

Rapport de séance:

Séance 3 – 10 /01/2010

Pendant cette troisième séance, j'ai amélioré le programme de simulation d'interactions entre les billes en revoyant le transfert de vitesse entre celles-ci lors d'une accélération du cube. Je me suis aussi renseigné sur le principe de l'IMU, un système de mesure de l'inertie dont nous aurons besoin pour correctement récupérer les données de position et rotation du Cube. J'ai trouvé un algorithme (algorithme de Madgwick) qui répond à nos besoins et que nous utiliserons pour traiter les données d'un accéléromètre, un gyroscope, et un magnétomètre.

Partie du code revue:

```
float[] correction = new float[]{x1-x2,y1-y2};
double d = distance(correction,new float[]{0F,0F});
if(d==0){
    d = this.size -1;
    correction = new float[]{this.size/2,this.size/2};
}
correction = mult(correction,(this.size-d)/d);

// les billes ont toutes la meme masse donc (m1/(m1+m2)) = 0.5
correction =mult(correction,0.5F);

for(int i=0; i<2; i++){
    float change = this.coords[i] + correction[i];
    float high = change + this.size/2;
    float low = change - this.size/2;
    float arrayHigh = ledArray.coords[i] + ledArray.size;
    float arrayLow = ledArray.coords[i];
    if(high > arrayHigh || low < arrayLow){
        isWallColliding = true;
        marble.coords[i] -= correction[i]*2F;
    }
    else {
        change = marble.coords[i] - correction[i];
        high = change + marble.size/2;
        low = change - marble.size/2;
        if(high > arrayHigh || low < arrayLow){
            marbleWallColliding = true;
            this.coords[i] += correction[i]*2F;
        }
        else {
            this.coords[i] += correction[i];
            marble.coords[i] -= correction[i];
        }
    }
}
```