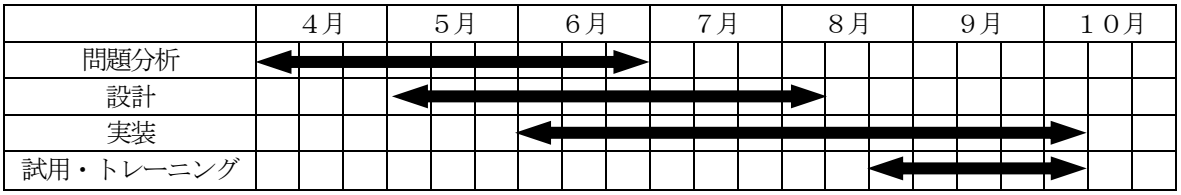
部　　門	競 技 部 門	No.1 登録番号	30046

No.2

1) 予定開発期間：6ヶ月
2) 予定開発人数：3人



No.3

実現方法

1) フィールド上のペアを多く作るアルゴリズム

勝敗条件の最優先事項は「フィールド上のペアを多くつくること」であるため、手数は気にせず着実に多くのペア数を作成することを検討する。

具体的には、周りのエンティティへの干渉を考えずに、あるエンティティを特定の位置まで移動させる導きを BFS にて導出する。また、固定したいエンティティを指定して二次元累積和を適用することによって、園に含むかを判定できる。これらの操作により、ペアをフィールドの端から敷き詰めることが可能となる。これらの処理を続けていって、最後に 4x4 だけが揃っていない状態になれば、BFS を適用し盤面を揃えられる。

また、フィールドのペアをすべて揃えられる判定の必要条件として、盤面のマンハッタン距離を検討する。

具体的には、奇数の園での導きはマンハッタン距離の偶奇が変わらないこと、さらに偶数の園での導き 2 回適用では、マンハッタン距離の偶奇が変化しないことに着目する。

2) 導きの手数を少なくする工夫

SIMD を使用した効率的な低レベル処理を検討する。併せて、OpenMP, MPI を使用した並列処理を検討する。

加えて、我々はオセロやルービックキューブといった既存のゲームに対するヒューリスティック解法を活用し、効率良く精度の高いプログラムを開発していく予定である。

具体的には、フィールドのエンティティ配置にはある程度のパターンが定まっていると考え、ペア化するのに最適化した導きを優先的に適用することで、手数を少なくする。

また、回答のパターンを複数想定し、それぞれの想定パターンに合わせるための導きを試行錯誤的に導出する。

3) その他（独創的なところ）

4x4 のフィールドでは、BFS により貪欲的に導きを導出できるため、いかなるサイズのフィールドでも、4x4 の不揃いフィールドができるまでの手数を最適化することに重きを置く。

また、手順が正しいことを簡単に検証するため、Nuxt.js を使用した Web ビジュアライザを作成する。

具体的には、JSON 形式で提供される盤面情報を丸ごとパースすることによって、情報を加工する時間を極力減らしながら可視化する。また、導出した回答をビジュアライザの盤面へ自動で適用できるようにすることで、視覚的に回答の正誤と誤り量を判断することができる。

No.4

開発環境
使用言語：C++, TypeScript
エディタ：Vim, Visual Studio Code
ライブラリ：Nuxt.js