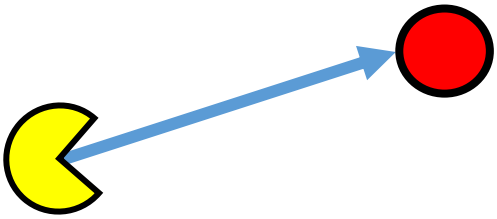
ゲームＰＧ　評価テスト

① プレイヤーの座標を PV、敵の座標を EV としたとき、プレイヤーから敵に向かう下記のようなベクトルを求める計算式を記述しなさい。

解答例)

PV + EV



EV－PV

② 敵からプレイヤーに向かうベクトルを EP としたとき、この２点間の距離を計算する時に一般的に使用される公式を下記から選択しなさい。

ア： 三平方の定理

イ： 平面の方程式

ウ： 余弦定理

エ： 外積

③ ゲームにはゲーム起動中終わることなくループし続けて、ゲームの状態の更新や描画を行うループが存在する。このループが一般的に何と呼ばれるか下記の４つから選びなさい。

ア：コールバック関数

イ：コルーチン

ウ: ゲームループ

エ：レンダーコンテキスト

④ 上記のループは固定 fps(frame per second)のゲームの場合、ある一定周期で回り続けている。例えば 60fps で動作しているゲームの場合、このループは何秒間隔で回っているか下記から選びなさい。

ア：$\frac{1}{30}$秒

イ：$\frac{1}{50}$秒

ウ: $\frac{1}{60}$ 秒

エ：60 秒

評価テスト

① 当たり判定などで使用される、オブジェクトを内包する箱形状のことを一般的になんというか下記から選びなさい。

ア　バウンディングスフィア

イ　バウンディングボックス

ウ　カプセルコリジョン

② 上記の箱形状のうち、特に軸に平行なものを何というか下記から選びなさい。

ア　OBB

イ　AABB

ウ　BSP

エ　BVH

③ 古典的な２Ｄゲームでよく使用されていて、今でもゲームの種類によっては十分に実用的な当たり判定に配列を使用したものがある。
例えば下記のようなコードでプレイヤーの現在いるマップの情報が取得できるものとする

```
int map[3][3] = {
 { 1, 1, 1},
 {1, 0, 0 },
 {1, 0, 1 },
};
void Func()
{
    int mapAttr = map[player.pos_y][player.pos_x]; //現在のマップの情報を取得。
}
```

では、プレイヤーのいるマップが１の場合、**移動できません。**と表示するプログラムを記述しなさい。
解答例)

```
int mapAttr = map[player.pos_y][player.pos_x];
if(mapAttr > 1){
   std::cout << “移動できません。”;
}
```

```
int mapAttr = map[player.pos_y][player.pos_x];
if(mapAttr == 1){
   std::cout << “移動できません。”;
}
```

評価テスト

① 3D オブジェクトの任意の軸周りの回転を表現するものとして適切なものを下記から選びなさい。

ア：ポリゴン

イ: クォータニオン

ウ：カメラ

② ９０度をラジアン単位で表しなさい。円周率はπとする。

解答例 2×π

0.5π

③ ４５度をラジアン単位で表しなさい。円周率はπとする。

0.25π

④ 0.75π(ラジアン単位)を度に変換しなさい。

0.75×180 = 135 度