

```

/* SSE Collision detection routines */

#include <stdio.h>

int main()
{
float x[3];

float y[3];

float distance;

float midpoint[3];

float two=2;

#define distance_sse(arg1, arg2)\
__asm__ __volatile__(\
    "movss %1, %%xmm0\n"\
    "movss %2, %%xmm1\n"\
    "movss %3, %%xmm2\n"\
    "movss %4, %%xmm3\n"\
    "movss %5, %%xmm4\n"\
    "movss %6, %%xmm5\n"\
    "subps %%xmm3, %%xmm0\n"\
    "subps %%xmm4, %%xmm1\n"\
    "subps %%xmm5, %%xmm2\n"\
    "mulps %%xmm0, %%xmm0\n"\
    "mulps %%xmm1, %%xmm1\n"\
    "mulps %%xmm2, %%xmm2\n"

```

```
"addps %%xmm1, %%xmm0\n\  
"addps %%xmm2, %%xmm0\n\  
    "sqrtss %%xmm0, %%xmm1\n\  
    "movss %%xmm1, %0\n\  
:"=m"(distance) \  
:"m"(arg1[0]),"m"(arg1[1]),"m"(arg1[2]),"m"(arg2[0]),"m"(arg2[1]),"m"(arg2[2])\  
    );
```

```
#define midpoint_sse(arg1,arg2)\  
__asm__ __volatile__ (\  
    "movss %3, %%xmm0\n\  
    "movss %4, %%xmm1\n\  
"movss %5, %%xmm2\n\  
"movss %6, %%xmm3\n\  
"movss %7, %%xmm4\n\  
"movss %8, %%xmm5\n\  
    "movss %9, %%xmm7\n\  
"addps %%xmm0, %%xmm3\n\  
"addps %%xmm1, %%xmm4\n\  
"addps %%xmm2, %%xmm5\n\  
    "divss %%xmm7, %%xmm3\n\  
    "divss %%xmm7, %%xmm4\n\  
    "divss %%xmm7, %%xmm5\n\  
    "movss %%xmm3, %0\n\  
    "movss %%xmm4, %1\n
```

```
"movss %%xmm5, %2\n"\  
:"=m"(midpoint[0]), "=m"(midpoint[1]), "=m"(midpoint[2])\  
:"m"(arg1[0]), "m"(arg1[1]), "m"(arg1[2]), "m"(arg2[0]), "m"(arg2[1]), "m"(arg2[2]), "m"(two) \  
);
```

```
x[0]=11;
```

```
x[1]=2;
```

```
x[2]=3;
```

```
y[0]=-1;
```

```
y[1]=-2;
```

```
y[2]=-33;
```

```
/* 3d diagonals length */
```

```
distance_sse(x,y);
```

```
printf("Distance: %f\n", distance);
```

```
midpoint_sse(x,y);
```

```
printf("Midpoint: %f,%f,%f\n",midpoint[0],midpoint[1],midpoint[2]);
```

```
/* radii code here */
```

```
distance_sse(x,midpoint);
```

```
printf("Radii: %f\n", distance);
```

}