知能情報工学実験演習II
C++演習 (Part 2)

下薗・嶋田・香野・小松

グループ演習

- 1グループ：5〜6人
  - グループ単位でプログラムを作成する
- 対象となるゲームは 大富豪/大貧民
- 演習課題
  - 知的な戦略を持つプレイヤーを実装すること
- テキスト
  - /home/i/shimada/DAIHUGOU2012/daihugou2012-1.pdf
- この資料
  - /home/i/shimada/DAIHUGOU2012/daihugou2012-2.pdf

大富豪/大貧民

- 52枚のカードとジョーカー1枚を使用する.
- 強さはジョーカー>2>A>K>Q>J>10>...>3の順とする.
- 各プレイヤーは現在場に出ているカード(リード)よりも強いカードを出さなければならない.
- 出せなかったり，出したくない場合はパスをする.
- 全員がパスをする状態になったら，それまで出たカードは捨てて，最後にカードを出したプレイヤーが新たなリードとなる.
- 最終的に最も早くカードがなくなったプレイヤーが大富豪となる.
- リードの出したカードが2ペアもしくは3カード，4カードならば，プレイヤーは同じ枚数でより強いカードを出さなければならない.
- 今回の演習では，シークエンスや革命，都落ち，その他地方ルールは考えない.

サンプルプログラム

- サンプルプログラム
  - /home/i/shimada/DAIHUGOU2012
    - Card.{h|cc}
    - CardSet.{h|cc}
      - 基本的には前回までと同じものだが，若干の拡張があるのでこちらを使うこと
    - Dealer.{h|cc}
      - ゲームの管理を行う
    - Player.{h|cc}
      - 基本となるプレイヤー
    - LittleThinkPlayer.{h|cc}
      - 拡張プレイヤーのサンプル
      - 実際には Player のサブクラス
    - ThinkTA1.{h|o}
      - 昨年度TA作成思考ルーチン
    - main.cc

C++ Card.cc CardSet.cc Dealer.cc Player.cc LittleThinkPlayer.cc ThinkTA1.o main.cc

Dealer.{h|cc}

- ゲームを管理するクラス
  - プレイヤー管理
  - カードの配布
  - 場に出されたカードが受理できるかの判定など
  - 人間が遊ぶ場合とは異なり，プレイヤーではない
- 演習中でこの中の変数や関数を変更してはいけない

Player.{h|cc}

- プレイヤーのクラス
  - 手持ちのカードとプレイヤー名
  - 重要なメンバ関数は
    - bool follow(CardSet &, CardSet &);
      - どのカードを出すか
    - bool approve(CardSet &, int[]);
      - 他人が場に何を捨てたかをチェック
      - LittleThinkPlayerを参照
  - この2つ以外の関数を変更しないこと
    - follow()およびapprove()でのみ使用する関数は作ってよい

Player::follow()

- 場のカード pile と出すカード s
- もっともシンプルなプレイヤー
  - ランダムにカードを引いて出す

```
bool Player::follow(CardSet & pile, CardSet & s) {
        Card tmp;
        s.makeempty();
        hand.pickup(&tmp, -1); // anyway, choose a card.
        s.insert(tmp);
        return true;
}
```

- CardやCardSetの必要なメンバ関数を使う
  - カードの大小関係比較：bool isGreaterThan(Card c)
  - 等しいかどうか：bool equal(Card tgt)
  - ジョーカーかどうか：bool isJoker()

Player::approve()

- 他人の出したカードを確認する
  - デフォルトは何もしない
- 引数で受け取った pile には，現在の場に出ているカードが渡される
  - これを使って，自分以外が場に出したカードを見ることができる
- 引数で受け取ったnumCardsには，現在の各プレーヤの保持枚数が格納されている
- LittleThinkPlayerを参考に
  - memoryに場に出されたカードが溜まる
  - 基本的には「他人の出した」カード
    - 正確には「場で通った」カード
  - （必要であれば）適当な変数を作成して，保持枚数を管理し，戦略に使う

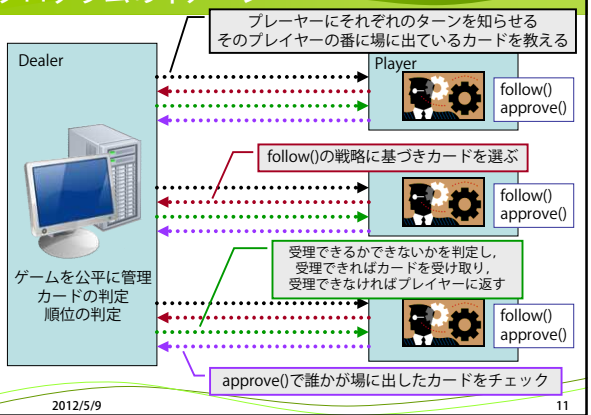
カードの判定(1)

- ゲームとしての判定
  - Dealerクラスが行うので考えなくても良い
  - 判定は Dealer::accept() が行う
    - 受理されれば，場にカードを出して true
    - 受理されなければ，カードは手持ちに戻り false
  - 複数枚(Jkr込)のチェックなども実装されている

カードの判定(2)

- ゲームとしての判定
  - 空のセットや通らない(場よりも弱い)カードが出されるとパス扱い
    - 通らなかったカードは自分の手持ちに戻る
  - 実際には，出すカードが通るかどうかは判定した方がいい
    - 受理されない＝パス

プログラムのイメージ



演習の進め方

- リーダーを中心に計画的に
- 大きな流れ
  - プレイヤー作成→対戦→議論→改良→....
- 例えば，
  - 最初は1人1つずつ作る
  - 対戦させ，問題点を考察する
  - いくつかのプレイヤーを統合して改良
  - 最終的に班で1つの最強プレイヤーを作る

プレイヤーの作り方

- 1回のターンであまり欲張らなくていい
  - 3人寄れば文殊の知恵
  - できるだけ弱いカードから出す
  - 選んだカードが場に通るか考えて出す
  - 複数カードを出せるようにする(かなり強力)
- follow()とapprove()以外を変更しないこと
  - 使用するのはCardとCardSet
  - 新しく関数を作っても良いが，既存のクラスやメンバ変数に影響しないこと

複数プレイヤーの実現

- 継承によりサブクラスを作成する
  - たとえば，LittleThinkPlayer.{h|cc} を参照

```
LittleThinkPlayer::LittleThinkPlayer(const char * s) : Player(s) {}

bool LittleThinkPlayer::approve(CardSet & pile, int numCards[]) {
        memory.insert(pile); // LTPのメンバ変数memoryにカードを格納
        // numCard[]には各プレイヤーの残り枚数が格納されている
}

bool LittleThinkPlayer::follow(CardSet & pile, CardSet & s) {
        Card tmp;
        s.makeempty();
        inHand().pickup(&tmp, -1); // anyway, choose a card.
        // Player の Private 変数には直接アクセスできないので注意
        s.insert(tmp);
        return true;
}

---main関数で
d.regist(new Player("Erika")); // 通常の player.cc の follow や approve
d.regist(new LittleThinkPlayer("Warabi")); // 通常の LittleThinkPlayer.cc の follow や approve
```


演習の進め方

- 一人の力に頼らない
  - みんなで分担すること
  - 必ず全員がプログラムを作成すること
    - それが採用されるかどうかは問わない
- 定義したサブクラスを読み込めば，他の班のプログラムでも動くこと
  - 班同士の対戦をするため
  - ソースファイル，クラス名は
    - 第1班ならば Group1 とすること

レポートについて(1)

- 班で1つ，以下のことを明記すること
  - 表紙には，演習名，班員の学生番号と名前，提出日
  - 各班員の役割分担
    - 例えば，各繰り返しの段階で実装した戦略や作業などを箇条書きする程度でよい
      - ただし，誰が何をやったかはわかるようにすること
        - » 最終的にそれが採用されたかは問わない
  - プログラムリスト
    - ただし，最終的に実装した部分のみでよい
  - 実装した戦略ルーチンの考え方
    - 工夫した点および不完全な点を含む
  - 実行結果
    - 主要な部分・レポートでの説明に必要な部分のみでよい
  - 全体の考察とまとめ

レポートについて(2)

- 個人評価表を提出
  - 自己評価と班員の評価
  - A4用紙で1枚
- 提出の締切
  - 演習最終日の翌週・同一曜日
  - 17時まで
  - 嶋田の部屋まで：E716

評価について

- 班のレポート＋個人評価票＋順位に基づく配点
- 最終プログラムを提出
  - Group[番号].cc および .h の提出
    - LittleThinkPlayer.{cc|h}のような形で
    - デバッグ用のコメントなどは出さないように！
- 最終回の15:30
  - 抽選により3班で1リーグの1次予選
    - ただし，各リーグともThinkTA1とベースのPlayerが参加
    - 順番を入れ替えて5回試行
    - コンパイルできなかったら，即敗退
  - 各リーグ1位が決勝進出
    - ただし，ThinkTA1が1名参加
    - 順番を入れ替えて5回試行
- 順位と得点
  - 1位：5点，2位：4点，3位：3点