知能情報工学実験演習II
C++演習 (Part 2)
グループ演習

下薗，中村
ＴＡ：佐藤 誠 / 古野 雅大

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
- カードゲーム「大富豪（大貧民）」の思考ルーチンを作成し強いコンピュータプレーヤーを作る
- テキスト，このスライド
  - ~sin/DAIHUGOU2014/text2014.pdf
  - ~sin/DAIHUGOU2014/howto2014.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.

- 親（リーダー）は，好きなカード（の組）を場に出せる.
- 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.

- つづく全員がパスをつづけたら，場のカードは流して，最後にカードを出したプレイヤーが新たにリーダーとなる.
- カードがなくなった「あがり」プレイヤーから順位がつく.

- 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

- ~sin/DAIHUGOU2014
  - Card.{h|cc}
  - CardSet.{h|cc}
    基本的には前回までと同じ．専用に拡張しているのでこちらをコピーし使う
  - Dealer.{h|cc}
    ゲームの管理者．コピーし使う
  - Player.{h|cc}
    プレイヤーの基礎になる．コピーし使う
  - LittleThinkPlayer.{h|cc}
    プレイヤーのクラスの拡張用テンプレート．
    コピーし，クラス名を変更し編集して，グループのプレーヤーを作る
  - ThinkTA1.{h|o}
    昔のTAが作成した思考ルーチン．ソースコードはひみつ．
    練習試合の相手に活用
  - main.{h|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
- このクラスはさわらない

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つ以外の関数の中身を作成する.
    - follow() および approve() で使用する変数，関数は自由に作って可

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 のコメントなどを参考に
  - たとえば，変数 memory に場に出されたカードをためる
  - 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）
  - 戦略に使ってよい

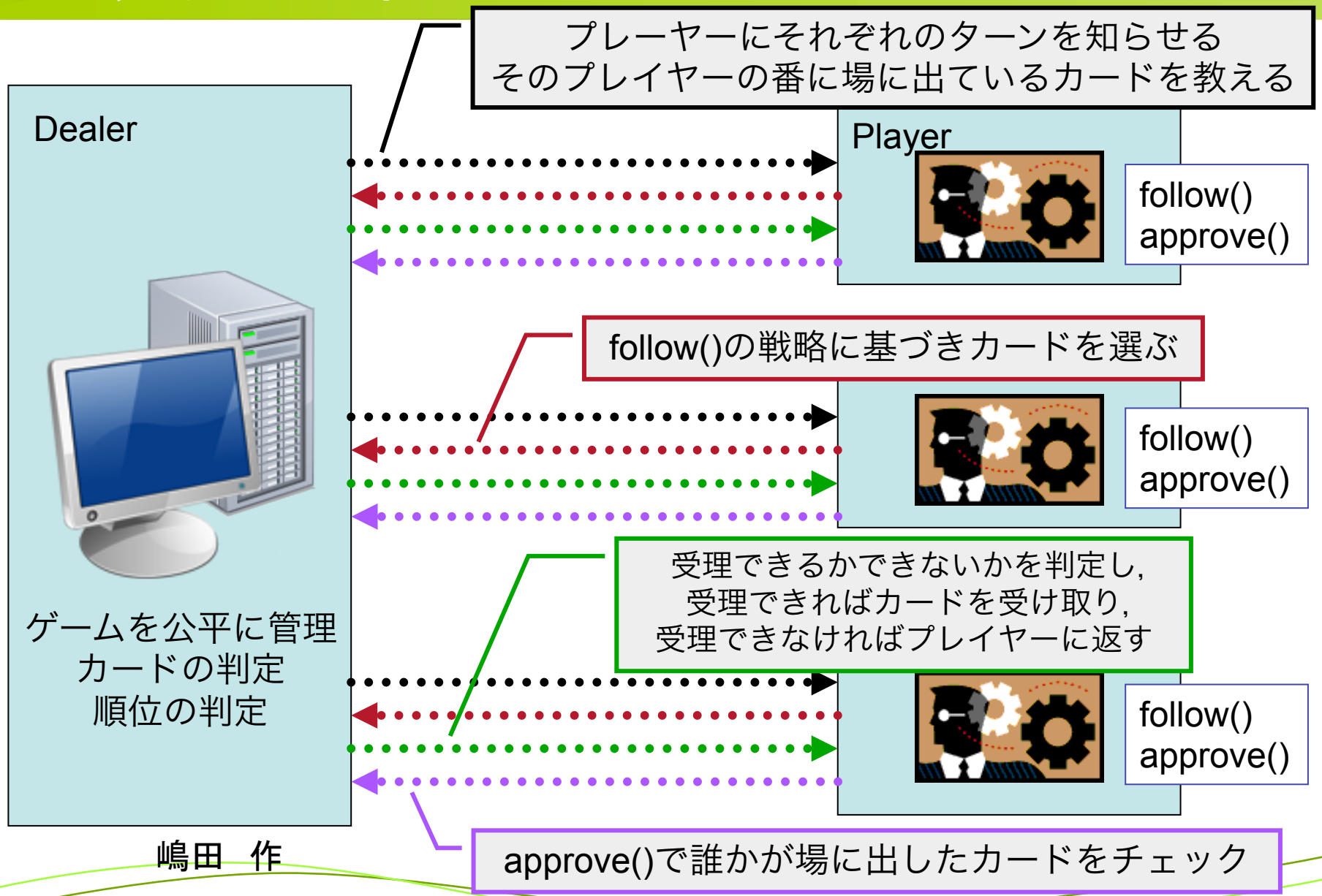
カードの判定(1)

- 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



嶋田　作

演習の進め方

- リーダーを中心に計画的に行う
  - 時間，スケジュールが大事
- 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

- 欲張らず，こつこつ進める
  - 3人寄れば文殊の知恵
  - できるだけ弱いカードから出す，等
  - 選んだカードが場に通るか考える
  - 複数カードを出せるようにする(かなり強力)
- follow() と approve() の中身以外，既存の関数を変更しないこと
  - Card, CardSet は public な関数をうまく使う.
  - 最後にグループ同士の対戦プログラムをコンパイルできればよい

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
  - 作業をみんなで分担すること
  - 全員がプログラムの作成にたずさわること
    - それが採用されるかどうかは問わない
- 定義したプレーヤーのクラスが他の班のプログラムとともに動作すること
  - 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)

- グループレポート：
  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
  - 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
  - 実装したアイディア，それを採用した根拠など
    - 工夫した点，できた部分，実際どうだったか，できなかった部分　など
  - 実行結果
    - 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：
  - 自己評価と班員の評価
  - A4，1枚の様式を後日配付

- 提出の締切（どちらも）
  - 演習最終日の次の本科目実施日の前日，18 時まで
  - 下薗教員室の斜め前の部屋：**E608（16:20~）**

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（~sin/DAIFUGOU2014/Groups/）
    - LittleThinkPlayer.{cc|h}のような形で
    - デバッグ用のコメントなどは，本番ではオフに！

- 最終回の後半
  - 抽選により4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
    - 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10回試行

- 順位と得点
  - 順位がランキングポイントとなる配点.
    ただしタイの場合１位の回数で優先.