デザインパターン

2I06 糸川倫太郎

1.

> Ball.h，BallFactory.h，BallFactory.cpp のソースコード

```
cpp : ball.h
#pragma once
#include "Timer.h"
#include <SFML/Graphics.hpp>

class Ball {
protected:
  double x, y;
  double vx, vy;
  double ax, ay;

  Timer timer;
  Timer rap;

  sf::Texture* texture;
  sf::Sprite sprite;

public:
  enum class TYPE {
    BASKET,
    BASE,
    TENNIS
  };
  Ball();
  virtual ~Ball();

  virtual void init();
  virtual void update();
  virtual void draw();
  virtual void clean();
};
```

```
cpp : ballfactory.h
#pragma once
#include "Ball.h"

class BallFactory {
public:
  static Ball* create(Ball::TYPE type);
};
```

```
cpp : ballfactory.cpp
#include "BallFactory.h"
#include "BasketBall.h"
```

```
#include "BaseBall.h"
#include "TennisBall.h"

Ball* BallFactory::create(Ball::TYPE type) {
  switch(type){
    case Ball::TYPE::BASKET:
      return new BasketBall;
    case Ball::TYPE::BASE:
      return new BaseBall;
    case Ball::TYPE::TENNIS:
      return new TennisBall;
    default:
      return nullptr;
  }
}
```

2.

> BallApp.cpp の修正した部分

```cpp
...
#include "BallFactory.h"


...

void BallApp::init()
{
  balls.resize(100);

  for (auto& b : balls) {
    int r = Rand::range(0, 9); // 0～9の乱数
    if (r < 1)
      b = BallFactory::create(Ball::TYPE::BASKET);
    else if (r < 4)
      b = BallFactory::create(Ball::TYPE::BASE);
    else
      b = BallFactory::create(Ball::TYPE::TENNIS);
    b->init();
  }
}
...
```