

応募内容(競技部門用)

No.1	タイトル	Inteli Pentomino Processor
No.2		<p>1) 予定開発期間： 5 ヶ月</p> <p>2) 予定開発人員： 3 人</p>
	実現方法	<p>本作品は、問題入力部分、展開図作成部分、展開図敷き詰め部分、解答出力部分の 4 つの部分から構成される。以下に各部分の基本仕様を示す。</p> <p>1. 問題入力部分 問題入力部分は、立体入力モードと面編集モードの 2 つを持ち、立体入力モードで立体の形を入力した後に、面編集モードで「抜けている」面を削除する。 また、入力的高速化のために、LAN で繋がれた 2 台の PC で分担して立体を入力できるようにする。</p> <p>2. 展開図作成部分 立体の一つの面を基準にして、その面に接している面を繋ぐ / 繋がらないの全ての組み合わせ(接している面が 4 つなら 16 通り)を再帰的に調べていく事で、全ての展開図を作成する。</p> <p>3. 展開図敷き詰め部分 基本的に、長方形領域を角から順番に敷き詰めていく。ただし、敷き詰め方の組み合わせは非常に多い為、以下の方法で効果的に良い解を見つける。</p> <p>No.3 また、バックアップとして GA を用いた展開図敷き詰めアルゴリズムを開発し、片方の PC で実行する事を現在考慮中である。</p> <p>3.1. 使用展開図の限定 一つの立体の展開図は、回転体を除いても数百から数千を数える。毎回、特定の地点に設置可能な展開図を全ての展開図から検索するのは非効率的である。 そこで、注目している地点に設置可能な展開図と、その地点の周囲の状況をハッシュを用いて保存しておく。以降、別の場所に注目している時に周囲の状況が以前に出てきた物であれば、保存しておいた設置可能展開図を用いる。</p> <p>3.2. 最良優先探索 ある場面で、展開図設置が終わった場所に残っている隙間や、これから展開図を設置する空間にある展開図を敷き詰めにくい場所(例:3 方向を囲まれた場所)を検出し、そこからその場面のコストを算出する。 このコストが少ない場면을優先的に探索する事で、効率的な探索を実現する。</p> <p>4. 解答出力部分 未定(解答入力ソフトの仕様が公開されてから、解答出力部分の仕様を決定する)</p>
	使用ソフト	
No.4		<p>問題入力・解答出力部分： Visual Basic</p> <p>それ以外： Visual C++</p>
全国高等専門学校 第 14 回プログラミングコンテスト：東京		