解答

解答とは言っても一例にすぎません．若干異なっていたとしてもプログラムが動きさえすれば問題ありません．

・穴埋め解答

穴埋め 1：DrawGraph(0, 0, img.clear, TRUE);

DX ライブラリのライブラリ関数です．第 1 第 2 引数が，画像の左上の xy 座標です．

穴埋め 2 ：KEY_INPUT_ESCAPE

穴埋め 3：clear();

触れているブロックがゴールなので，ゲームクリアです．

穴埋め 4：

```
for (int j = 0; j<HEIGHT_SIZE; j++) {
        for (int i = 0; i<WIDTH_SIZE; i++) {
                if (map[i][j] != 0)
                        judge_h(i*BLOCK, j*BLOCK);
        }
}
```

呼び出し方は上下の壁判定と同じです．左右の動きを制限するので，主人公の座標（player.x,player.y）を変化させる前に呼び出す必要があります．

穴埋め 5：CENTER + ATTACK_SIZE

ATTACK__SIZE は HERO_SIZE でも大丈夫です．…というか HERO_SIZE のほうがマクロの考え方からすると正しいです．

穴埋め 6：player.y

穴埋め 7：CENTER - ATTACK_SIZE

・左右壁判定の補足

実行すればわかると思います．例えば，主人公の右に壁があり，その壁にぶつかっているとします．このとき+4,-4 が書いてないと左にも動けなくなってしまいます．その逆も起こります．なぜかというと，例でいえば，右の壁にぶつかっているときも右だけでなく左にも壁

がある，と判断してしまうからです（デバッグ用の変数描画を見るとよくわかると思います)．

そして，+4,-4 なのは，主人公の X 方向の速さが 4 だからです．

わからないところがあったら質問をぶつけましょう．