PC教室での使用を想定した授業アンケートシステムの研究開発

ー 授業コミュニケーションの向上への取り組み ー

東京工芸大学工学部コンピュータ応用学科

宇田川 佳久、土井　勇介、齋藤　幸弘

目　次

1. 研究の背景と概要
2. システム構成
3. 実装の概要
4. 実験結果
5. まとめと今後の研究方針

1. 研究の背景と概要

- 情報通信技術（ICT）を活用した授業アンケートが、授業の改善に役立つことから、多くの教育機関で推進されている。
- ICTツールとしてクリッカーと呼ばれる製品が，導入されている。

1. 研究の背景と概要

- クリッカーの利点
  - アンケートの集計がリアルタイム
    ⇒ 授業の進め方を修正できる
  - 匿名性 ⇒ 積極的に授業に参加できる。

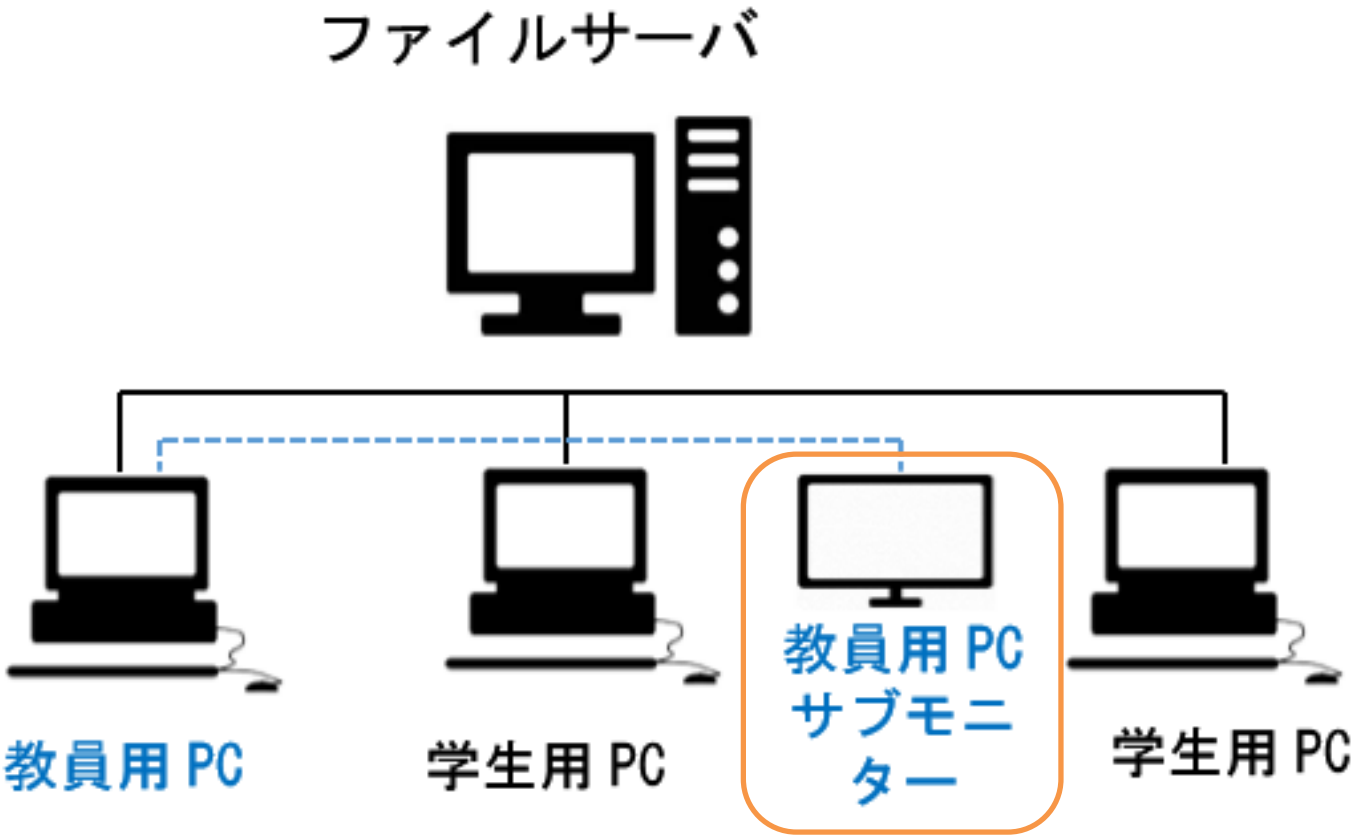
- クリッカーの問題点
  - リモコンの配布と回収に時間がかかる。
  - 高価である。
  - Webアプリとして作った場合、ブラウザの起動とURLの入力が必須。
  - ボタンでしか回答できない（文章は使えない）。

研究の方針

1. PC教室内での利用を前提とし，ソケット通信機能を使うアプリケーションとする。
   ⇒ これにより，Webブラウザを使う必要がなく，URLの入力も不要（即座に起動する）。
2. 学生（クライアント側）では，回答番号とメッセージを送信する機能も提供する。
3. 教員（サーバー側）では，回答番号の集計・表示を行う。

PC演習室の構成

教員用PCのサブモニターにアンケートを表示する
（すなわち、通常の授業の資料として表示する）。



2. システム構成

- 起動が約1秒
- Webサーバが不要

学生1

学生2

…

学生N

アンケートサーバ

(1) クライアント接続とパレメータ送信機能

・学生からの接続要求があるたびに、各学生用のスレッドを起動する

・クライアントの動作を制御するパラメータを接続中のクライアントに送信する。

学生1用スレッド

学生2用スレッド

…

学生N用スレッド

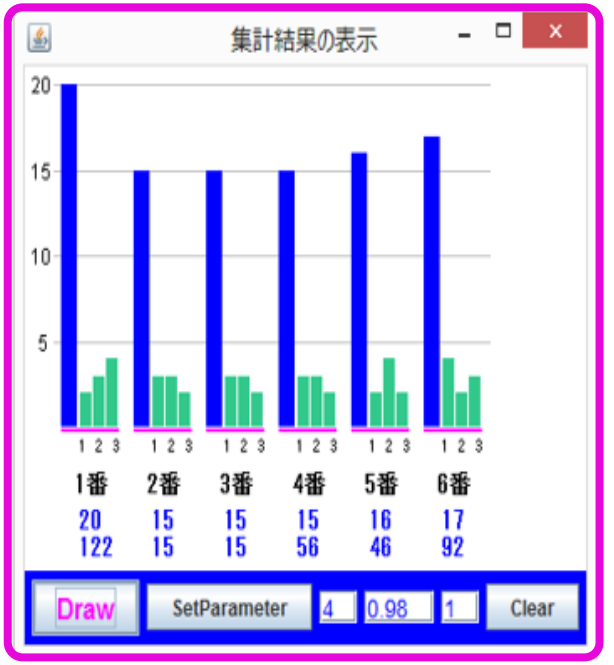
接続中クライアントの管理機能

取得データ

データ集計・グラフ表示機能

(2)データ集計・グラフ表示機能の強化

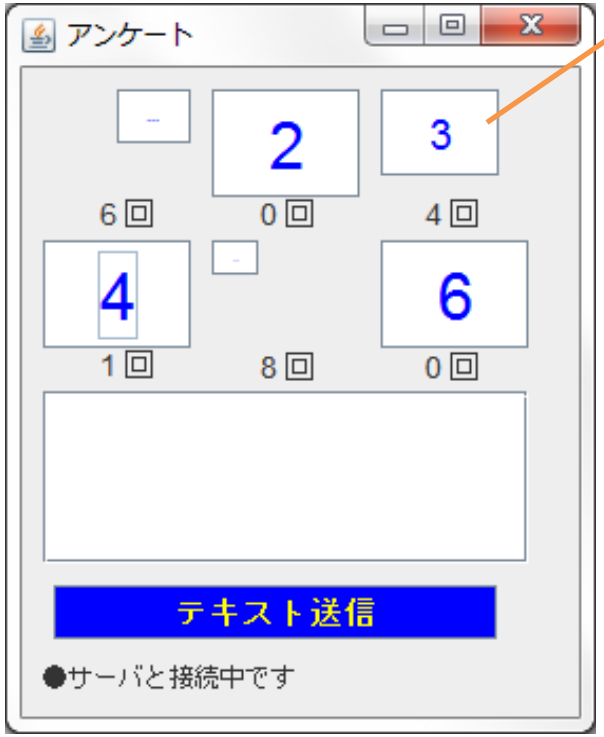
その時点で取得したデータを、理解度別に集計し、サーバ上にグラフで表示する。



① クライアント画面の仕様と実装

前年度のクライアント実装の問題点の解消

- クライアントでの過剰クリック
  - 回答ボタンのクリック回数が表示されない
  - 過剰クリックの回避機能がない



過剰クリック時の動作、表示については、サーバから指示可能

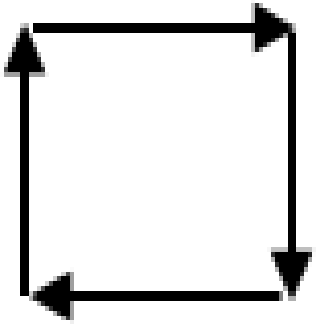
実装のポイント

```
 1 public void DrawClientButton( … , int numB) {
 2 …
 3   // 回答ボタンのクリック回数をカウントする.
 4   bCount[numB]= bCount[numB] + 1;
 5   if (bCount[numB] >= nStart){ // 剰クリック対応
 6     wButt[numB]= (int)( wButt[numB]*nScale);
 7     hButt[numB]= (int)( hButt[numB]*nScale);
 8     sFont[numB]= (int)( sFont[numB]*nScale*nScale);
 9     wMax= (float) (width - wButt[numB]);
10     hMax= (float) (hight - hButt[numB]);}
```

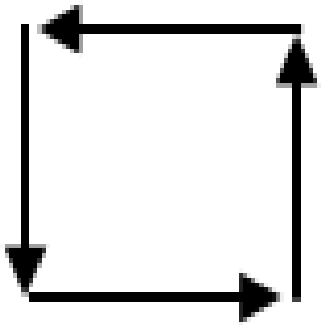
ランダムの場合

```
 1  if( nPattern == 1){
 2    wdist= (int)(Math.random() * wMax);
 3    hdist= (int)(Math.random() * hMax);
 4  }
```

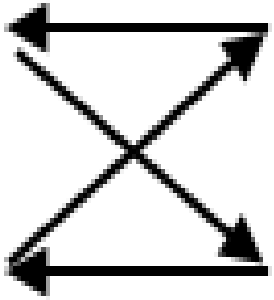
表示パターンの例



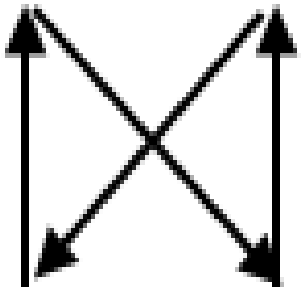
(1) 表示パターン2



(2) 表示パターン3

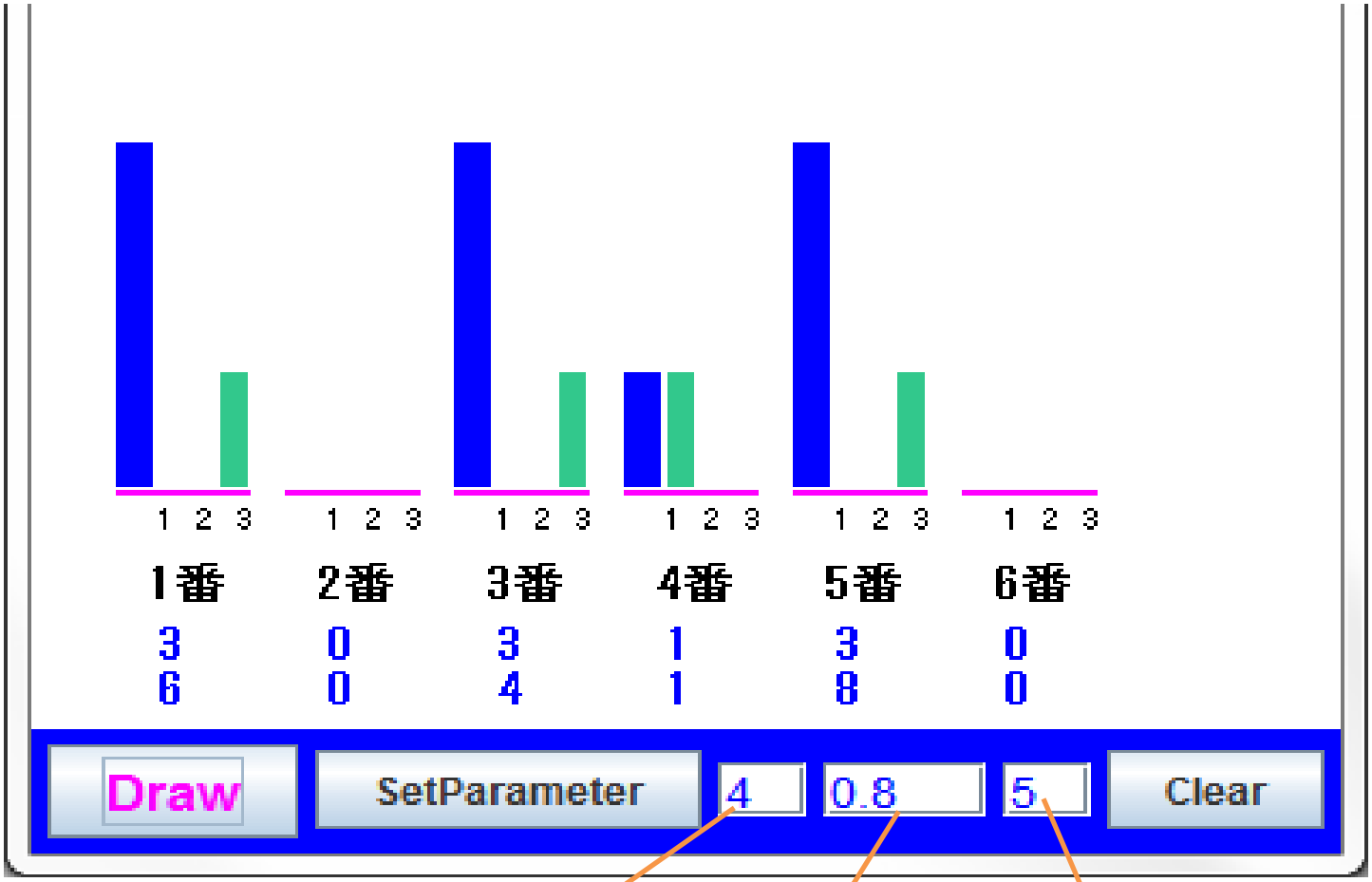


(3) 表示パターン4



(4) 表示パターン5

② サーバ機能の仕様と実装

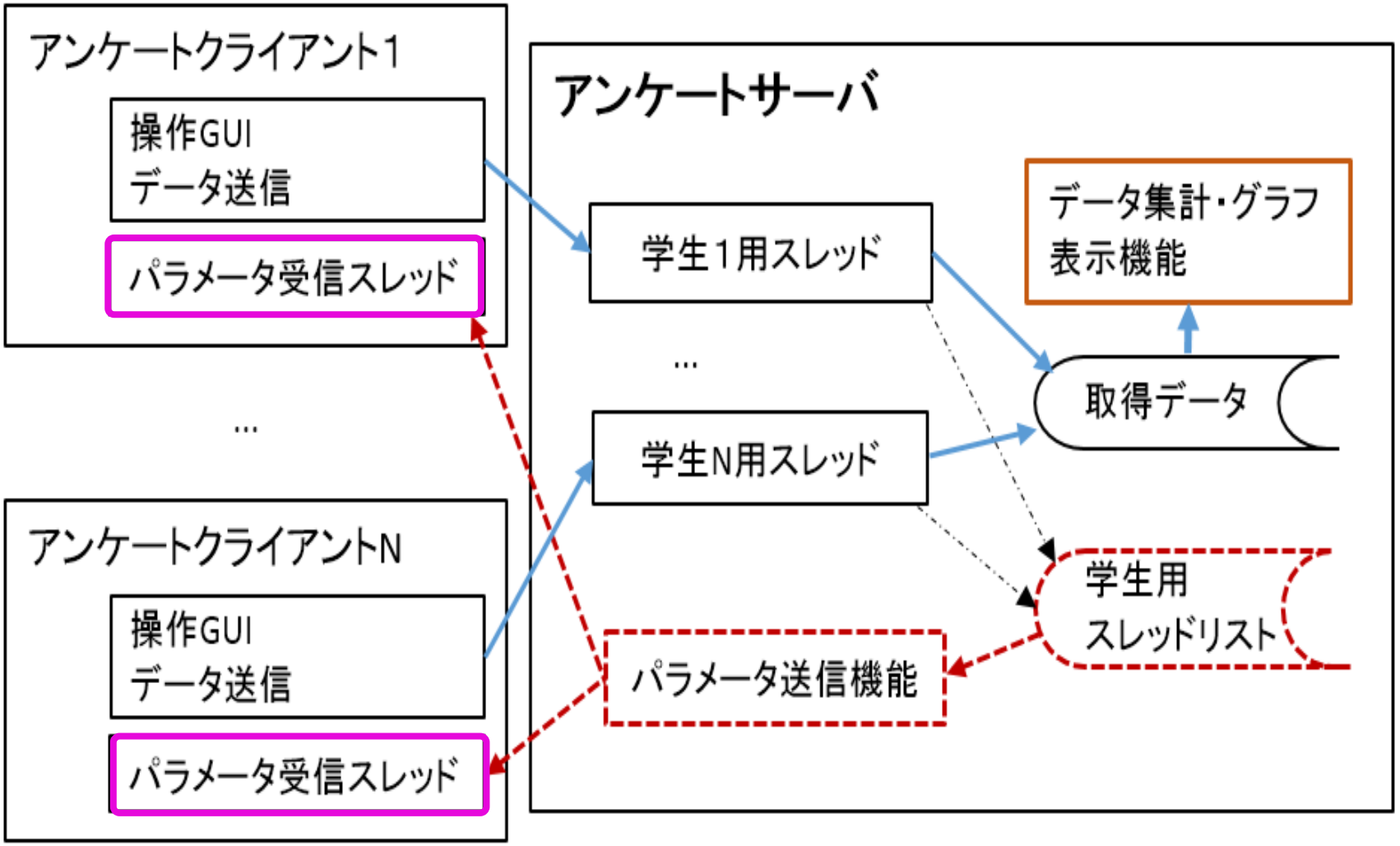


回避処理の開始回数

図形の表示比率

図形の表示パターン

サーバおよびクライアントでのスレッドの構造



データ集計スレッドの主要な処理

```
class Click_ServerThread extends Thread {
    Socket conn; // クライアントに対応するソケット
    public Click_ServerThread(Socket s) {
        super();
        conn = s;
    }
    public void run() { // スレッドで実行される内容
        try {
            BufferedReader in =
                new BufferedReader(
                    new InputStreamReader(conn.getInputStream()));
        while(true) {
            try {
                String s = in.readLine(); //一行入力
                  <クライアントからのメッセージに対応する処理>
            }
        }
     }
}
```

スレッドとして実行される処理

集計処理の実装

変数 s = “クライアントID　＆　回答番号”

CkAll, Ck のデータ型 : HashMap <String, Integer>

送信した 総数のカウント

```
if ( ! CkAll.containsKey(s)){
   CkAll.put(s, 1);// 初めて登録
   pollall[id]= pollall[id] + 1
} else {
   xx= CkAll.get(s);
   xx= xx + 1;
   CkAll.put(s, xx);
   pollall[id]= pollall[id] + 1;
}
```

有効回答数のカウント

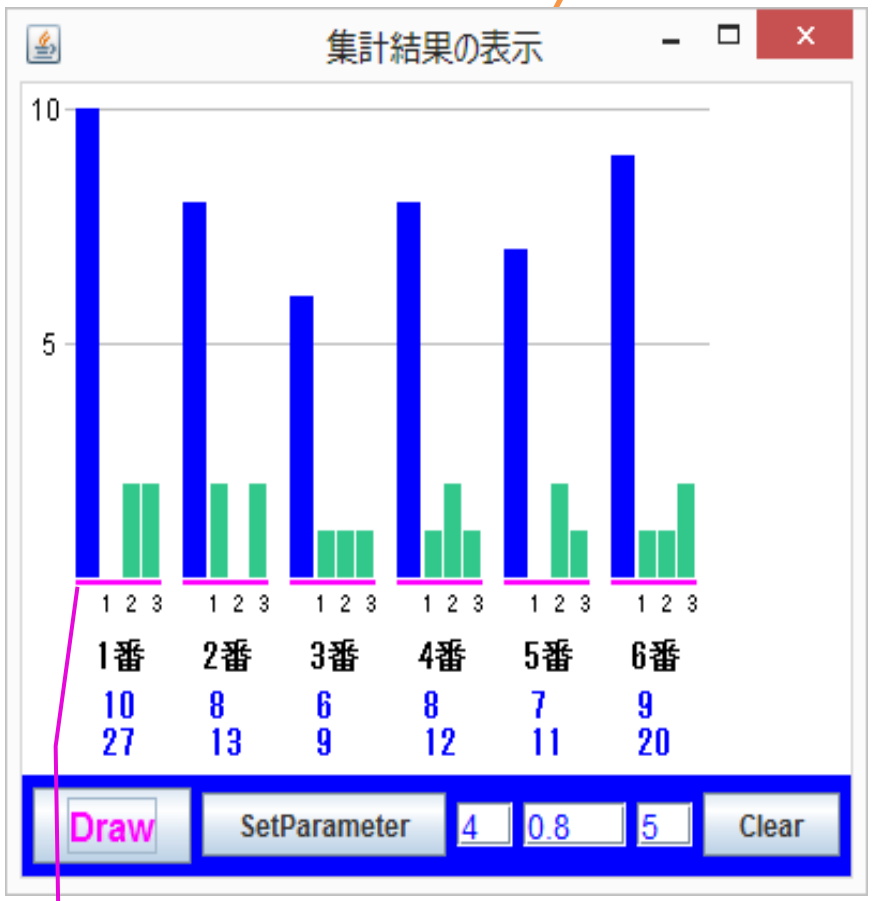
```
if ( ! Ck.containsKey(s)){
   Ck.put(s, 1);//  初めて登録
   poll[id]= poll[id] + 1
} else {
   xx= Ck.get(s);
   if ( xx < MaxVote ){
      xx= xx + 1;
      Ck.put(s, xx);
      poll[id]= poll[id] + 1;
   }
}
```

回答ボタンのクリック回数を計算する処理

```
1  Iterator it = Ck.keySet().iterator();
2  while (it.hasNext()) {
3     Object o = it.next();
4     s2= (String) o;
5     v= Ck.get(o);      //  クリック回数
6     s1= s2.trim();     //  ボタンID＋学生ID
7     s3= s1.substring(1,2);
8     k= Integer.parseInt(s3); //ボタンIDの番号
9     if (v==1){
10          pollCN1[k]= pollCN1[k]+ 1;
11    } else if (v==2){
12          pollCN2[k]= pollCN2[k]+ 1;
13    } else if (v==3){
14          pollCN3[k]= pollCN3[k]+ 1;
15    }
16  }
```

グラフ表示機能の実装

- 回答件数に応じてスケールを変更する。
- 棒グラフで表示する。



ここの y座標が 200pixel

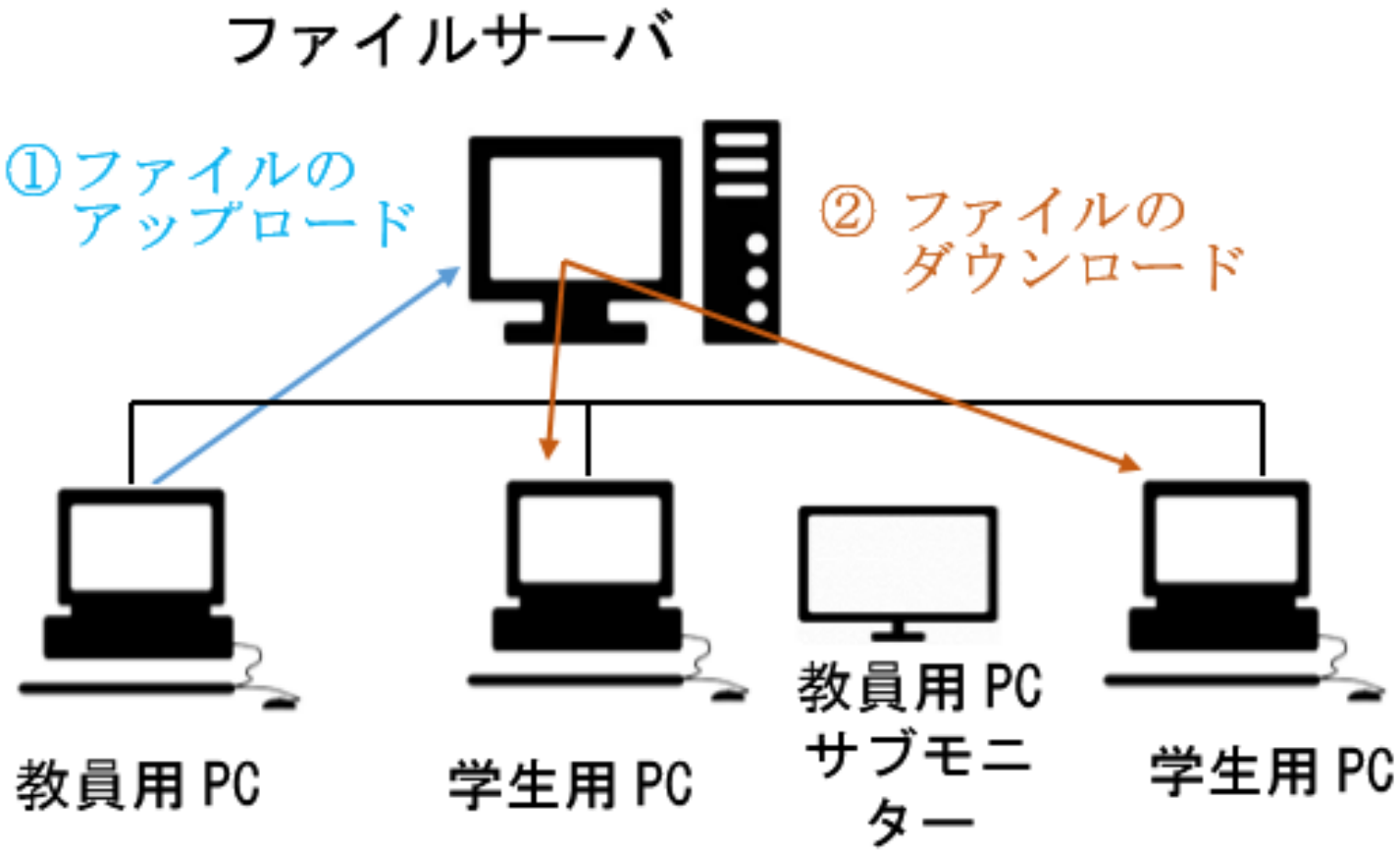
```
int maxP= 0;
  float ratio= 0.0F;
 String stp;

for (int k=1; k<=6; k++){
    if( poll[k] > maxP){
        maxP= poll[k];
    }
}
if( maxP < 5) { maxP= 5; }
ratio= 190.0F / maxP;
for (int k=1; k<=6; k++){
    xb= 50*(k-1);
    y= (int) (poll[k] * ratio);
    myg.setColor(Color.blue);
    myg.fillRect(20+xb, 200-y, 40, y);
}
```

グラフが画面からはみ出ない様にする

4．実験結果

1．クライアントモジュールの配布

クライアントモジュールはファイルサーバー経由で配布した。



実験結果

① 教員は授業アンケートを教員モニターに表示する。

② その授業アンケートに対して、学生はクライアント機能を使って3段階で回答する

良く分かった3回； 大体分かった2回； 少し分かった1回

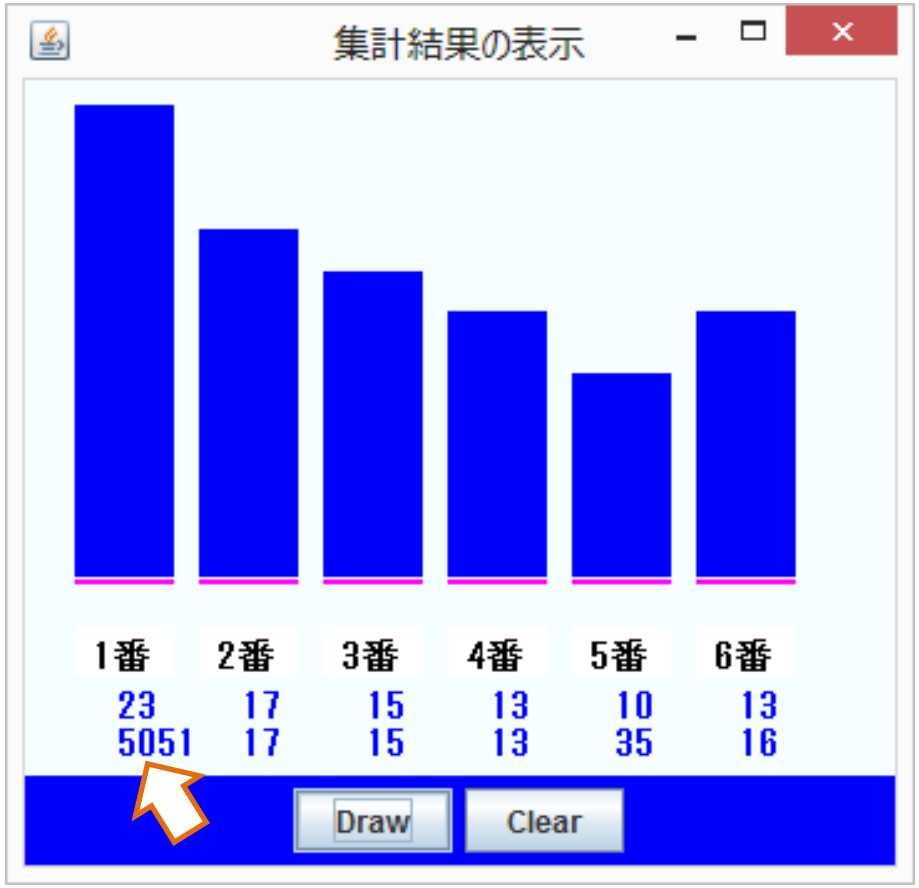
アンケートの例

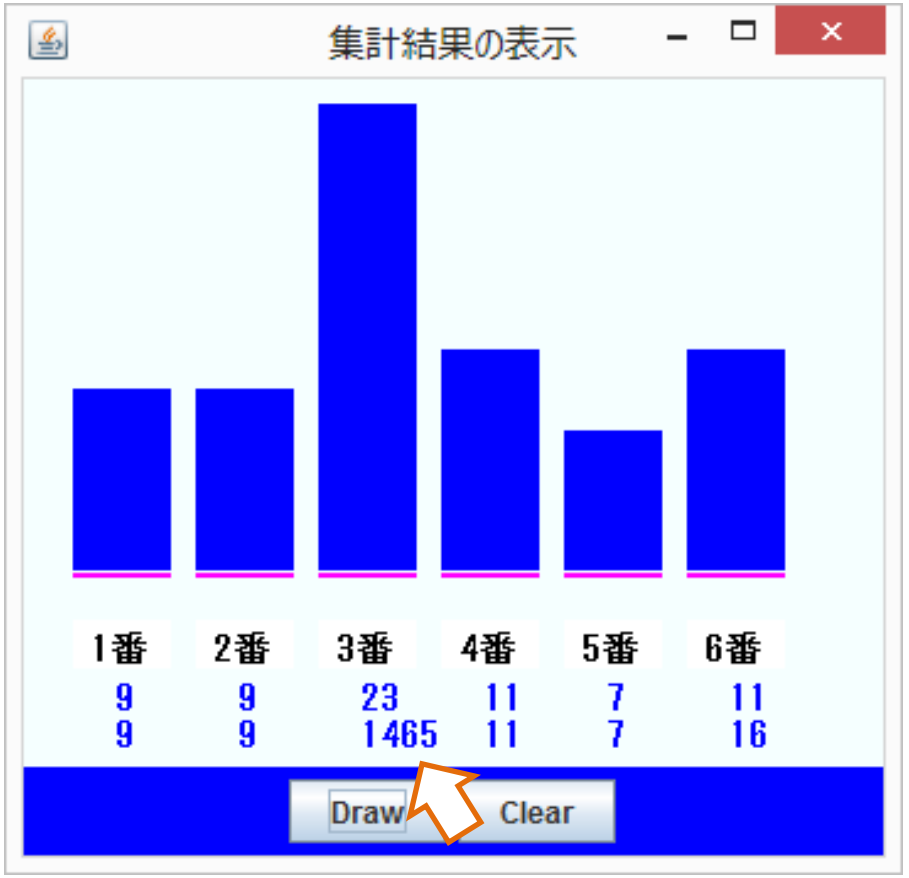
今回の授業で、理解できた項目を選択してください（複数回答可能）。選択肢に無いものは、テキストとして送信してください。

1. 内部結合とは
2. 2つの表の内部結合
3. 表の別名を使った内部結合
4. 3つの表の内部結合
5. 3つの表の内部結合（別の書き方）
6. 内部結合と絞り込み、行の並べ替え

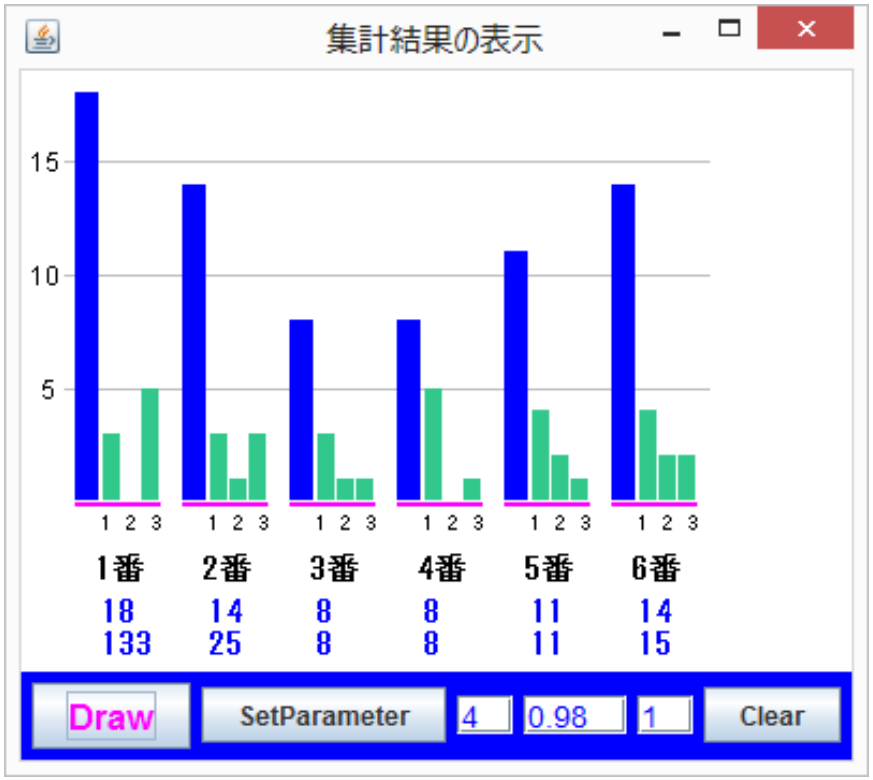
クリック1回：（今までより）少し分かるようになった
クリック2回：講義の内容は大体理解できた
クリック3回：講義の内容はほぼ完璧に理解できた

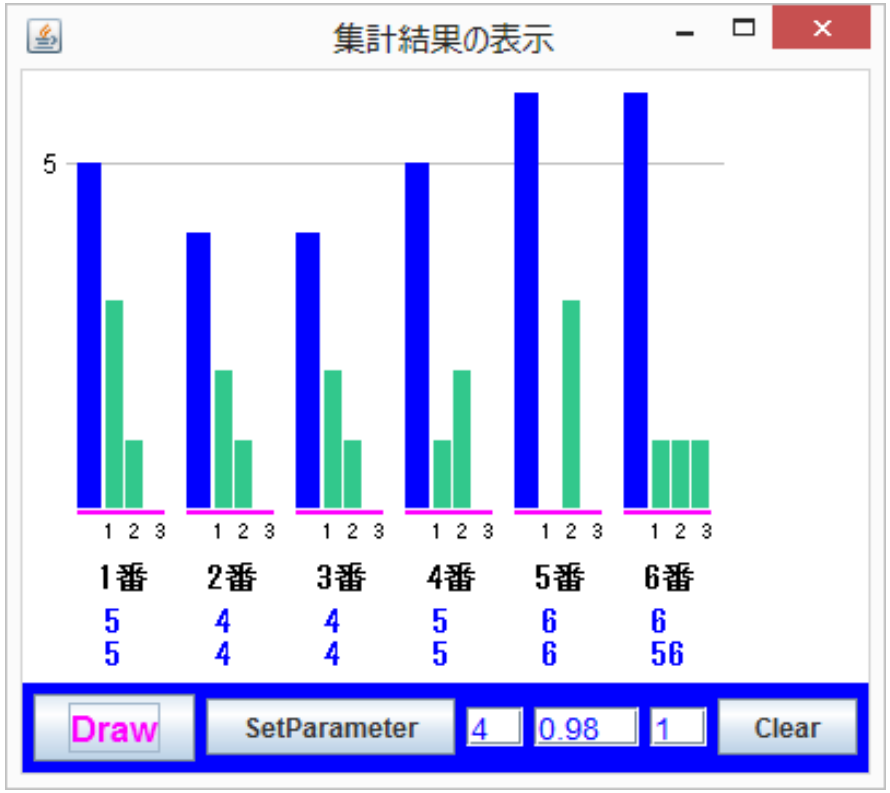
前年度のアンケートの例





今年度のアンケートの例





5. まとめと今後の研究方針

① 当初の目論見どおり過剰クリックを抑制することが出来た。

② 学生の授業に対する理解の程度をリアルタイムに把握できた。

③ 授業の理解度を、教員と学生で共有することができた。
⇒「できないのが当たり前」といった主観的な考え方を排除し、客観的なデータにもとづいた学習への興味を高めることができた。

5. まとめと今後の研究方針

以下の機能開発が残されている。

①クライアントでのクリックの取り消し機能

②文章での回答を分析し画面に表示する機能

③ログデータの分析機能