(*)

```
satellites[i].a += OrbitCalculator::G
                  * satellites[a].m
((satellites[b].pos - satellites[i].pos)
^2) : }
```

```
satellites[i].v += satellites[i].a * delta
```

wichtig!!!

```
{ satellites[i].posn = satellites[i].a
                       * delta * delta / 2
                       + satellites[i].v
                       * delta
                       + satellites[i].pos
satellites[i].a = [0,0,0]
```

```
}
}
```

```
satellites[] satellites; // header
```

```
OrbitCalculator::_ready() {
```

```
OrbitCalculator::_process(){
for(int i=0; i++; i<satellites.length()
{ for(int b=0; b++; b<satellites.length()
{ if (i==b) continue;
```

```
OrbitCalculator::new_body(satAttr attr){
OrbitCalculator.append(satellites, satAttr);
```

Orbitalcalculation

satatt[] satellites

```
struct satatt {
  int[] pos = [0,0,0];
  int[] v   = [0,0,0];
  int[] acc = [0,0,0];
  int[] collisionArea = [-1,-1,-1
                         ,1,1,1];

  int m
  int[] posn = [0,0,0]
}
```