

部 門	競 技 部 門	No.1 登録番号	30016
-----	---------	-----------	-------

No.2	1) 予定開発期間：6か月																																								
	2) 予定開発人数：3名																																								
	<table border="1"> <thead> <tr> <th></th> <th>4月</th> <th>5月</th> <th>6月</th> <th>7月</th> <th>8月</th> <th>9月</th> <th>10月</th> </tr> </thead> <tbody> <tr> <td>問題分析</td> <td colspan="2">←————→</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>設計</td> <td></td> <td colspan="3">←————→</td> <td></td> <td></td> <td></td> </tr> <tr> <td>実装</td> <td></td> <td></td> <td colspan="4">←————→</td> <td></td> </tr> <tr> <td>試用・トレーニング</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td colspan="2">←————→</td> </tr> </tbody> </table>		4月	5月	6月	7月	8月	9月	10月	問題分析	←————→							設計		←————→						実装			←————→					試用・トレーニング						←————→	
		4月	5月	6月	7月	8月	9月	10月																																	
	問題分析	←————→																																							
設計		←————→																																							
実装			←————→																																						
試用・トレーニング						←————→																																			

No.3	<p>実現方法</p> <p>1) 陣地の取得アルゴリズム</p> <p>自チームの陣地の情報は配列に保存しておく。ターンの推移後、以下のようにこの配列を更新することを考える。</p> <ol style="list-style-type: none"> 自チームや相手チームを問わず職人が新しく壁を建築した場所のうち陣地であった場所を配列から削除する。 フィールドの端全てを始点とした4方向への幅優先探索を行うことで、閉鎖された陣地でない場所を列挙する。この際に列挙されなかったフィールドが閉鎖された陣地である。これを自チームと相手チームの壁のみをそれぞれ考慮して2回行う。 この探索で得た相手チームの閉じた領域の場所を配列から削除し、その後に自チームの閉じた領域の場所を配列に追加する。 <p>以上の方法により常に自チームの陣地を取得できるようになる。相手チームの陣地も同様である。</p>
	<p>2) 職人の行動決定方法</p> <p>モンテカルロ木探索で行動を決定する。しかし、モンテカルロ木探索のみでは有効な着手が得られない可能性がある。その場合には人が考えた有効な作戦を優先的に採用することにする。例えば、城を優先的に囲む作戦などが挙げられる。これにより、探索結果があまり良くない場合にも最低限の良い行動を保証する。この場合でも、十分に信頼のできる結果が得られなかった場合に、次のようなアルゴリズムの使用も検討する。</p> <p>一つはThunderサーチを用いる。これは、モンテカルロ木探索のプレイアウト部分を評価関数による盤面評価に置き換えた探索アルゴリズムである。本競技においては、得点は勿論、フィールドの有利位置や他職人や城との距離などを用いて評価関数を設計する余地があると考えており、理想的な評価関数を用意することで、プレイアウトの時間を大幅に削減し高速かつ効率的に探索することができる。</p> <p>もう一つは非同期方策価値更新モンテカルロ木探索を用いる。これは、モンテカルロ木探索に、次の一手の確率を予測する方策ネットワークと盤面を評価する価値ネットワークの二つのニューラルネットワークを用いた機械学習を加え、より良い手をより深く探索することができるものである。</p>
	<p>3) その他（独創的などころ）</p> <p>ビジュアライザを作成し、現在の戦況をリアルタイムで視認可能にする。また、ビジュアライザに運営との通信を代理させることで、メインプログラムの通信部分を削減できる。我々のチームは通信部分の担当とメインプログラムのアルゴリズムの実装部分の担当が異なるため、これにより仕事の分担がより簡単になる。</p> <p>また、複数のアルゴリズムの中からフィールドによって最適なアルゴリズムを選択することができる。</p> <p>アルゴリズムには、ニューラルネットワークを用いた機械学習も含まれる。事前にフィールドなどの情報が公開されることを利用し、与えられたフィールドに対して学習を行うことで、実質的に探索時間を増やすことができるといえるだろう。有名な囲碁AIのAlphaGoに用いられていた非同期方策価値更新モンテカルロ木探索をはじめ、さまざまな手法について検討していきたいと考えている。</p>



図1 ビジュアライザの動作画面

No.4	<p>開発環境</p> <ul style="list-style-type: none"> Python 3.9 (NumPy 1.23.5, Cython 0.29.34, Keras 2.10.1, TensorFlow 2.10.1) C++ (OpenSiv3D 0.6.8) Java17
------	---