Geometry Exercise

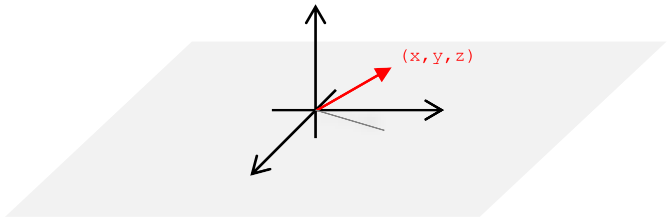
Similar to:
Old Exam Question Feb. 2010, Ex. 5

Exercise

In this exercise we will implement a representation of 3D-geometrical objects in a computer game.

Given is a struct `vec` which stores 3D-vectors.

```
struct vec {
  double x, y, z;
};
```



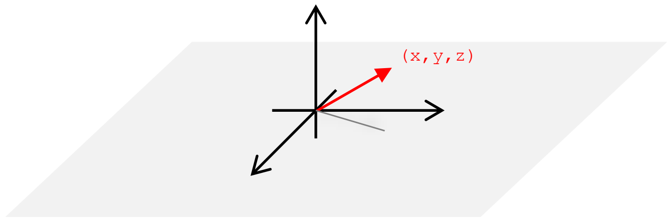
Exercise a)

Exercise a)

Implement the following function which computes a new vector obtained by adding two vectors.

```
// POST: returns the sum of two vectors
vec sum(const vec& a, const vec& b);
```

```
struct vec {
  double x, y, z;
};
```



Exercise a)

Solution a)

```
// POST: returns the sum of two vectors
vec sum(const vec& a, const vec& b) {
  vec tmp;
  tmp.x = a.x + b.x;
  tmp.y = a.y + b.y;
  tmp.z = a.z + b.z;
  return tmp;
}
```

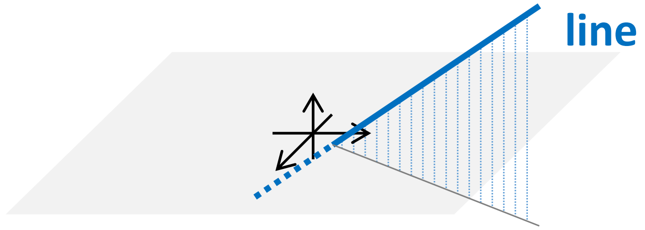
Exercise b)

Exercise b)

Propose a struct named `line`, which can be used to represent 3D-straight-lines.

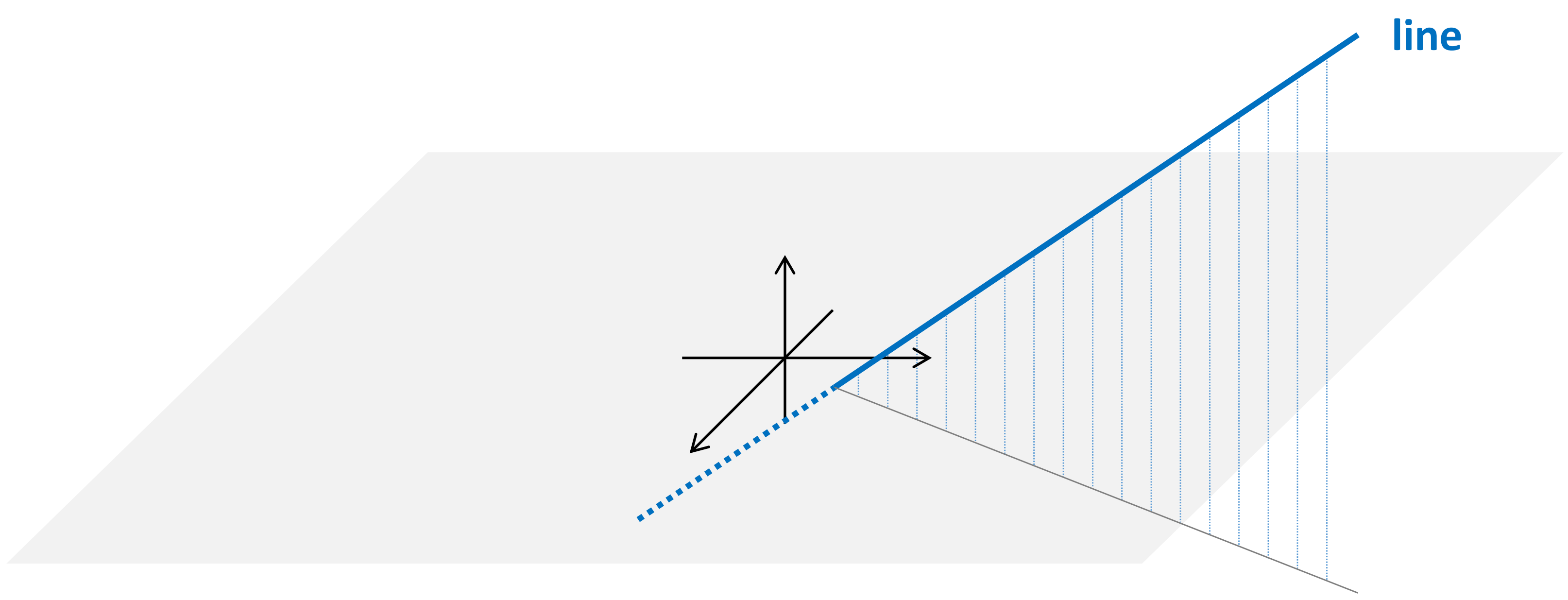
A particular straight line does not have to be representable uniquely, but conversely every object of type `line` has to represent a unique straight line. If necessary you can for this reason define a suitable invariant (`// INV:...`) which has to be met when using the `line` struct.

```
struct vec {
  double x, y, z;
};
```



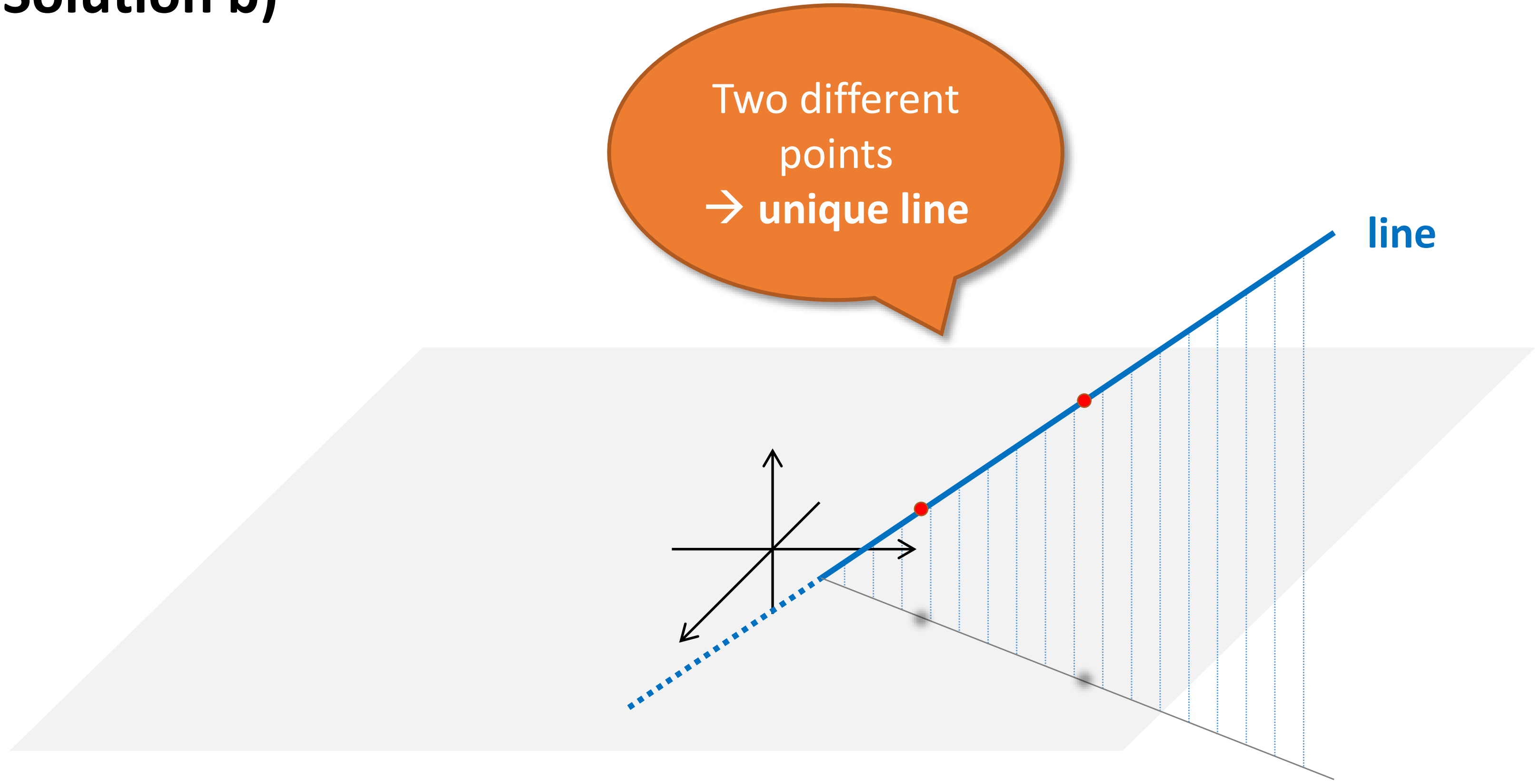
Exercise b)

Solution b)



Exercise b)

Solution b)



Exercise b)

Solution b)

```
struct line {
  vec a, b; // INV: a != b
};
```

Exercise c)

Exercise c)

Based on your struct `line` implement the following function which returns a new shifted `line`.

```
// POST: returns a new line obtained by shifting l
//       by v.
line shift_line (const line& l, const vec& v);
```

```
struct vec {
  double x, y, z;
};
```

```
struct line {
  vec a, b; // INV: a != b
};
```

Exercise c)

Solution c)

```
// POST: returns a new line obtained by shifting l
//       by v.
line shift_line (const line& l, const vec& v) {
  line tmp;
  tmp.a = sum(l.a, v);
  tmp.b = sum(l.b, v);
  return tmp;
}
```