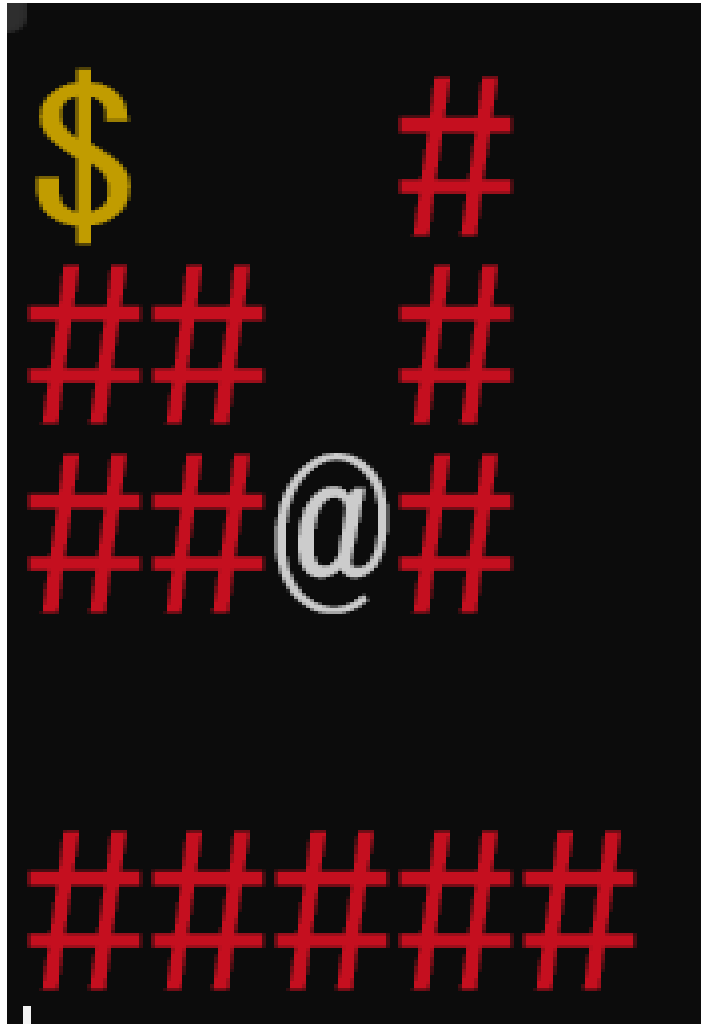
演習：2D迷路

- コマンドプロンプトの画面上で
  迷路を探索して、宝物（＄）を探す
  簡単なゲームを制作してみる

- ＃：壁
  @：自キャラ
  ＄：宝物

- WASDキーで上下左右移動



演習：2D迷路

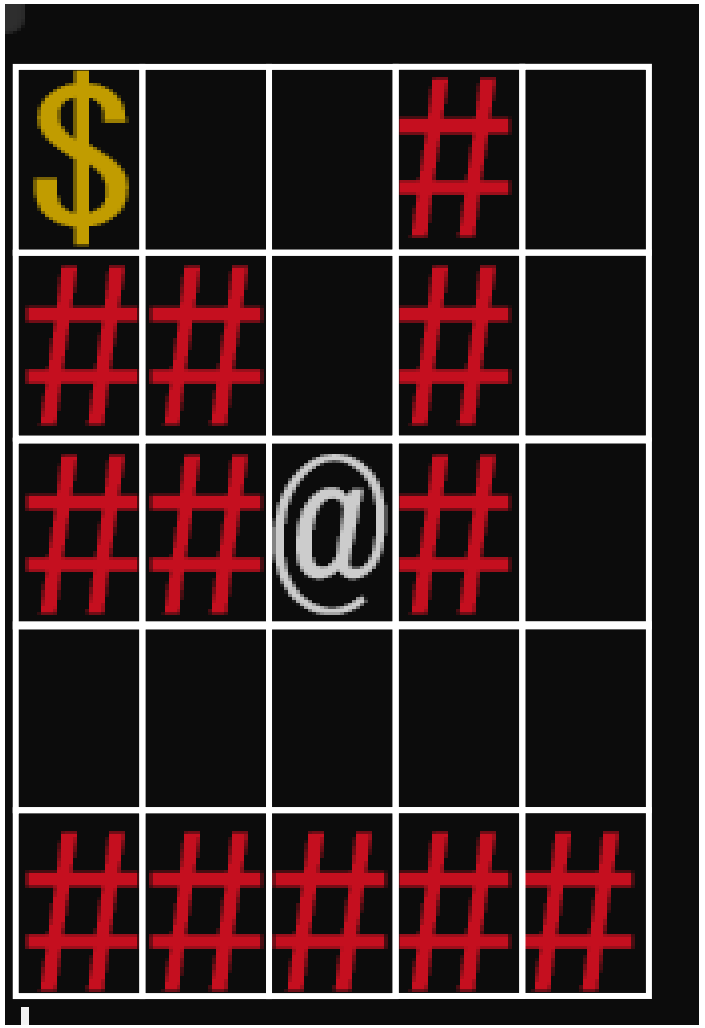
演習：2D迷路

- C++作業フォルダ内に**PracMaze**フォルダを作成
```
mkdir PracMaze
cd PracMaze
```

- **PracMaze**フォルダ内に main.cpp を作成する
```
copy nul main.cpp
```

演習：2D迷路

- 迷路概要
  - ① 迷路のデータはCSVに格納
  - ② 自キャラは通路のみ通行可
  - ③ 自キャラの周囲5×5マス
しか表示しない



演習：2D迷路

- 迷路ファイル2d_maze.csvの作成
  ① Excelで1を壁、0を通路として迷路データを作成する
  （※条件付き書式を使えば壁を判別しやすくなる）

  ② 制約条件
  ・縦10～50マス×横10～50マスの迷路データを作成する
  ・最外周の壁のみ2マスにする

  ③ PracMazeフォルダに2d_maze.csvのファイル名で
  「CSV（コンマ区切り）」形式で保存（一旦Excelブック形式で保存したほうが後から変更するときに楽？）

演習：2D迷路

- 開発方針
  - ① 迷路のマップデータは配列に格納する
  - ② マップデータの行数と列数は変更できるように動的配列を用いる
  - ③ 機能別にクラス化を行い、main関数はクラスのメンバを呼び出す機能のみを持つ
  - ④ とりあえずマップデータの読み込みと、その表示を行う機能から実装する

演習：2D迷路

- データ等の管理方法
  - マップデータの行数と列数が不定のため、vectorの二次元配列で管理する
  - 二次元配列は**MapData**クラスで管理
    マップ配列へのデータの格納や取り出しはこのクラスで行う
  - 壁や通路、宝物の種類は列挙型（enum）で管理
  - マップの描画やCSVファイルの読み込みや配列への格納は**Map**クラスで管理

演習：2D迷路

- **PracMaze**フォルダ内に maze.h, maze.cpp, map.h, map.cpp を作成

```
copy nul maze.h
copy nul maze.cpp
copy nul map.h
copy nul map.cpp
```

演習：2D迷路

maze.h

```
#pragma once
#include <vector>
using namespace std;

enum MapObj {
  ROAD,              //通路    0
  WALL,              //壁      1
  TREASURE           //宝物    2
};

class MapData {
private:
  vector<vector<int>> mapData{};//マップ格納用二次元配列
public:
  MapData();                    //コンストラクタ
  int getMapValue(int x, int y);//指定座標(x,y)のデータを返す
  int getMapSize(int i);  //列数を取得
  int getMapSize();       //行数を取得
  void addMap(int j, int value);//指定行(j)にvalueを追加
  void resizeMap();       //二次元配列に新しい行を追加
  void setMapValue(int x, int y, int value);
                          //指定座標(x,y)にvalueを格納
};
```

演習:2D迷路

maze.cpp

```
#include “maze.h“
MapData::MapData() = default;//デフォルトコンストラクタ
int MapData::getMapValue(int x, int y) {
  return mapData[y][x]; }     //座標(x,y)の要素の値valueを返す
int MapData::getMapSize(int i) {
  return mapData[i].size(); }//列数を返す
int MapData::getMapSize() {
  return mapData.size(); }    //行数を返す
void MapData::addMap(int j, int value) {
  mapData[j].push_back(value); } //行を指定してvalueを追加
void MapData::resizeMap() {
  mapData.resize(mapData.size() + 1); } //新しい行を追加
void MapData::setMapValue(int x, int y, int value) {
  mapData[y][x] = value; }          //座標(x,y)へvalueを格納
```

演習：2D迷路

map.h

```
#pragma once
#include “maze.h”
#include <iostream>

class Map {
public:
  MapData mapdata; //MapDataクラスのインスタンス（マップデータ）
  Map();                              //コンストラクタ
  void Load(string filename); //CSVファイルの読み込みと格納
  void DrawMap();                     //マップ描画
};
```

演習：2D迷路

map.cpp

```
#include "map.h“
#include "maze.h“
#include <fstream>
#include <sstream>
#include <random>
#include <algorithm>
using namespace std;

Map::Map() = default;                    //デフォルトコンストラクタ
void Map::Load(string filename) {//CSVファイルの読み込み
  ifstream ifs(filename);                //入力ファイルストリームの生成
  if (ifs.fail()) {                              //エラーが発生したとき
    cout << “FileOpen Error” << endl; //エラーメッセージ表示
    exit(-1);                                    //コード-1で強制終了
  }
```

演習:2D迷路

map.cpp

```
  string text;
  int j = 0;     //二次元配列の行を管理する変数
  while (getline(ifs, text)) {          //ファイルから一行読込み
    mapdata.resizeMap();                //二次元配列の行を追加
    istringstream iss(text);            //文字列ストリームを生成
    while (getline(iss, text, ',')) {//コンマで分割
      mapdata.addMap(j, stoi(text)); //マップデータに格納
    }
    j++;          //行数をカウントアップ
  }
  ifs.close();
}
```

演習:2D迷路

map.cpp

```
void Map::DrawMap() {
  for (int y = 0; y < mapdata.getMapSize(); y++) {
    for (int x = 0; x < mapdata.getMapSize(y); x++) {
      if (mapdata.getMapValue(x, y) == WALL) {
        cout << “#”; //壁として#を出力
      }
      else if (mapdata.getMapValue(x, y) == ROAD) {
        cout << “.”; //通路として.を出力
      }
    }
    cout << endl;
  }
}
```

演習:2D迷路

main.cpp

```
#include <iostream>
#include <conio.h>
#include <Windows.h>
#include “map.h“
using namespace std;
int main()
{
  Map map;                          //Mapクラスの生成
  map.Load(”2d_maze.csv”);//マップファイルの読み込みと格納
  system(“cls”);                    //コマンドプロンプトの画面消去
  //コマンドプロンプトの表示位置を(0,0)へ設定
  SetConsoleCursorPosition(
    GetStdHandle(STD_OUTPUT_HANDLE), COORD{ 0, 0 });
  map.DrawMap();                    //マップの描画
  return 0;
}
```

演習：2D迷路

- コマンドプロンプト上の表示場所を固定する

カーソル位置を指定する関数

- SetConsoleCursorPosition(

画面上のウィンドウからコマンドプロンプトを指定

GetStdHandle(STD_OUTPUT_HANDLE),

COORD{ 0, 0 });

↑座標(0,0)

演習：2D迷路

- 壁と通路の色を変更する
  - エスケープシーケンスという機能を利用することで、画面に表示される色を変更可能

```
黒: ¥033[30m
赤: ¥033[31m
緑: ¥033[32m
黄: ¥033[33m
青: ¥033[34m
白: ¥033[37m
無: ¥033[m　※これでリセットしないと色が残る
```

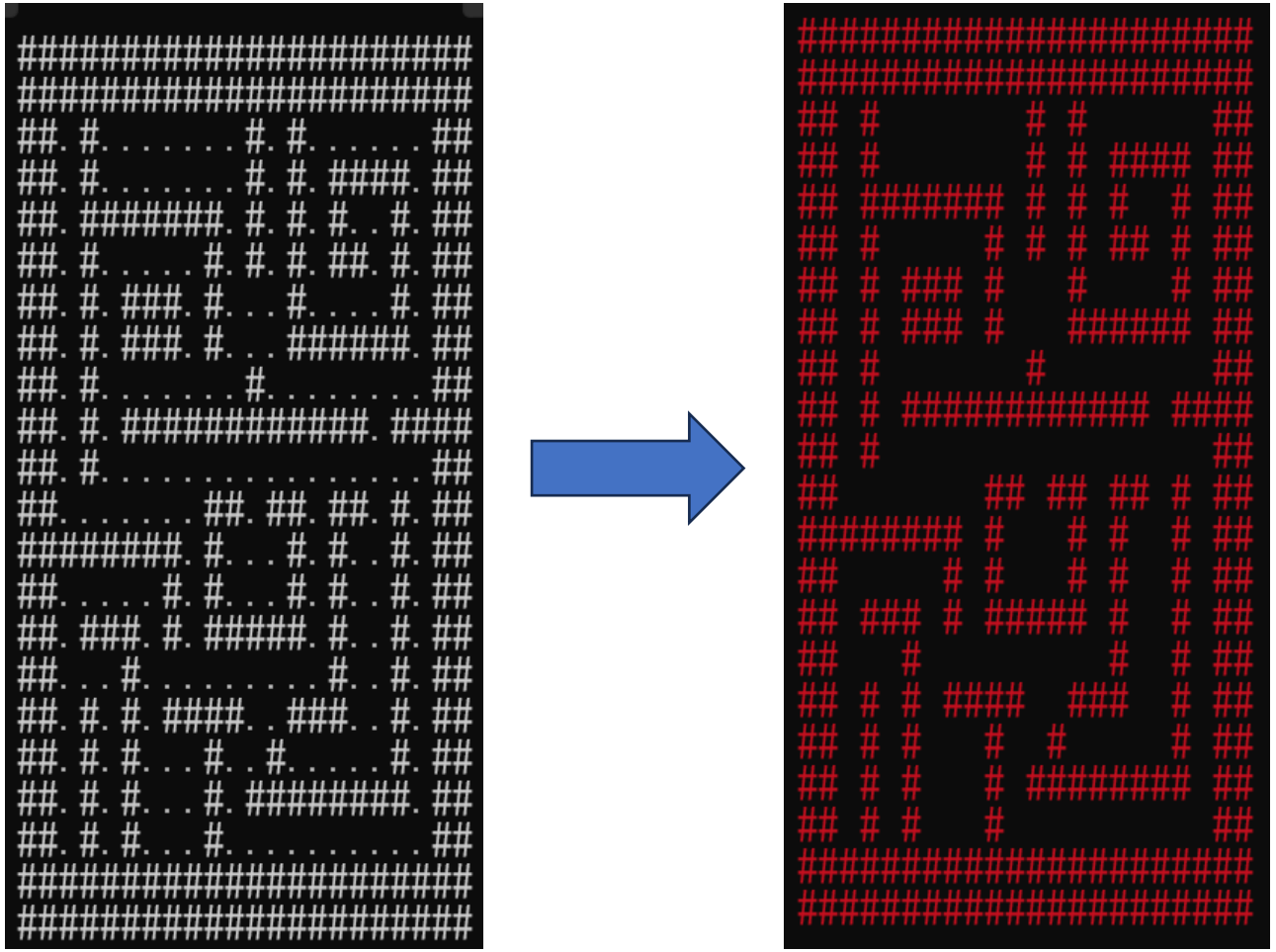
演習：2D迷路

- 壁と通路の色を変更する
  - 例
    ```
    cout << “¥033[31m赤色で表示¥033[m” << endl;
    ```
    表示したい色のエスケープシーケンスと、それをリセットするエスケープシーケンスで挟んで表示する

演習：2D迷路

- 壁と通路の色を変更する
  - 壁を赤色
  - 通路を黒色

  にしてみる



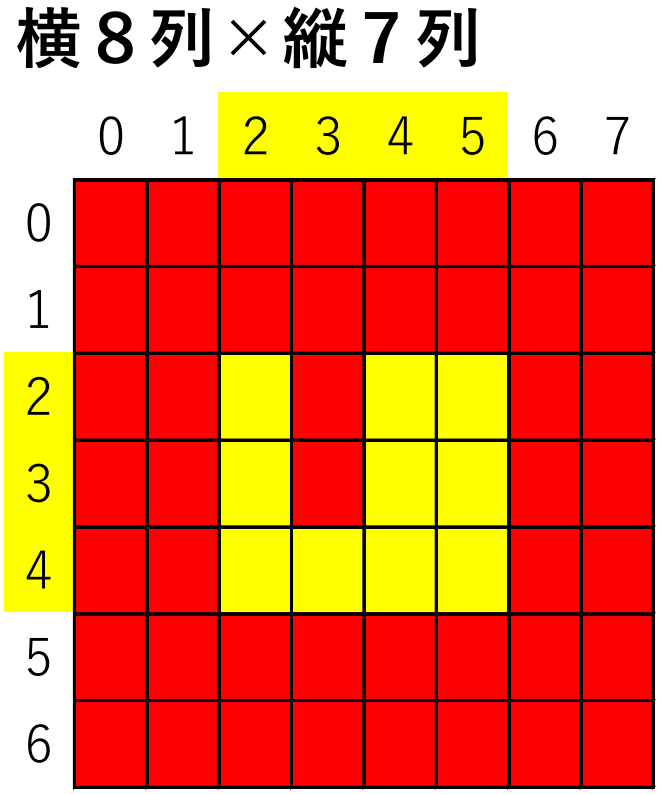
演習:2D迷路

map.cpp

```
void Map::DrawMap() {
  for (int y = 0; y < mapdata.getMapSize(); y++)
  {
    for (int x =0; x < mapdata.getMapSize(0); x++)
    {
      if (mapdata.getMapValue(x, y) == WALL) {
        cout << “¥033[31m#¥033[m”; //壁として#を出力
      }
      else if (mapdata.getMapValue(x, y) == ROAD) {
        cout << “¥033[30m.¥033[m”; //通路として.を出力
      }
    }
    cout << endl;
  }
}
```

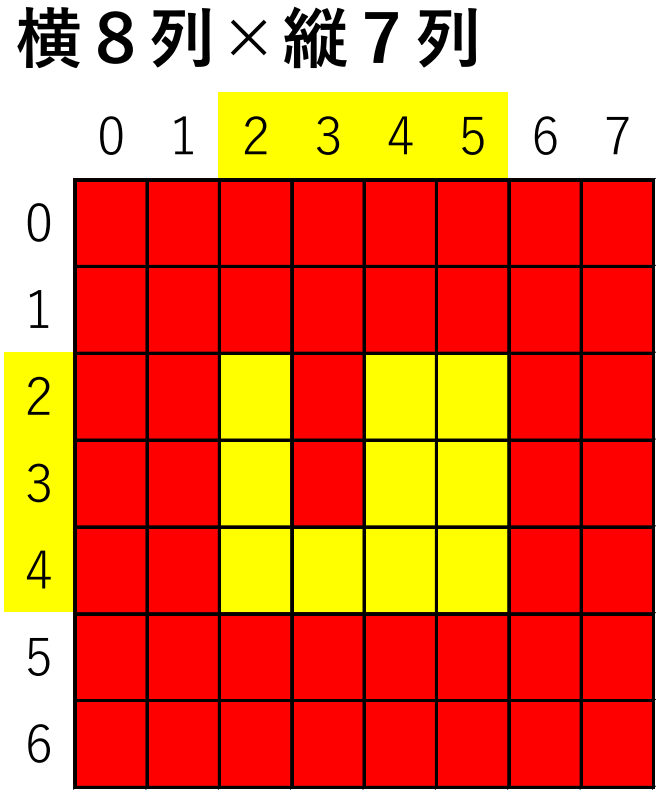
演習：2D迷路

- 迷路上のランダムな場所に宝物を配置する SetTreasure()関数
  - 乱数を使ってプレイする毎に異なる場所に配置
  - 配置するx座標は、外周を除いた2から(列数−3)の範囲
  - 配置するy座標は、外周を除いた2から(行数−3)の範囲



演習：2D迷路

- 迷路上のランダムな場所に宝物を配置する SetTreasure()関数
  - 配置予定場所が通路なら、その地点のvalueを2(TREASURE)にする
  - 配置予定場所が壁なら、再度乱数で場所を決定



演習：2D迷路

map.h

```
#pragma once
#include “maze.h”

class Map {
public:
  MapData mapdata; //MapDataクラスのインスタンス（マップデータ）
  Map();                                  //コンストラクタ
  void Load(string filename); //CSVファイルの読み込みと格納
  void DrawMap();                         //マップ描画
  void SetTreasure();                     //マップ上に宝物をランダム配置
};
```

演習：2D迷路

map.cpp

```
void Map::setTreasure() {
  random_device rnd_dev{};           //乱数生成器を作成
  mt19937 rand_engine(rnd_dev());
  //範囲指定した分布生成器(X,Y座標用)を生成
  uniform_int_distribution<int>
                          rnd_x(2, mapdata.getMapSize(0) - 3);
  uniform_int_distribution<int>
                          rnd_y(2, mapdata.getMapSize() - 3);
  while (true) {
    int trX = rnd_x(rand_engine); //X座標の乱数生成
    int trY = rnd_y(rand_engine); //Y座標の乱数生成
    //宝箱の位置が通路上であればOK。そうでなければ乱数生成を繰り返す
    if (mapdata.getMapValue(trX, trY) == ROAD) {
      mapdata.setMapValue(trX, trY, TREASURE); //宝物を配置
      break;
    }
  }
}
```

演習:2D迷路

map.cpp

```
void Map::DrawMap() {
  for (int y = 0; y < mapdata.getMapSize(); y++) {
    for (int x =0; x < mapdata.getMapSize(0); x++) {
      if (mapdata.getMapValue(x, y) == WALL) {
        cout << “¥033[31m#¥033[m”; //壁として#を出力
      }
      else if (mapdata.getMapValue(x, y) == ROAD) {
        cout << “¥033[30m.¥033[m”; //通路として.を出力
      }
      else if (mapdata.getMapValue(x, y) == TREASURE) {
        cout << “¥033[33m$¥033[m”; //宝物として$を出力
      }
    }
    cout << endl;
  }
```

演習:2D迷路

main.cpp

```
#include <iostream>
#include <conio.h>
#include <Windows.h>
#include “map.h“
using namespace std;
int main()
{
  Map map;                          //Mapクラスの生成
  map.Load(”2d_maze.csv”);//マップファイルの読み込みと格納
  map.setTreasure();          //宝物の配置
  system(“cls”);                    //コマンドプロンプトの画面消去
  //コマンドプロンプトの表示位置を(0,0)へ設定
  SetConsoleCursorPosition(
    GetStdHandle(STD_OUTPUT_HANDLE), COORD{ 0, 0 });
  map.DrawMap();                    //マップの描画
```

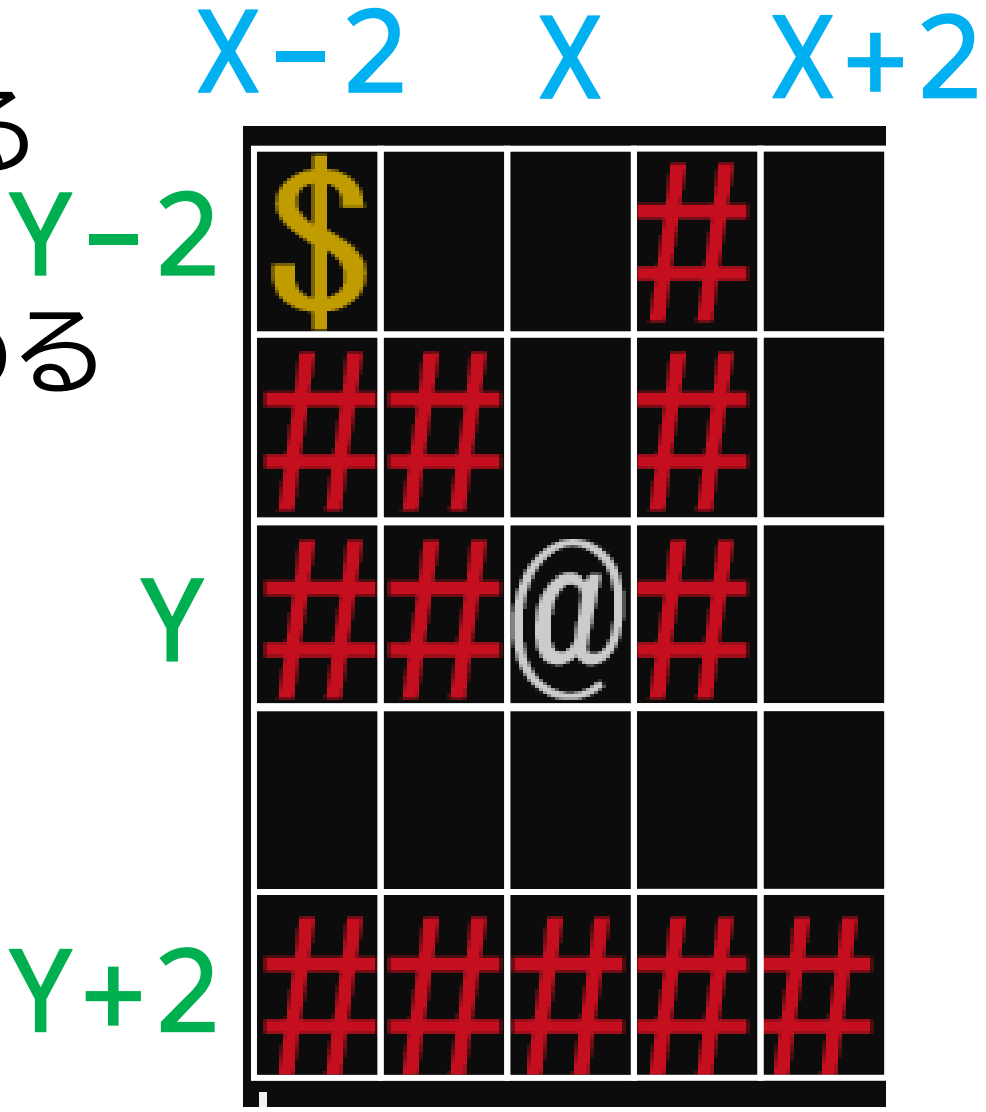
演習：2D迷路

- 迷路の表示範囲を5×5に限定する

① 自キャラの座標(X,Y)を求める

② 迷路の表示範囲を
X−2～X+2
Y−2～Y+2
に限定して表示する

③ 中心に自キャラを表示する

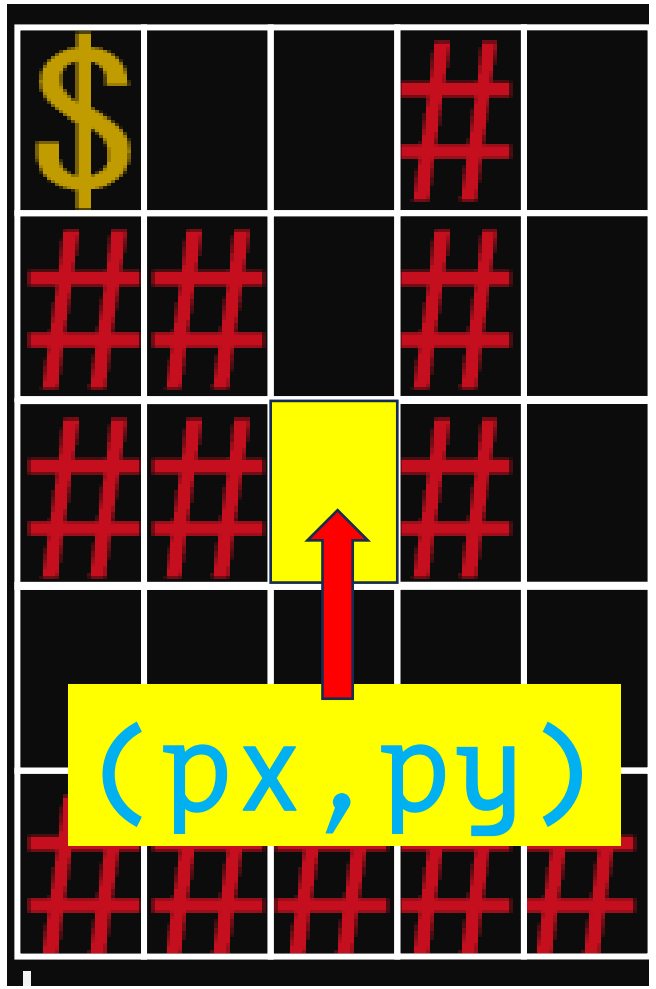


演習：2D迷路

- 迷路の表示範囲を5×5に限定する
  - Map::DrawMap()を

    Map::DrawMap(int px,int py)

    として、自キャラのX,Y座標を引数として受け取れるように変更して処理を行う



演習:2D迷路

map.h

```
#pragma once
#include “maze.h”

class Map {
public:
  MapData mapdata; //MapDataクラスのインスタンス（マップデータ）
  Map();                                  //コンストラクタ
  void Load(string filename); //CSVファイルの読み込みと格納
  void DrawMap(int px,int py);//マップ描画
  void SetTreasure();                     //マップ上に宝物をランダム配置
};
```

演習:2D迷路

map.cpp

```
void Map::DrawMap(int px, int py) {
  for (int y = py - 2; y <= py + 2; y++) {
    for (int x = px - 2; x <= px + 2; x++) {
      if (mapdata.getMapValue(x, y) == WALL) {
        cout << “¥033[31m#¥033[m”; //壁として#を出力
      }
      else if (mapdata.getMapValue(x, y) == ROAD) {
        cout << “¥033[30m.¥033[m”; //通路として.を出力
      }
      else if (mapdata.getMapValue(x, y) == TREASURE) {
        cout << “¥033[33m$¥033[m”; //宝物として$を出力
      }
    }
    cout << endl;
  }
```

演習:2D迷路

main.cpp

```
#include <iostream>
#include <conio.h>
#include <Windows.h>
#include “map.h“
using namespace std;
int main()
{
  Map map;                          //Mapクラスの生成
  map.Load(”2d_maze.csv”);//マップファイルの読み込みと格納
  map.setTreasure();          //宝物の配置
  system(“cls”);                  //コマンドプロンプトの画面消去
  //コマンドプロンプトの表示位置を(0,0)へ設定
  SetConsoleCursorPosition(
    GetStdHandle(STD_OUTPUT_HANDLE), COORD{ 0, 0 });
  map.DrawMap(2, 2);                    //マップの描画
```

演習:2D迷路

map.cpp

```
void Map::DrawMap(int px, int py) {
  for (int y = py - 2; y <= py + 2; y++) {
    for (int x = px - 2; x <= px + 2; x++) {
      if (x == px && y == py) {  //中心座標のとき自キャラ表示
        cout << "@";
      }
      else
      {
        if (mapdata.getMapValue(x, y) == WALL) {
          cout << "¥033[31m#¥033[m"; //壁として#を出力
        }
        else if (mapdata.getMapValue(x, y) == ROAD) {
          cout << "¥033[30m.¥033[m"; //通路として.を出力
        }
        else if (mapdata.getMapValue(x, y) == TREASURE) {
```

```
#####
#####
##@#
## #
## ##
```

演習：2D迷路

- 迷路内を移動できるようにする
  - 自キャラの情報を管理する`Player`クラスを作成する
  - 自キャラが指定の方向へ移動可能かどうかのチェックを行って、その結果で自キャラの座標を更新する
  - 更新した座標は`Map::DrawMap`の引数として与える

演習：2D迷路

- **PracMaze**フォルダ内に player.h, player.cpp を作成

```
copy nul player.h
copy nul player.cpp
```

演習：2D迷路

player.h

```
#pragma once
#include “maze.h”
using namespace std;
struct Vector2 {                                    //構造体Vector2
  int x, y;                                         //自キャラのX座標とY座標を管理
  Vector2(int x, int y) : x(x), y(y) {}; //コンストラクタ
};

class Player {
private:
  Vector2 Pos;              //構造体Vector2をPosとして定義
public:
  Player();                 //コンストラクタ（引数なし）
  Player(int x, int y);//      〃        （引数あり）
  void setX(int x);         //セッター
  void setY(int y);
  int getX();               //ゲッター
  int getY();
  void move(char key, MapData& mapdata);//マップ上で移動可能かを調べる
};
```

演習：2D迷路

player.cpp

```
#include "maze.h“
#include "player.h“
#include <iostream>
using namespace std;

Player::Player() : Pos(0, 0) {};
Player::Player(int x, int y) : Pos(x, y) {};
void Player::setX(int x) {
  Pos.x = x;
}
void Player::setY(int y) {
  Pos.y = y;
}
int Player::getX() {
  return Pos.x;
}
int Player::getY() {
  return Pos.y;
}
```

演習:2D迷路

player.cpp

```
void Player::move(char key, MapData& mapdata) {
  int newPosX = Pos.x;
  int newPosY = Pos.y;

  switch (key) {
  case 'w':              //上
    newPosY--;
    break;
  case 'a':              //左
    newPosX--;
    break;
  case 's':              //下
    newPosY++;
    break;
  case 'd':              //右
    newPosX++;
    break;
  case '@':              //強制終了用
    exit(0); }
```

演習：2D迷路

player.cpp

```
  //移動範囲チェック：X,Y座標ともに0以上かつ列数や行数より小さい
  if (newPosX >= 0 && newPosX < mapdata.getMapSize(0)
   && newPosY >= 0 && newPosY < mapdata.getMapSize())

  { //移動予測位置が壁でないかどうかのチェック
    if (mapdata.getMapValue(newPosX,newPosY) != WALL) {

      Pos.x = newPosX;    //移動可能ならプレイヤー位置を更新
      Pos.y = newPosY;

      //プレイヤーの位置が宝箱と重なるとき
      if (mapdata.getMapValue(newPosX, newPosY) == TREASURE)
      {
        cout << "¥033[33mお宝発見！！ゲームクリア¥033[m" << endl;
        exit(0);
      }
    }
  }
}
```

演習：2D迷路

main.cpp

```
#include <iostream>
#include <conio.h>
#include <Windows.h>
#include “map.h“
#include ”player.h“
using namespace std;
int main()
{
  Map map;                          //Mapクラスの生成
  map.Load(”2d_maze.csv”);//マップファイルの読み込みと格納
  map.setTreasure();                //宝物の配置
  Player player(2,2);               //出現位置の設定(X,Y) = (2,2)
  system(“cls”);                    //コマンドプロンプトの画面消去
  SetConsoleCursorPosition(
    GetStdHandle(STD_OUTPUT_HANDLE), COORD{ 0, 0 });
  map.DrawMap(player.getX(), player.getY());
```

演習：2D迷路

main.cpp

```
 system("cls");   //コマンドプロンプトの画面消去
 while (true)     //ゲームループ
 {
    // 標準出力画面のカーソル位置を(0,0)へ設定する
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE)
                                          , COORD{ 0, 0 });

    map.DrawMap(player.getX(), player.getY());
    char input = _getch();
    player.move(input, map.mapdata);
 }
```

演習：2D迷路

- 自キャラの初期位置をランダムにする
  - 乱数生成を用いて、自キャラの初期位置を迷路上の通路の上に設定する
  - MapクラスのSetTreasure関数と同様に、乱数の初期位置が壁になるようなら、再度乱数を求めなおす
  - main.cpp内で実装する

演習：2D迷路

main.cpp

自キャラの位置を乱数で決定する

```
#include <iostream>
#include <conio.h>
#include <Windows.h>
#include “map.h“
#include ”player.h“
#include <random>
#include <algorithm>
using namespace std;
int main()
{
  Map map;                          //Mapクラスの生成
  map.Load(”2d_maze.csv”);//マップファイルの読み込みと格納
  map.setTreasure();                //宝物の配置
  Player player;                    //出現位置を定めずplayer生成
  system(“cls”);                    //コマンドプロンプトの画面消去
```

演習：2D迷路

main.cpp

```
random_device rnd_dev{};
mt19937 rnd_engine(rnd_dev());
uniform_int_distribution<int>
  rnd_x(2, map.mapdata.getMapSize(0) - 3);
uniform_int_distribution<int>
  rnd_y(2, map.mapdata.getMapSize() - 3);
while (true) {
  int px = rnd_x(rnd_engine);//x座標用の乱数生成
  int py = rnd_y(rnd_engine);//y座標用の乱数生成
  if (map.mapdata.getMapValue(px, py) == ROAD) {
    player.setX(px);//生成した場所が通路の上なら
    player.setY(py);//セッターで自キャラの位置を設定
    break;
  }
}
while (true) {
```

自キャラの位置を乱数で決定する

演習：2D迷路

- 宝物を守るモンスターとの戦闘を入れる
  - 以前C言語で作成したRPG風コマンドバトルを組み込む
  - 共有フォルダ＞C++ソースプログラム＞PracMaze_0内の
    battle.h
    battle.cpp
    chara.csv
    mob.csv
    skill.csv
    をPracMazeフォルダへコピー

演習：2D迷路

- 宝物を守るモンスターとの戦闘を入れる
  - 以前C言語で作成したRPG風コマンドバトルプログラムを`Battle`クラスとしてC++に書き換え済み
  - `main`関数から`Battle`クラスのメンバ関数を呼び出してコマンドバトルを実行
  - モンスターを倒せば宝物ゲット、やられるとゲームオーバー

演習：2D迷路

player.h

戦闘モードを追加する

```
#pragma once
#include “maze.h”
using namespace std;
struct Vector2 {                                    //構造体Vector2
  int x, y;                                         //自キャラのX座標とY座標を管理
  Vector2(int x, int y) : x(x), y(y) {}; //コンストラクタ
};

class Player {
private:
  Vector2 Pos;              //構造体Vector2をPosとして定義
public:
  Player();                 //コンストラクタ（引数なし）
  Player(int x, int y);//      〃        （引数あり）
  void setX(int x);         //セッター
  void setY(int y);
  int getX();               //ゲッター
  int getY();
  int move(char key, MapData& mapdata);//マップ上で移動可能かを調べる
};
```

演習：2D迷路

player.cpp

```
  { //移動予測位置が壁でないかどうかのチェック
    if (mapdata.getMapValue(newPosX,newPosY) != WALL) {

      Pos.x = newPosX;    //移動可能ならプレイヤー位置を更新
      Pos.y = newPosY;

      //プレイヤーの位置が宝箱と重なるとき
      if (mapdata.getMapValue(newPosX, newPosY) == TREASURE)
      {
        return 99;
        cout << "¥033[33mお宝発見！！ゲームクリア¥033[m" << endl;
        exit(0);
      }
    }
  }
  return 0;
}
```

宝物を発見したらリターンコード99でmain関数へ戻る

演習：2D迷路

main.cpp

```
Battle battle;  //Battleクラスのインスタンス生成
while (true)    //ゲームループ
{
  SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE)
                                        , COORD{ 0, 0 });
  map.DrawMap(player.getX(), player.getY());
  char input = _getch();
  if (player.move(input, map.mapdata) == 99) {//宝物発見
    if (battle.Initialize() == 0) { //バトルに勝利したとき
      cout << “¥033[33mお宝発見！！ゲームクリア¥033[m” << endl;
      return 0;
    }
    else {//バトルに負けたとき
      cout << "¥033[31mゲームオーバー…¥033[m" << endl;
      return 0;
    }
  }}
```