

## 応募内容(競技部門用)

<b>No.1</b>	タイトル	Kyogix—きよぎつくす—
<b>No.2</b>	1) 予定開発期間：5ヶ月 2) 予定開発人数：3人	
<b>No.3</b>	<p>実現方法</p> <p>本プログラムは試合情報入力部分、盤面状況入力部分、思考部分、思考結果表示部分からなる。以下に思考部分の仕様を示す。</p> <p>思考部分は次のような流れで行う。</p> <ol style="list-style-type: none"> <li>1. 盤面の状況を元に、現在の評価値を得る。</li> <li>2. 現在の状態空間から選択可能な全ての状態を考慮する。</li> <li>3. 考慮した結果、評価値が一番良い手を次の手として採用する。</li> </ol> <p>評価値は次のような複数の評価関数の線形結合で求める。</p> $\text{評価値} = a \times \text{評価関数 1(状態)} + b \times \text{評価関数 2(状態)} + c \times \text{評価関数 3(状態)} + \dots$ <p>ただし、a, b, c, はユーザーが設定するパラメータである。</p> <p>以下に導入予定の6つの評価関数を示す。</p> <ol style="list-style-type: none"> <li>1. 現在の自陣地を評価する関数</li> <li>2. 現在の敵陣地を評価する関数</li> <li>3. 今後、自陣地になる可能性がある陣地を評価する関数</li> <li>4. 今後、敵陣地になる可能性がある陣地を評価する関数</li> <li>5. 今後、自陣地になる可能性がない陣地を評価する関数</li> <li>6. 今後、敵陣地になる可能性がない陣地を評価する関数</li> </ol> <p>陣地の評価値は陣地の数ではなく、陣地の位置を数値化した評価関数用データベースを用いて計算する。例えば、「フィールドの端の陣地より中央の陣地のほうが良い値とする」などと設定しておく。</p> <p>最良手を探索する方法として、人工知能の手法である「ミニ・マックス法」や枝刈りにより効率的に探索できる「<math>\alpha</math> <math>\beta</math>法」などを行い、制限時間に間に合うように効率的に先読みする方法を導入する。</p> <p>本システムの設計方法として、複数人で効率的に開発できるようにするために、オブジェクト指向モデリングで行い、UMLを用いて、クラス図、シーケンス図、状態遷移図を作成し、開発している。</p>	
<b>No.4</b>	使用ソフト	開発言語：Java 言語、開発環境：フリーのエディタ
<b>全国高等専門学校 第17回プログラミングコンテスト：茨城</b>		