

コードの工夫点

- ・構造体で座標一致をノード構造体と座標構造体両方でできるようにした点
 - ・コーディングする上での利便性が向上した
 - ・他情報も格納されていることもあり `operator ==()` は慎重に書くべきだが、今回は座標以外の情報を比較しないため実装に踏み切った
- ・後々変えそうな部分やデバッグ周りのコードを分けた点
 - ・完成時に不要となるコードをまとめて 1 つの `bool` で管理した
 - ・プリプロセッサ命令も考えたが慣れないため今回はこちらで一括管理した
 - ・色変更も列挙型で呼び出し、カラーコード自体は 1 個にまとめたため微妙な色変更をする際に役立つ
 - ・外部ファイルにまとめたら更に見やすく変更もしやすいと感じた これはゲーム制作で若干活かすことができた
- ・関数化した点
 - ・ `between` 関数(比較値、最小値最大値内にあるかを `bool` で返す)ほど作ってよかった関数は無い
 - ・単純に見やすさが向上した
 - ・上記にも当てはまる
- ・4 方向と 8 方向に対応した点
 - ・初期は 8 方向のみで、4 方向はコピペで実装していたため行数が無駄に長くなっていたが、最終的に引数で受け取れるようにして実行部分の行数を 1/2 にできた

コードの苦労した点

- ・ `enum` でループさせる方法
 - ・時間が経った今は慣れたが、実装当初はかなり詰まった
 - ・もしかしたらよろしくない実装方法かもしれない
- ・ `f,g,h` を求める際に値がおかしくなる不具合
 - ・初歩的な問題だが、ポインタが破棄されることで発生する不具合に悩まされていた (`int` が初期値に戻る、変な値になるなどでゴールしない問題)
 - ・他機構も同時並行しつつ 1 週間程度詰まったが無事解決