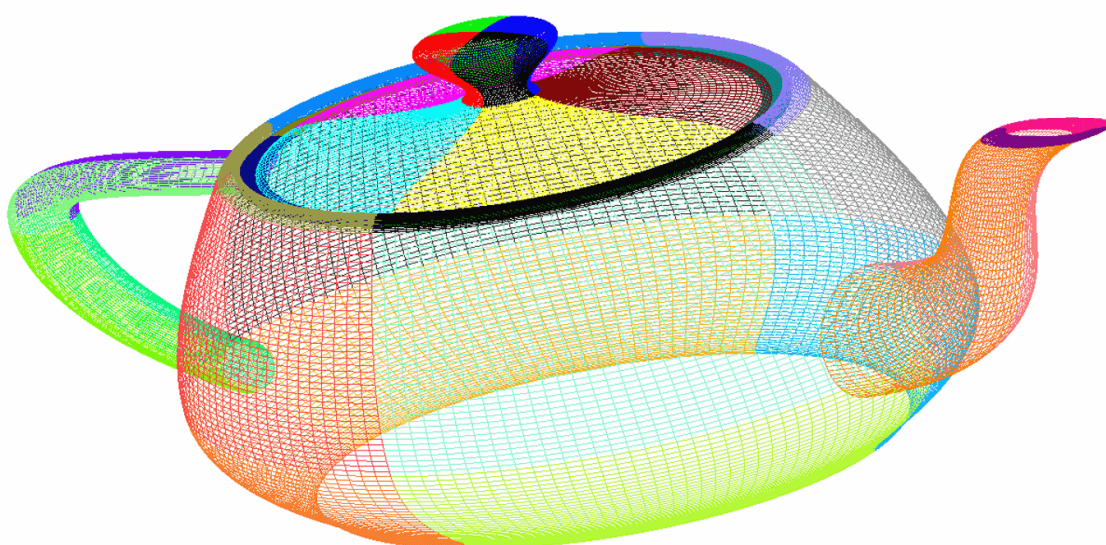
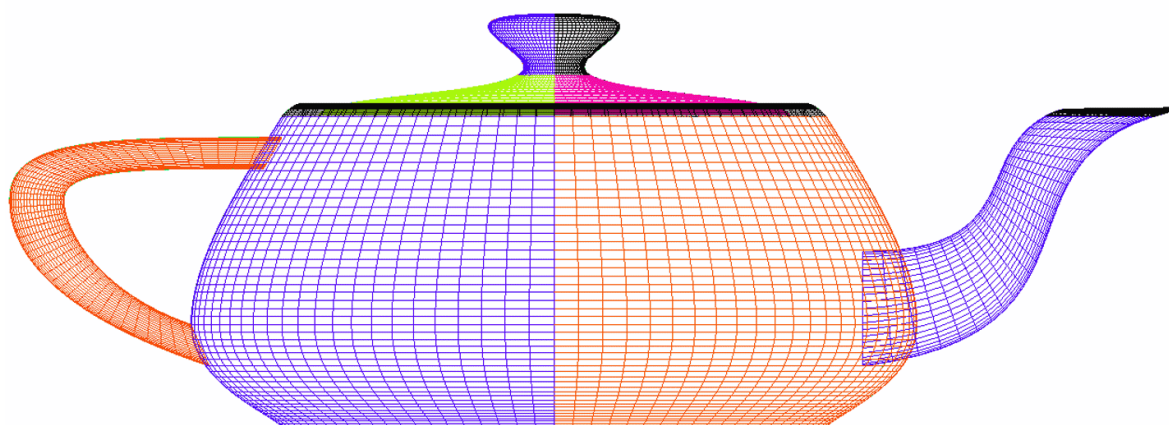
```

1))    for(float t = 0; t <= (1 + (1.0f / MAX_KROKU)); t += 1.0f / (MAX_KROKU -
      {
          point1 = Souradnice_Bezierova_Kubika_Iterativne(A,B,C,D,t);
          Cara->Bresenham(point0.x, point0.y,point1.x, point1.y);
          point0 = point1;
      }

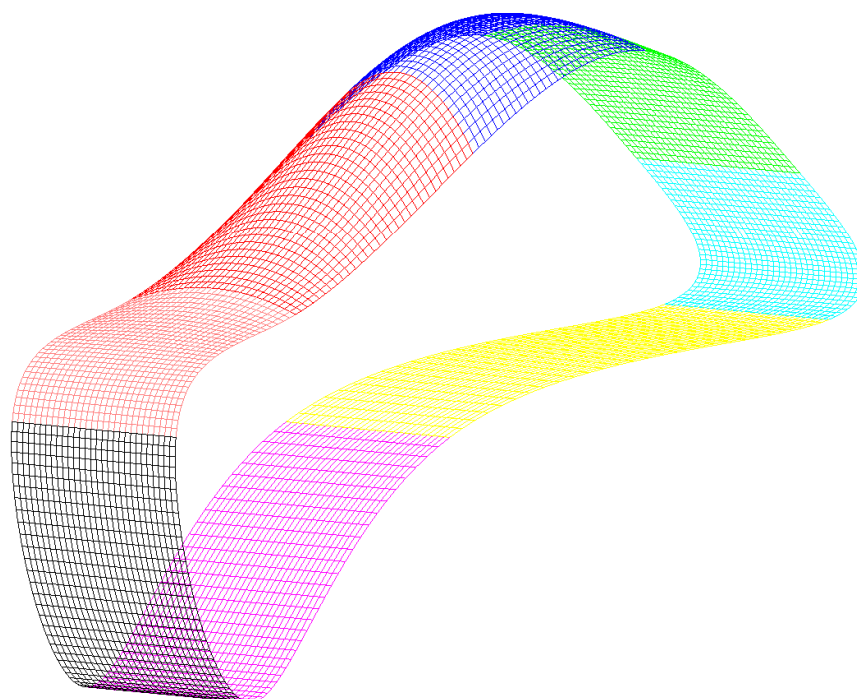
      Casovac->Pause();
      return Casovac->GetTime();
}

```

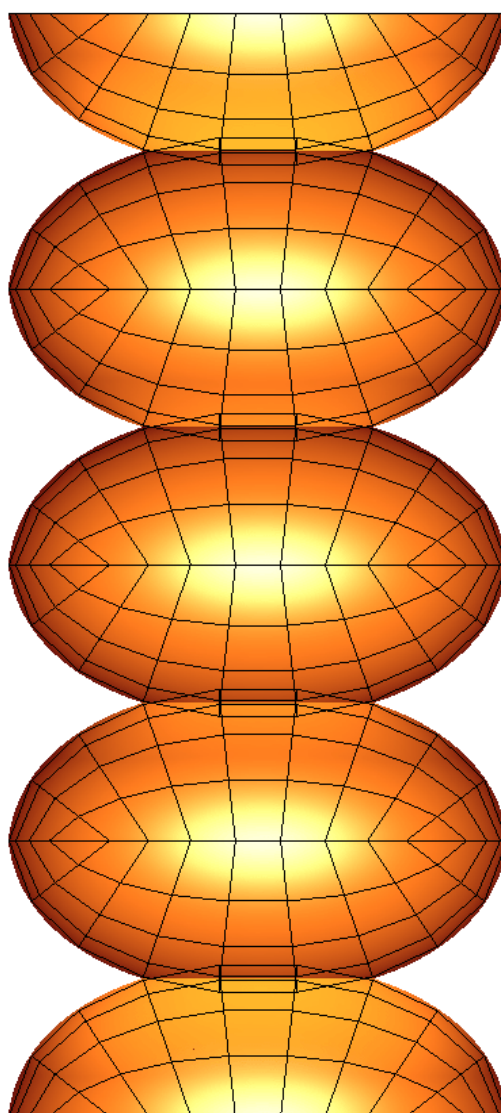
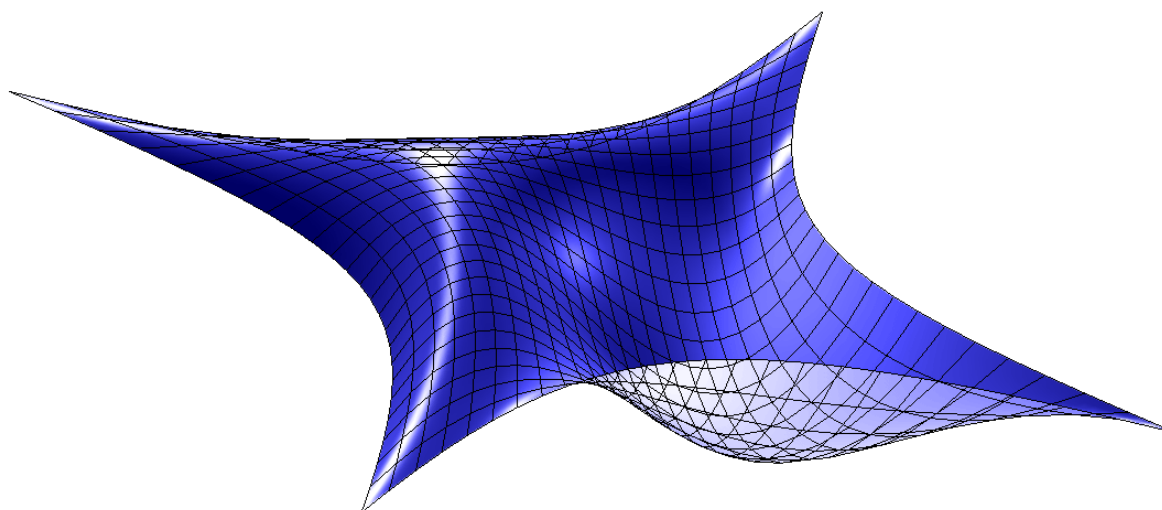
PŘÍLOHA P3: ČAJNÍK TVOŘENÝ 32 BEZIÉROVÝMI PLÁTY



PŘÍLOHA P4: PLÁTOVANÁ COONSOVA PLOCHA



PŘÍLOHA P5: BEZIÉROVY BIKUBIKY



PŘÍLOHA P6: NÁVRH KAROSERIE AUTA POMOCÍ COONSOVY KŘIVKY

