CG基礎 第十回レポート

201811395 山本雄太

必須課題

実装したupdateClothのコードを以下に示す

|  |
| --- |
| // 布の形状の更新  void updateCloth(void) {  // ★次の手順で質点の位置を決定する  // 1. 質点に働く力を求める  // 2. 質点の加速度を求める  // 3. 質点の速度を更新する  // 4. 質点の位置を更新する  //1-1. 質点に働く力をリセット  // 全ての質点について順番に処理する  for (int y = 0; y < POINT\_NUM; y++) {  for (int x = 0; x < POINT\_NUM; x++) {  cloth->points[x][y].f.set(0, 0, 0);  }  }  //1-2. バネの両端の質点に力を働かせる  // （考え方）質点を1つ1つ調べるのではなく  // バネを1つ1つを見て、その両端の質点に力を加算していくアプローチをとる  for (int i = 0; i < cloth->springs.size(); i++) {// 全てのバネについて順番に処理する  Spring\* spring = cloth->springs[i];  // (a) バネの自然長と現在の長さの差分を求める  // バネの自然長は spring->restLength に格納されている  // 現在の長さは、両端の質点間の距離を計算して求める  double nowLength = (spring->p0->p - spring->p1->p).length();  double difLength = spring->restLength - nowLength;  // (b) バネが質点に加える力の大きさを求める  // （自然長 - 現在の長さ）にバネ定数 Ks を掛けた値が求める大きさ  double fm = Ks \* difLength;  // (c) バネが質点に加える力の向き（単位ベクトル）を求める  // バネには、両端に質点がついているので、一方から他方に向かう方向が力の向き  Vector3d fmDir = spring->p0->p - spring->p1->p;  fmDir.normalize();  // (d) 両端の質点に対して、力ベクトル（ 大きさは(b)で求めた。向きは(c)で求めた）を加算する  // spring->p0->f バネの一方の質点に加わる力を表すので、これに力ベクトルを加える（向きに注意）  // spring->p1->f バネのもう一方の質点に加わる力を表すので、これに力ベクトルを加える（向きに注意）  spring->p0->f = spring->p0->f + (fmDir \* fm);  spring->p1->f = spring->p1->f - (fmDir \* fm);  }  //1-3. 重力、空気抵抗による力を加算する  // 全ての質点について順番に処理する  for (int y = 0; y < POINT\_NUM; y++) {  for (int x = 0; x < POINT\_NUM; x++) {  // cloth->points[x][y].f に、重力による力を加算する  Vector3d Mg = Mass \* gravity;  cloth->points[x][y].f = cloth->points[x][y].f + Mg;  // cloth->points[x][y].f に、空気抵抗による力を加算する  // 空気抵抗による力は速度に定数Dkをかけたもの。ただし向きは速度と逆向き  // 速度は cloth->points[x][y].v で表される  cloth->points[x][y].f = cloth->points[x][y].f - (cloth->points[x][y].v \* Dk);  }  }  // ここまでで、質点に加わる力をすべて計算し終わった  // 全ての質点について順番に処理する  for (int y = 0; y < POINT\_NUM; y++) {  for (int x = 0; x < POINT\_NUM; x++) {  // 頂点が固定されている場合は何もしない  if (cloth->points[x][y].bFixed) continue;  // 2. 質点の加速度（ベクトル）を計算 (力ベクトル cloth->points[x][y].f を質量で割った値)  Vector3d accelt = cloth->points[x][y].f / Mass;  // 3. 質点の速度 (cloth->points[x][y].v) を加速度に基づいて更新する  cloth->points[x][y].v = cloth->points[x][y].v + (accelt \* dT);  // 4. 質点の位置 (cloth->points[x][y].p) を速度に基づいて更新する  cloth->points[x][y].p = cloth->points[x][y].p + (cloth->points[x][y].v \* dT);  // オプション. 球体の内部に入るようなら、強制的に外に移動させる  }  }  } |

実行結果は動画を添付した。