

部 門	競 技 部 門	No.30021 登録番号
-----	---------	---------------

No.2	1) 予定開発期間：6 カ月 2) 予定開発人数：3 人									
		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	実現方法									
	<p>1) 陣地の取得アルゴリズム</p> <p>今回のゲームにおいて、解放された陣地は、ターン前の陣地状態と比較することで判断可能である。ターン前に囲われており、ターン後に解放されている、かつターン後に相手に囲われていなければそれは解放された陣地である。よって解放された陣地は考えず、囲われた陣地の取得アルゴリズムのみを考えれば良い。</p> <p>マスをもととして見た時、4方向には辺が張られており、連結であればそれが陣地となる。辺が切られるとき、そこには壁が存在する。よって連結を高速に調べるデータ構造である素集合データ構造を用いる。</p> <p>計算量は盤面の高さを H、幅を W として結合処理と結合判断がそれぞれ $O(\alpha(H \times W))$ であり、アッカーマン関数の逆関数に比例するため高速である。</p> <p>また、更なる高速化手法としてスキャンラインアルゴリズムがあるため、実装の余裕があればそちらに切り替える。</p>									
	<p>2) 職人の行動決定方法</p> <p>明らかな善手について考える。味方の職人が、最短経路で城へ向かい壁を閉じるまでのターン数が、敵の職人が最短経路で城へ向かい妨害するターン数より短ければ、他の手を選ばず行った方が良い。</p> <ul style="list-style-type: none"> ・平均的な善手の推定 (1)モンテカルロ法による優良建築マスの推定 <p>味方の壁のみを考えた盤について以下を実行する。また、建築優先度を保存する盤を考える。</p> <ol style="list-style-type: none"> 0. 現在の盤面に初期化する。 1. ランダムにマスを決め注目箇所とする。 2. 壁を設置し、ランダムに決定した近傍8方向に注目箇所を更新する。これを任意の k 回繰り返す。 3. 盤面の閉鎖マスの数をカウントし、設置した k 箇所の建築優先度を加算する。 4. 一定時間が過ぎるまで 0~3 を繰り返す。 5. 建築優先度の盤を、味方職人との距離に対して単調減少、敵職人との距離に対して単調増加するように変化させる。この時、距離はある地点→敵の壁は距離2、ある地点→空マスは距離1、ある地点→池は距離 INF としてマスの近傍は8方向のダイクストラ法によって定める。 <ul style="list-style-type: none"> (2)モンテカルロ法による優良破壊マスの推定 <p>最初に優良建築マスの推定を敵について行う。また、敵の壁のみを考える。</p> <p>建築優先度が一定以上のマスについて盤面に敵の壁を建築し、閉鎖を検出する。</p> <ol style="list-style-type: none"> 0.現在の盤面に初期化する。 1. ランダムにマスを決め注目箇所とする。 2.そのマスと近傍1マスを破壊し、閉鎖を検出する。 3.閉鎖マスの減少量を破壊したマスに加算する。 4.一定時間が過ぎるまで 0~3 を繰り返す。 5.味方職人との距離に対して単調減少になるように変化させる。距離の定義は優良建築マスと同じである。以上の優先度を上位から満たすように職人を移動させる。 									
<p>3) その他 (独創的なところ)</p> <p>優先度について、GUI を操作しマスの優先度を人為的に操作できるようにする。</p>										

No.4	<p>開発環境</p> <p>OS:Windows10 home edition,Windows11 home edition,macOS Ventura13.3.1</p> <p>プログラミング言語:C++23,Python3.7</p> <p>フレームワーク:OpenSiv3D</p>									
------	---	--	--	--	--	--	--	--	--	--