グループ4
発表

役割分担

- 佐藤 嵩晟
  GUI作成担当

- 塩谷 昂平
  スライド作成担当

- 長谷川 達也
  プランニング拡張担当

- 瀬戸口 大貴
  プランニング拡張担当

プランニングの仕様について

プランニングの仕様について

ゴールリストの並び替え

- 積まれているオブジェクトは下から順に状態が並ぶように内部的に
- 他のオブジェクトと関係がないゴール状態は後ろに来るように

[例]

{ontable E, A on B, C on D, B on C}

→ {C on D, B on C, A on B, ontable E}

プランニングの仕様について

ゴール状態の追加①

・初期状態で、一番下にくるべきと思われるオブジェクトに対して、ユーザがそれを記述していない場合、“ontable -”をリストに追加

[例]
{B on C, A on B, ontable D}
→ {ontable C, B on C, A on B, ontable D}

プランニングの仕様について

ゴール状態の追加②

- その時点で対象となっているゴール状態を満たすために操作したいオブジェクトについて”clear -”が存在していない場合、そのオブジェクトの上に載っていると判別できるオブジェクトについて”ontable -”を一時的なゴール状態としてプランニングを行う

[例]
ゴール状態: A on B,
場の状態: {C on B, clear A, ontable B, ontable B}
→ ゴール状態: ontable C

プランニングの仕様について

場の状態の並び替え

- 対象のゴール状態を満たすためにオペレータに対する変数束縛がやりやすいように、ゴール状態に使用されている順でオブジェクトの状態がリストの前にくるように逐次並び替えている

[例]
ゴール状態: A on B,
場の状態:　{clear C, clear B, clear A}
→ {clear A, clear B, clear C}

プランニングの仕様について

オペレータの制御

・現在使おうとしているオペレータに対して変数束縛を行った後、一つ前で使用したオペレータの変数束縛と比較して、無駄な行動であると判断される場合、別のオペレータを使用するように進路変更

[例]
現在: put A down on the table
一つ前: pick up A from the table
→ 変更!

プランニングの仕様について

考察

- ゴール状態の追加②の処理をゴール状態から置くべき場所を探索させるとさらに最適化されるが、現在満たしたいゴール状態には関係ないのと、”ontable -”にしておけば他のゴール状態のときに邪魔にならないので、今回のような実装にした

GUIについて

GUIについて

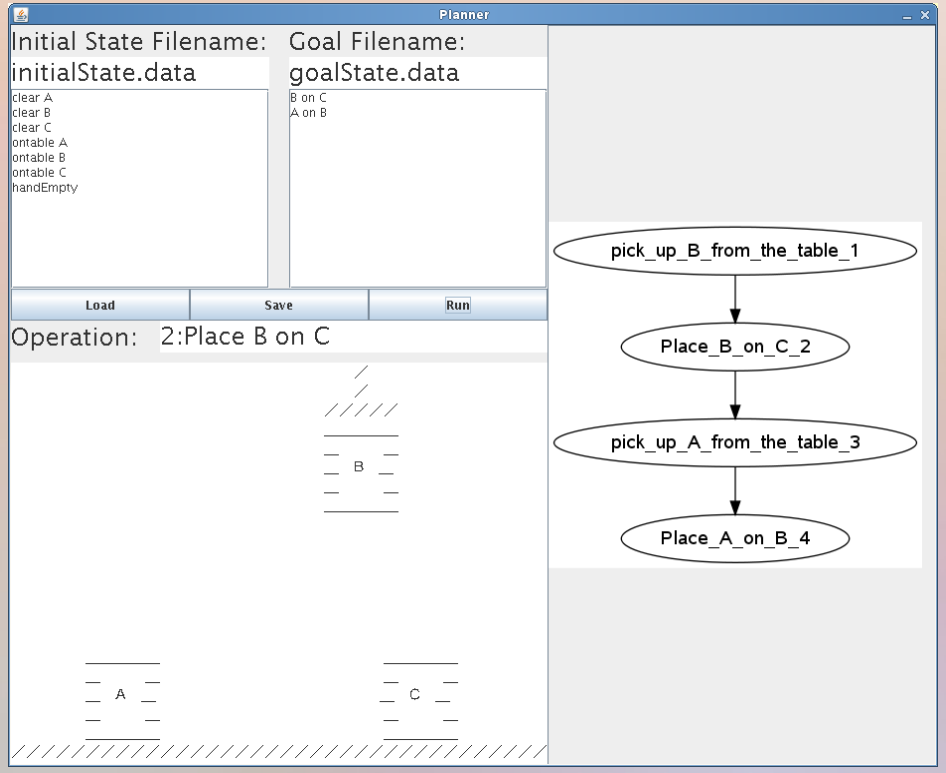
仕様

- 初期状態、ゴール状態を入力(ファイルから読み込みでもOK)
- Runボタンを押して実行
- 手順と実際の動きを出力

拡張前の問題点

- 初期状態が課題7のものにしか対応してない。
- 形が四角形しかない。
- A～Cの3種類でないとダメ。
- プランニング実行直後に動きが始まってしまう。

etc...



拡張した点

- 初期状態がデモのものにしか対応してない。
  →任意の状態に対応。
- 図形が四角しかない。
  →他の図形にも対応(今回は三角形と円)。
- A～Cの3種類でないとダメ。
  →ある程度の個数に対応。
- プランニング実行直後に動きが始まってしまう。
  →実行と動きの出力をそれぞれ別のボタンで。

課題

- 色を指定できるようにする。
- 形による制約を追加する。
- 初期状態にholding ?xが含まれていても対応できるようにする。
- 動きの一時停止や、1ステップごとの表示をできるようにする。

感想1

本当はもう一つプログラムを作成する予定だったが、うまくプランニングに適応させることが出来なかった。

取り扱う問題によって解法に向き不向きがあるのでそれをもう少し考えた上で決定すべきだった。

感想2

この課題を通して...

- グループワークの難しさ
- 報告や進捗管理の重要性
- プログラムを読みやすく書くことを意識する