THE UNIVERSITY of EDINBURGH
informatics

Adaptive Comparative Judgement (ACJ)

Paul Anderson

`<dcspaul@ed.ac.uk>`

`http://homepages.inf.ed.ac.uk/dcspaul`

IJP/IPPO

Introduction to Java Programming

- MSc course
- practical programming
  - being renamed as “Introduction to Practical Programming with Objects”
- no exams
  - two extended practical assignments
- 85 students last year
  - very varied experienced (no programming, to professionals)
  - including distance students
- no lectures
  - work guided by assignments
  - lab sessions
  - Piazza
  - some online videos
  - some online notes

Code readability

“unreadable” code is useless - no matter how well it appears to work

- readability is not easy (impossible?) to assess with an “auto-marker”
- “analytic” grading does not work well (Sadler 2009)
  - e.g. summing criteria such as variable names, layout, comments, etc.
- a large variation between different markers would not be unusual (Bloxham 2016)

```
class Div{static $1 $_;class $1{void _$(String $_){System.//
out.print($_);}$1(){_();}void _(){int _,$,_$,$$,__,a=(1<<5),
b=100,c=12,bc=a*c;b-=1<<4;while(bc>0){for($$=_$=_=__=$=(int)
b;_$>($$-(1<<2));_$(""+(char)(_$)),_$-=1<<1)for(_=$=__-9;_>(
$-6);_-=1<<1,_$(""+(char)(_$!=b?_:a)));char S$=(char)(b+c+1)
;_$("te"+S$+(char)(c+(int)S$));bc--;}}}Div(){$_=new $1();}
public static void main(String []$){Div b=new Div();}}
```

Comparative judgement

Comparative judgement (CJ)

- pairwise comparisons of “scripts”
- simple binary judgement of which is “best”
  - comparisons can be made very quickly (eg. 1-2 minutes)
- algorithm to generate ranking from comparisons
- marking of “reference” scripts to establish absolute grades
  - avoids the problem of multiple markers with different absolute standards

Adaptive comparative judgement (ACJ)

- selects pairs so as to improve convergence
  - interesting algorithms
- good evidence that this works well in some situations
  - particularly for large numbers (e.g exam boards)

Peer judgements

"*An alternative is to extend holistic appraisal to a context in which students themselves engage in making multiple holistic judgements of complex works, the source material being the work of their peers …*" (Sadler 2009)

Having students make the comparisons themselves ...

- encourages them to read other peoples code and see different approaches to the problem
- helps them to understand the difficulties of marking and what is expected (assessment literacy)

It also ...

- has the potential to create a ranking as an aid to marking
- and to provide an insight into what the students themselves consider to be readable
  - which by definition might be considered "readable code" ?
- scales well to larger courses

A couple of projects in the University have attempted this ...

- E.g. Vets & Physics (Hardy 2016)

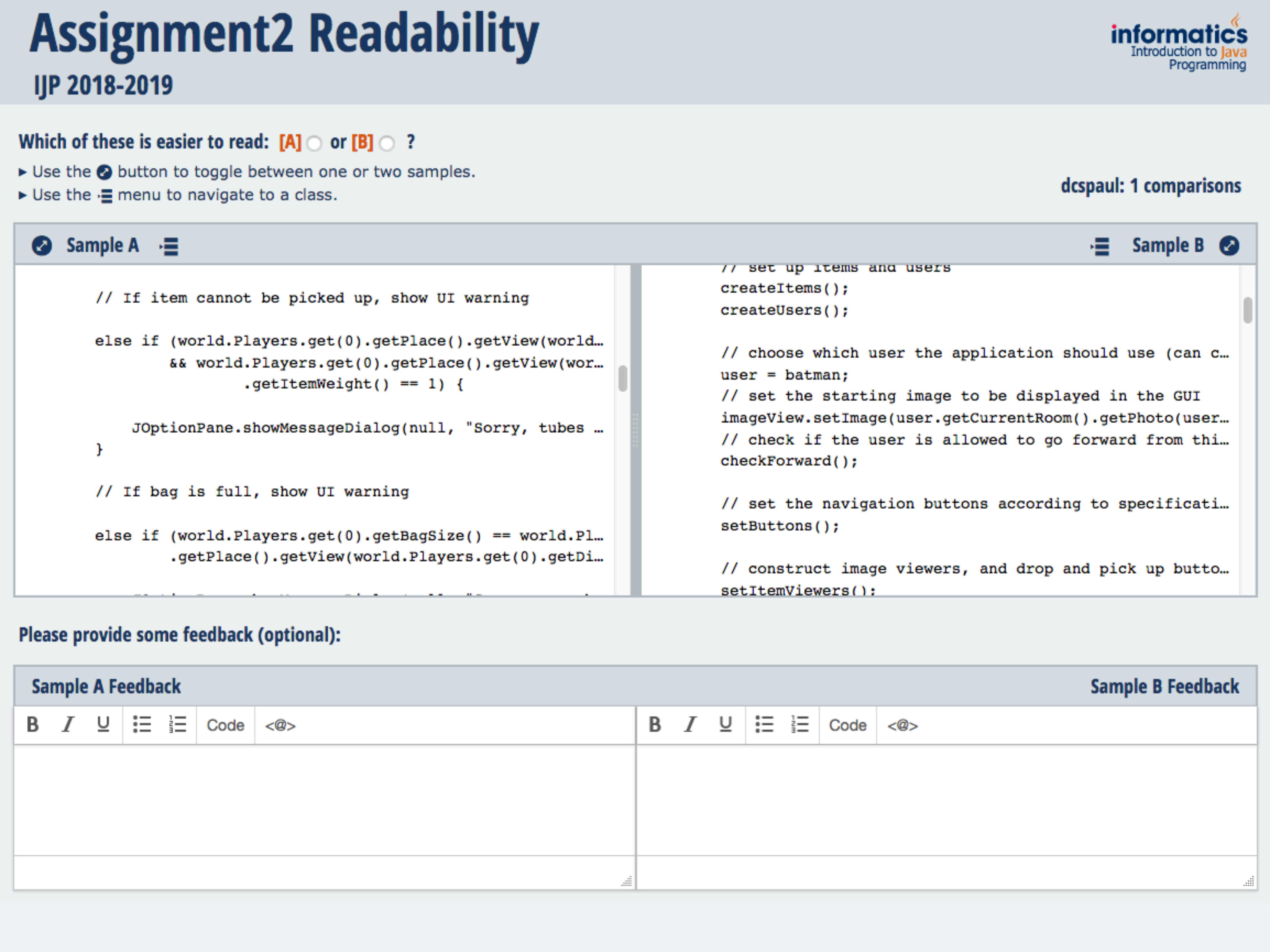
IJP 2017

We attempted this initially with IJP and code readability

- after submission of the first assignment …
  - we asked the students to view pairs of submissions (from other students) and say which of the two samples they found easier to follow
- this was not “marked"
  - but we said that it was necessary for the student to participate in order to obtain one of the higher grades
- in addition to the results, we collected peer comments to feedback to the authors

Results

- some good feedback from the students
- but we made some mistakes -
  - showing only one file
  - software errors
- no meaningful ranking!



Assignment2 Readability

IJP 2018-2019

Which of these is easier to read: [A] ○ or [B] ○ ?

- Use the button to toggle between one or two samples.
- Use the menu to navigate to a class.

dcspaul: 1 comparisons

Sample A

```
        // If item cannot be picked up, show UI warning

        else if (world.Players.get(0).getPlace().getView(world…
                && world.Players.get(0).getPlace().getView(wor…
                        .getItemWeight() == 1) {

            JOptionPane.showMessageDialog(null, "Sorry, tubes …
        }

        // If bag is full, show UI warning

        else if (world.Players.get(0).getBagSize() == world.Pl…
                .getPlace().getView(world.Players.get(0).getDi…
```

Sample B

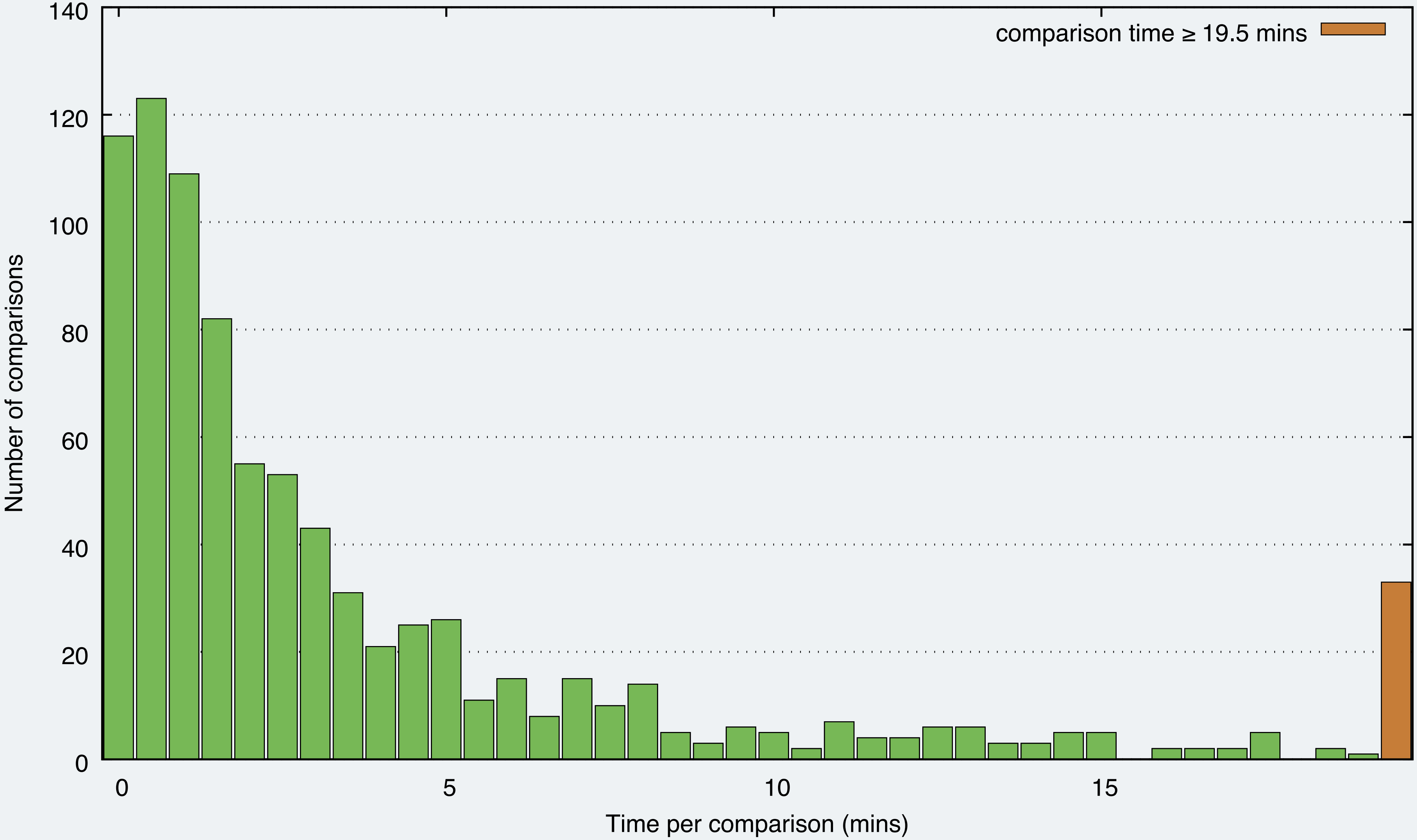
```
// set up items and users
createItems();
createUsers();

// choose which user the application should use (can c…
user = batman;
// set the starting image to be displayed in the GUI
imageView.setImage(user.getCurrentRoom().getPhoto(user…
// check if the user is allowed to go forward from thi…
checkForward();

// set the navigation buttons according to specificati…
setButtons();

// construct image viewers, and drop and pick up butto…
setItemViewers();
```

Please provide some feedback (optional):

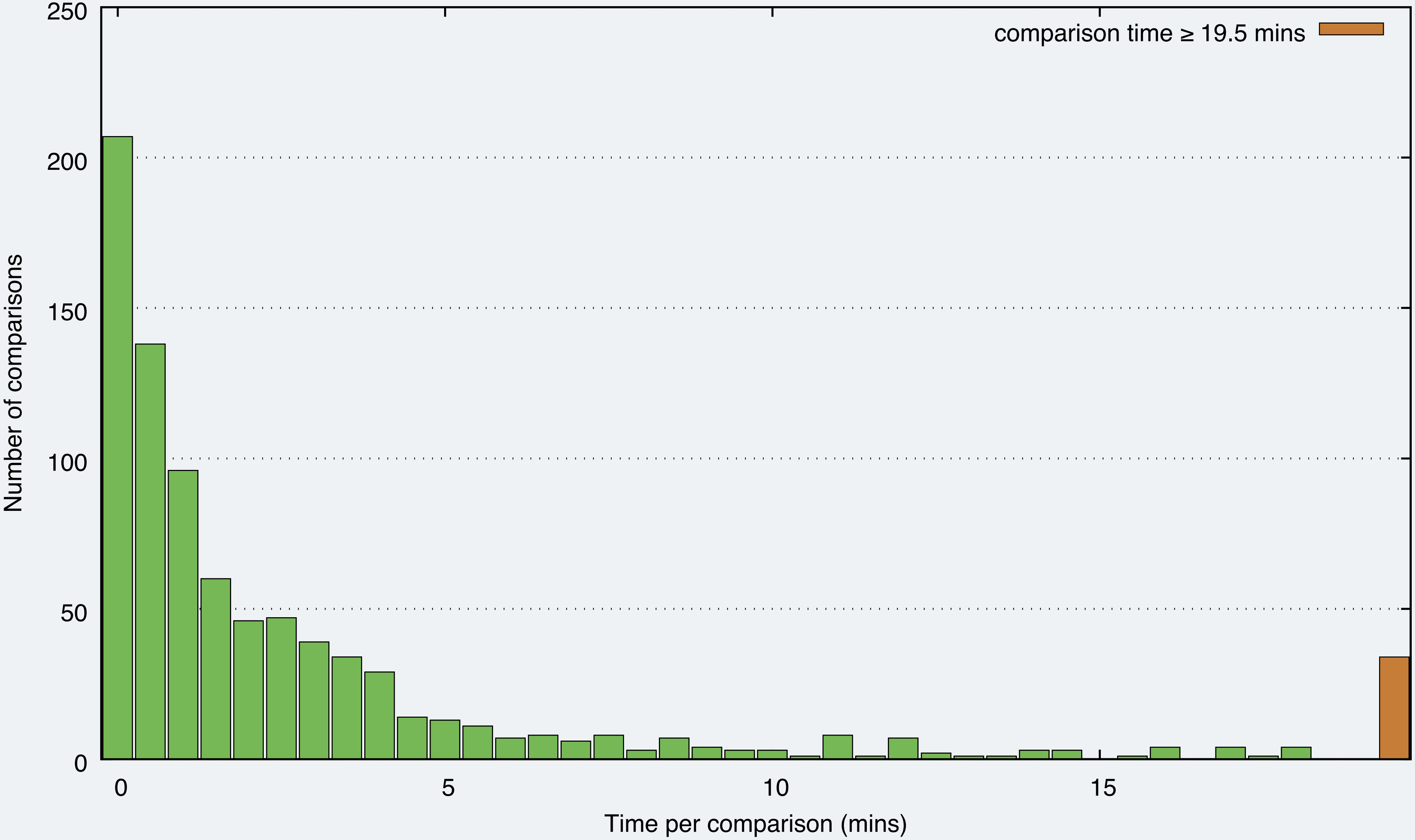
Sample A Feedback

Sample B Feedback

Readability comparisons: time spent



Readability comparisons: time spent



IJP 2018

We repeated the exercise using draft program designs

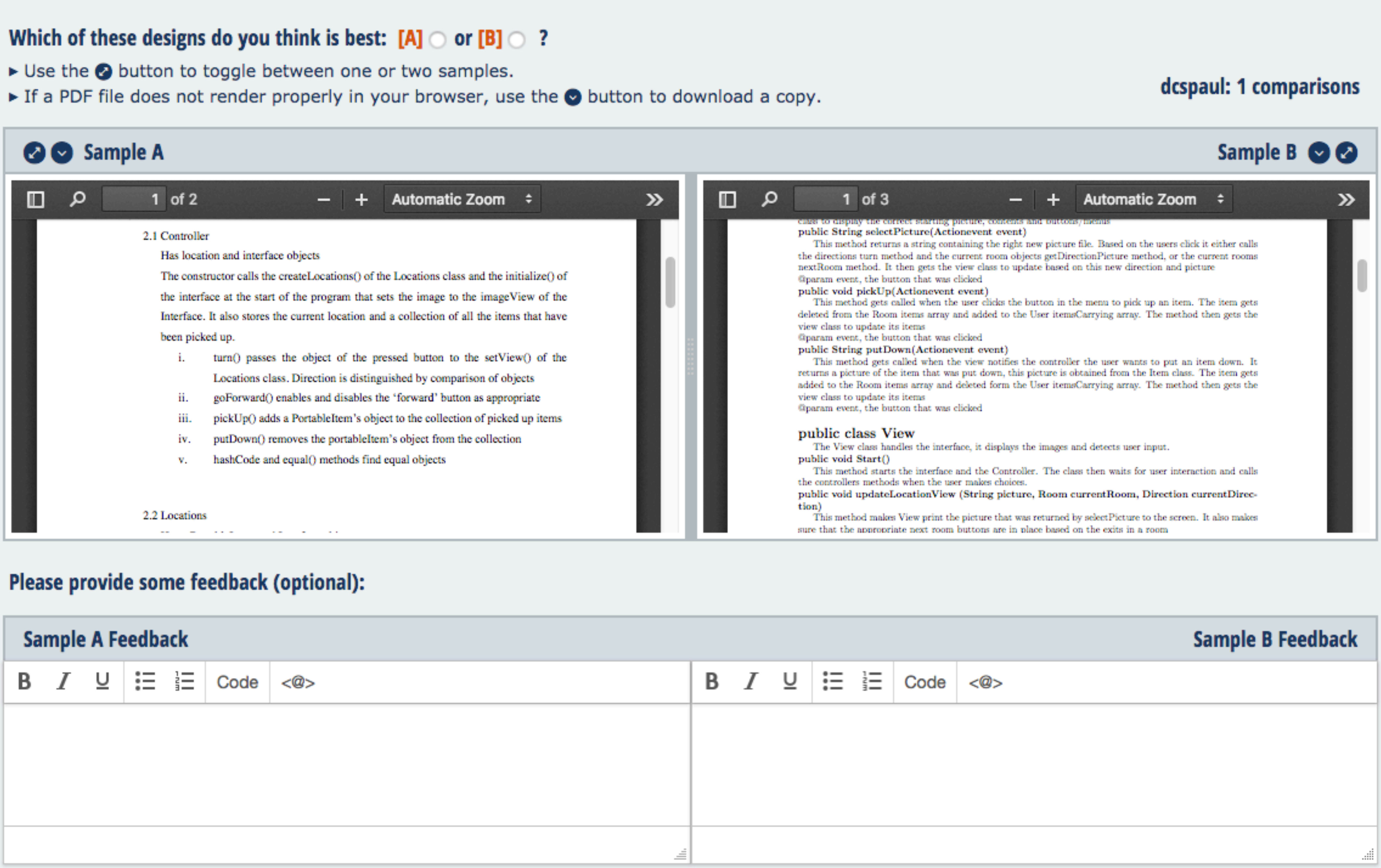
- we specifically wanted students to see different design approaches before finalising their own implementation
- we hoped that this would encourage them to look at the ACJ comparisons without the need for artificial incentives (marks)
- we used newer software with better testing and good logging/auditing
  - but we were still not confident about the ranking algorithm

Results

- 75% of the students engaged
- one student made 35 judgements! (almost all the scripts)
  - which might have skewed things …
- students made some helpful comments (but not a huge number)
- the generated ranking was still poor (see later)
- some students improved their designs
  - but some got worse!!!

Assignment2 Design

IJP 2018-2019



Which of these designs do you think is best: [A] ○ or [B] ○ ?

- ▸ Use the button to toggle between one or two samples.
- ▸ If a PDF file does not render properly in your browser, use the button to download a copy.

dcspaul: 1 comparisons

2.1 Controller

Has location and interface objects

The constructor calls the createLocations() of the Locations class and the initialize() of the interface at the start of the program that sets the image to the imageView of the Interface. It also stores the current location and a collection of all the items that have been picked up.

i. turn() passes the object of the pressed button to the setView() of the Locations class. Direction is distinguished by comparison of objects

ii. goForward() enables and disables the 'forward' button as appropriate

iii. pickUp() adds a PortableItem's object to the collection of picked up items

iv. putDown() removes the portableItem's object from the collection

v. hashCode and equal() methods find equal objects

2.2 Locations

class to display the correct starting picture, contents and buttons/menus

public String selectPicture(Actionevent event)

This method returns a string containing the right new picture file. Based on the users click it either calls the directions turn method and the current room objects getDirectionPicture method, or the current rooms nextRoom method. It then gets the view class to update based on this new direction and picture
@param event, the button that was clicked

public void pickUp(Actionevent event)

This method gets called when the user clicks the button in the menu to pick up an item. The item gets deleted from the Room items array and added to the User itemsCarrying array. The method then gets the view class to update its items
@param event, the button that was clicked

public String putDown(Actionevent event)

This method gets called when the view notifies the controller the user wants to put an item down. It returns a picture of the item that was put down, this picture is obtained from the Item class. The item gets added to the Room items array and deleted form the User itemsCarrying array. The method then gets the view class to update its items
@param event, the button that was clicked

public class View

The View class handles the interface, it displays the images and detects user input.

public void Start()

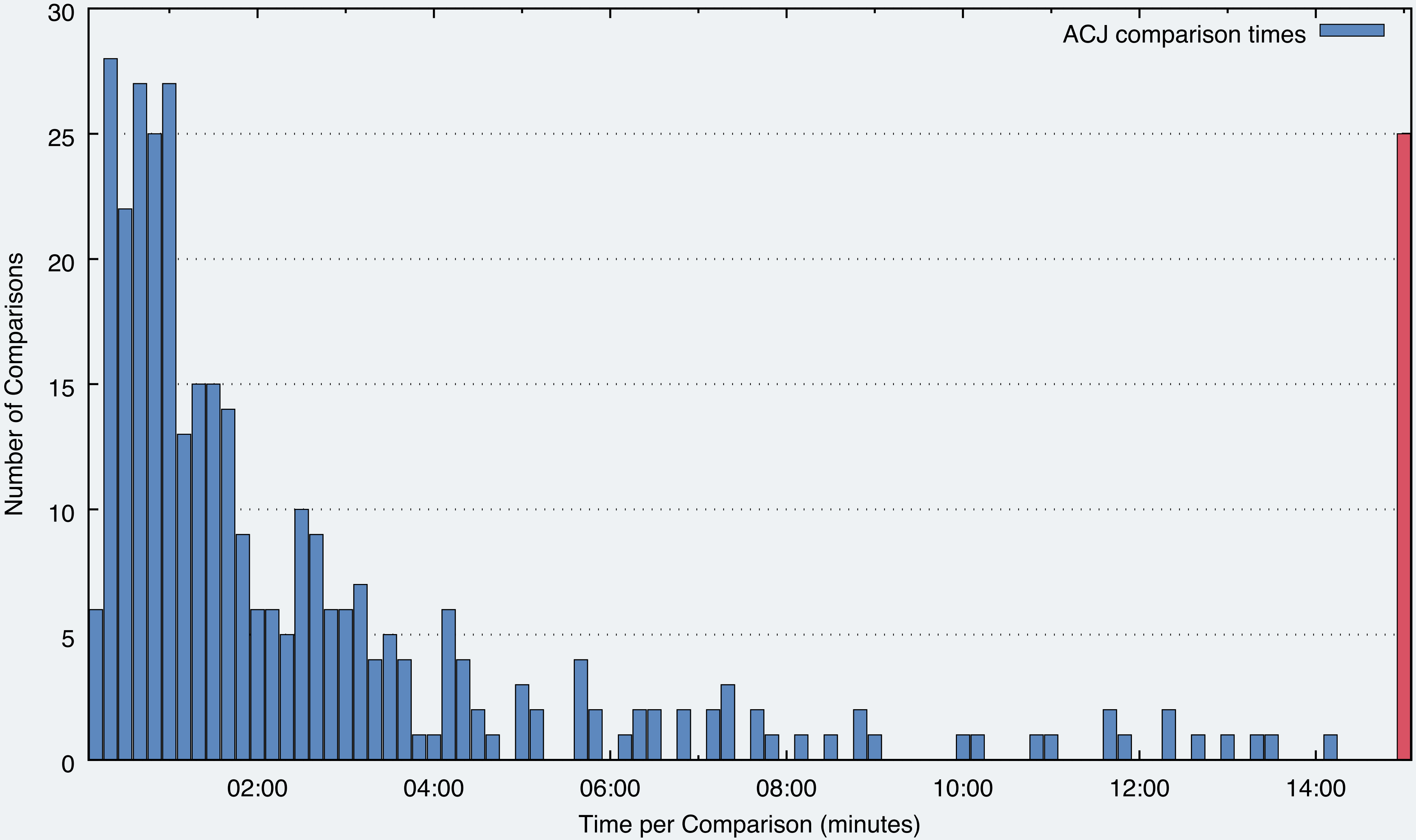
This method starts the interface and the Controller. The class then waits for user interaction and calls the controllers methods when the user makes choices.

public void updateLocationView (String picture, Room currentRoom, Direction currentDirection)

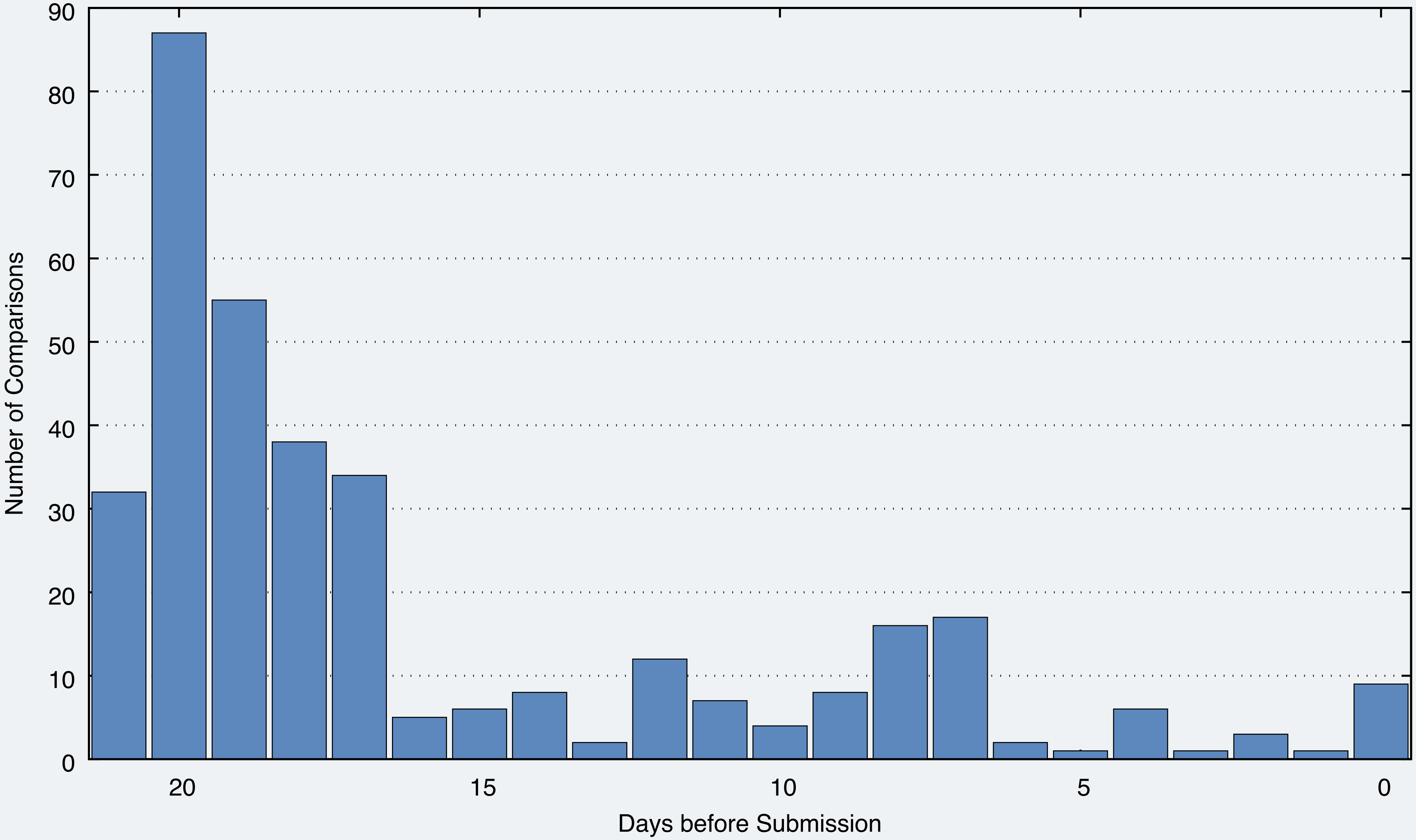
This method makes View print the picture that was returned by selectPicture to the screen. It also makes sure that the appropriate next room buttons are in place based on the exits in a room

Please provide some feedback (optional):

Design comparisons: time spent



Design comparisons



Student comments

- *“I particularly liked your Items methods, and I will see how I could implement them in my design to make it more versatile. Thank you!”*

- *“I really like how you thought about Location. You employed inheritance and abstract methods in your design which make it easier to expand it."*

- *“Overall, I like the design, and I think some of the work is cleverly divided between classes (I actually adopted the idea of having a separate JSONread object from this design). Well done!”*

- *“Some of the classes seem to have some overlapping functions, such as Direction and Location - both contain directions, they just store different things."*

- *“I am not experienced in programming, so this seems a bit chaotic to me. All I see are huge tables listing functions and I have to go back and forth, switching between the diagram and the tables to understand what is happening. Your interface also seems quite complicated to me.”*

- *“This has completely re-defined the way I think about life. However, some of the methods in WorldView and Controller essentially carry out the same action, could they be condensed into one?”*

Ranking

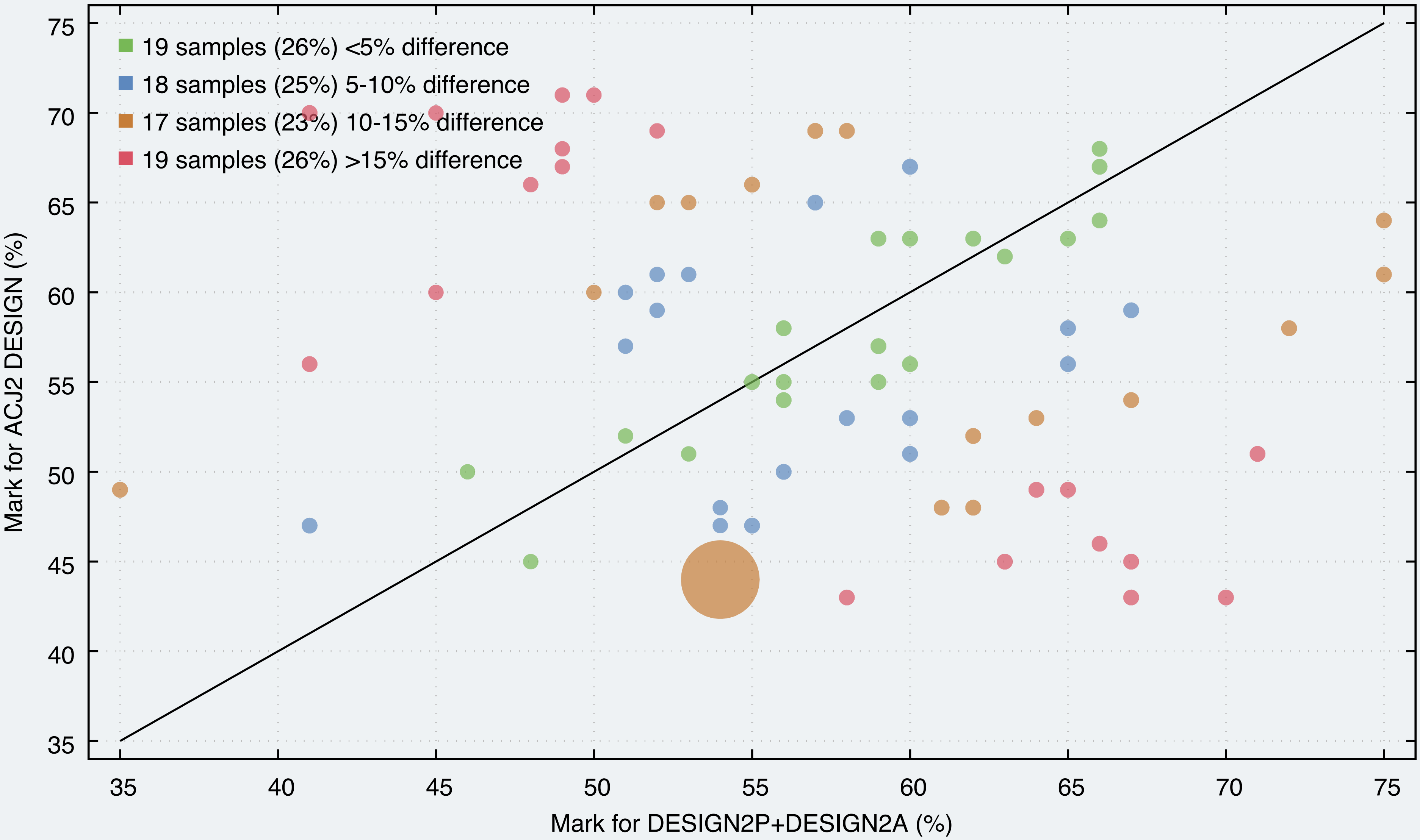
Correlation between ACJ ranking and manual marking was poor

- why ?
  - we need more judgements?
  - we need better algorithms ?
  - we want grade-bands, rather than a full ranking ?
  - we need to be clearer about the criteria ?
  - (novice) students have a different idea for (experienced) staff about what makes a clear description ?
  - it’s just not a good idea!
- or maybe it isn’t as bad as it first seems …

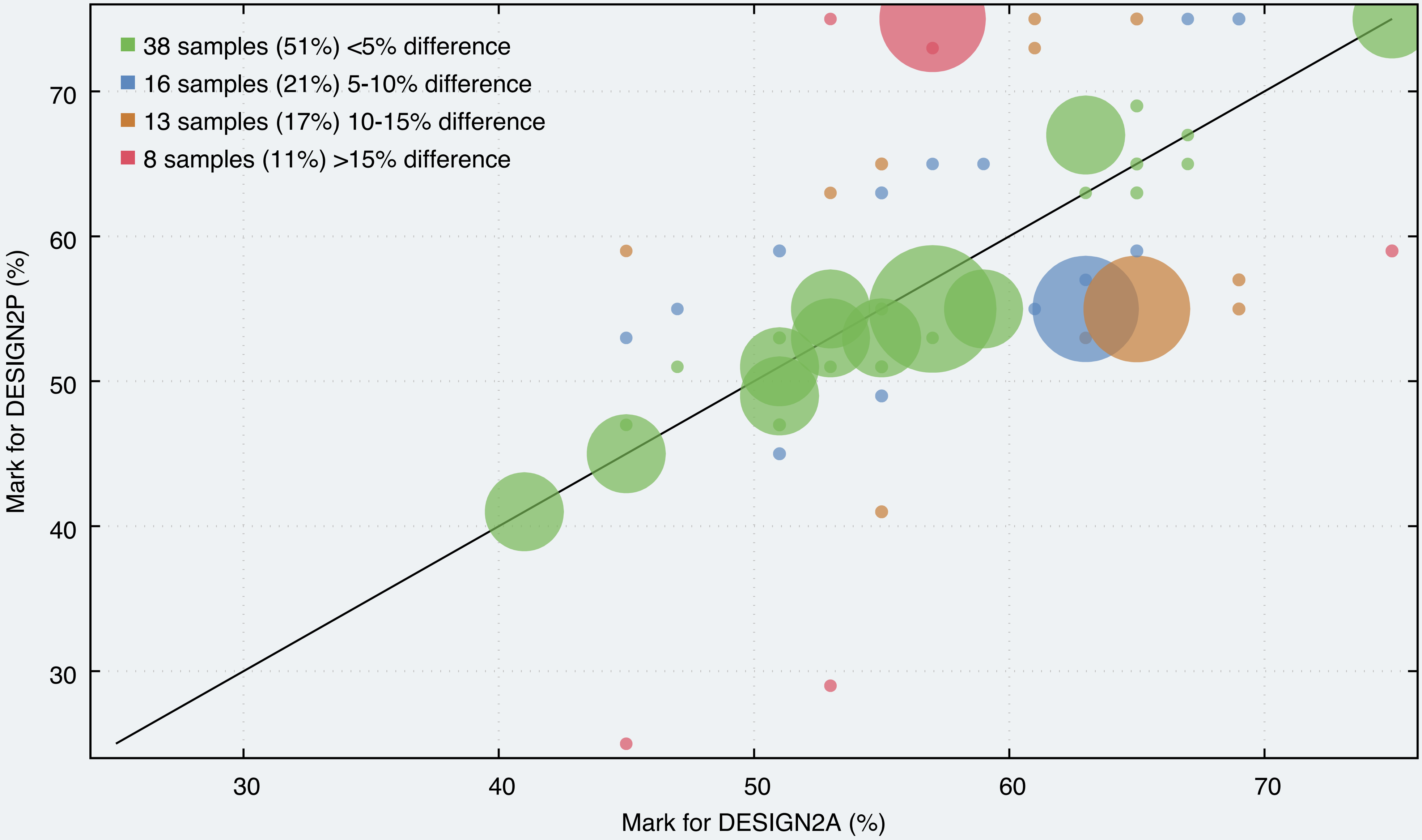
So …

- we double-marked and correlated the manual marking to check
  - this was better
- we did some simulations on the algorithm
  - does not converge well when there is “fuzz”
- we looked at other studies

ACJ vs manual marking

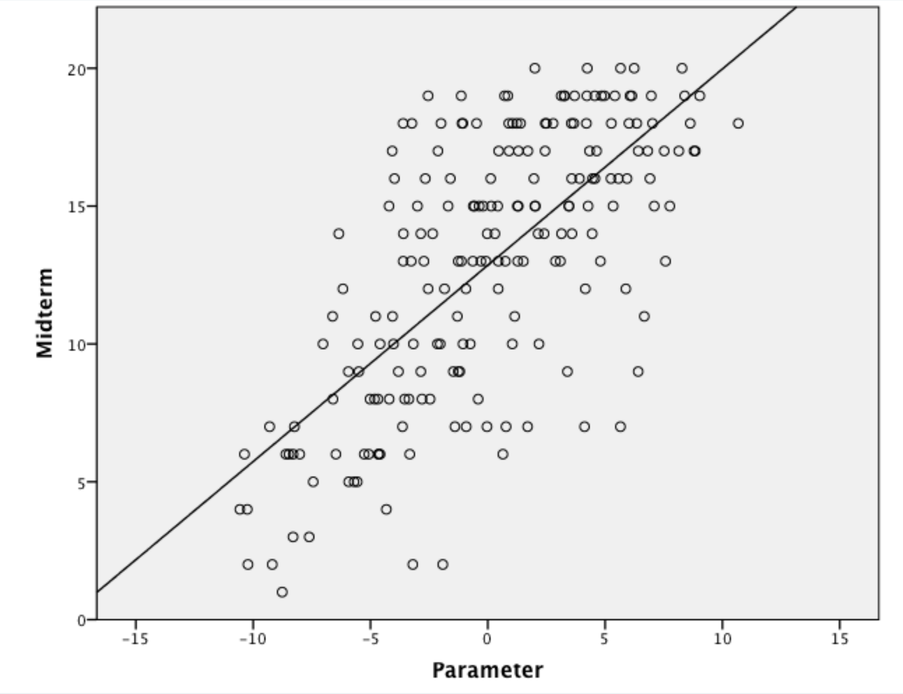


Correlation between markers

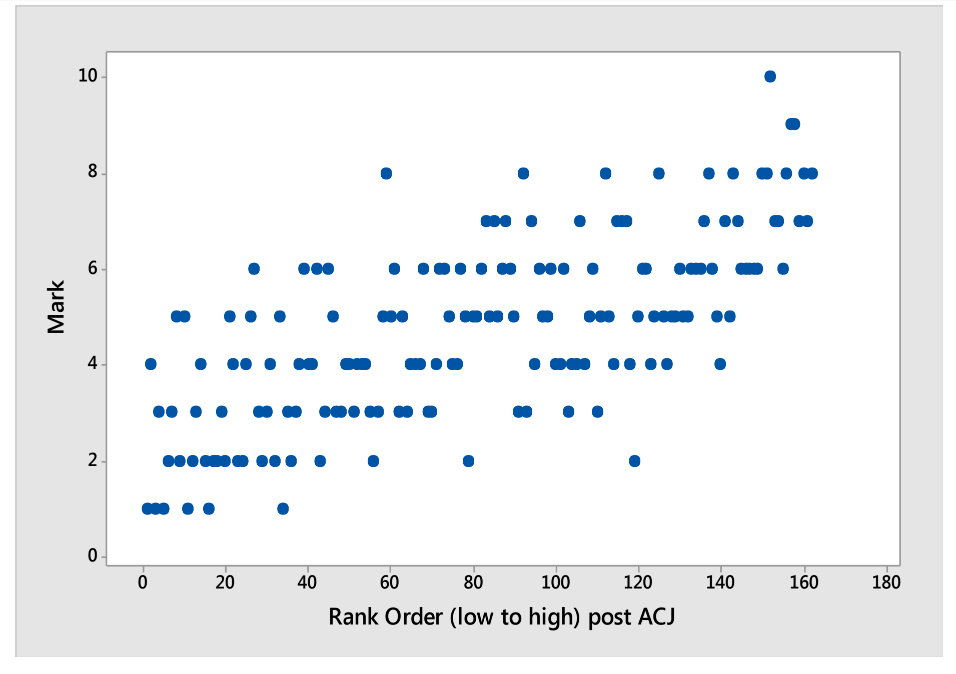


Hardy (2016)

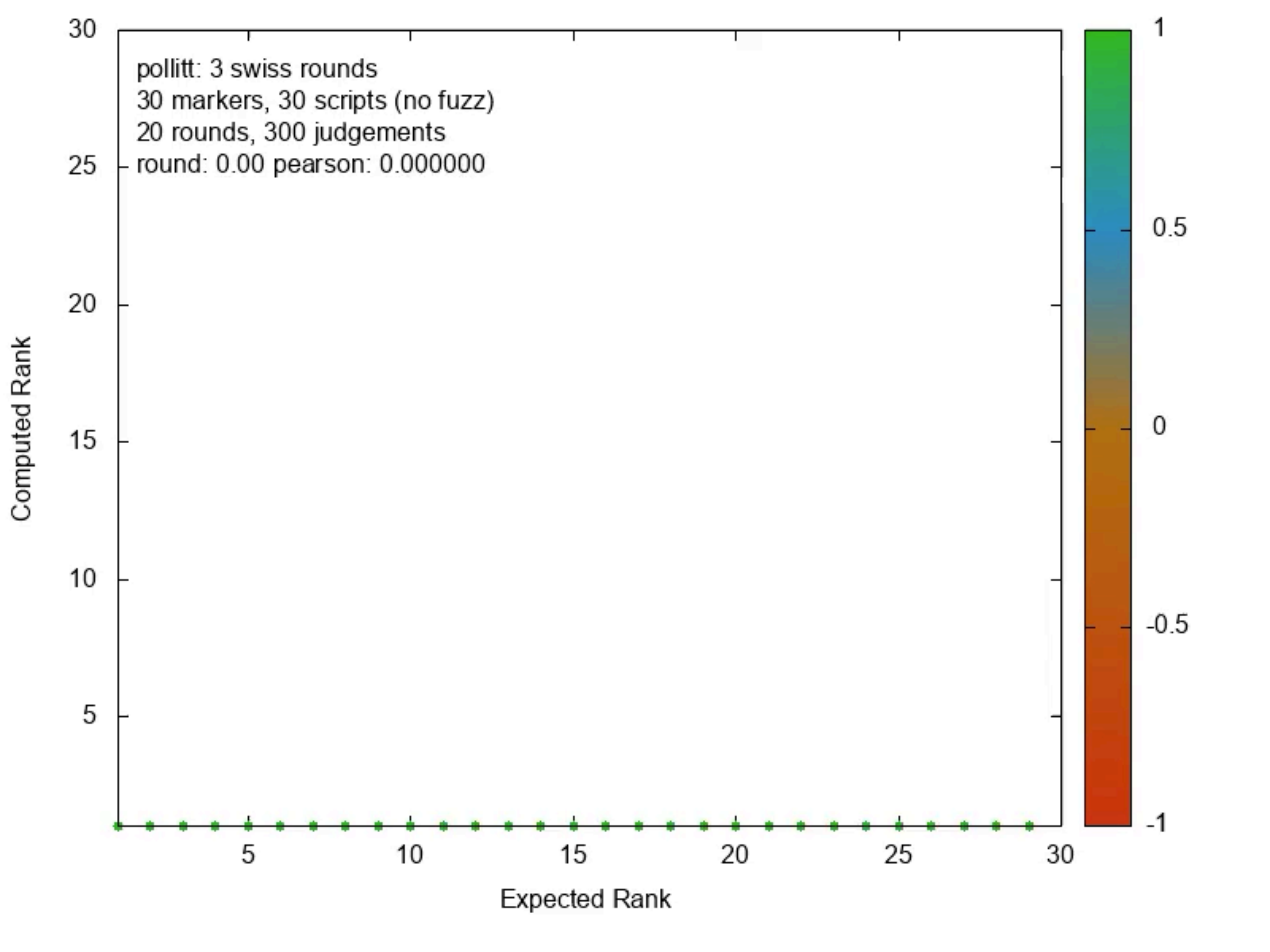
Physics



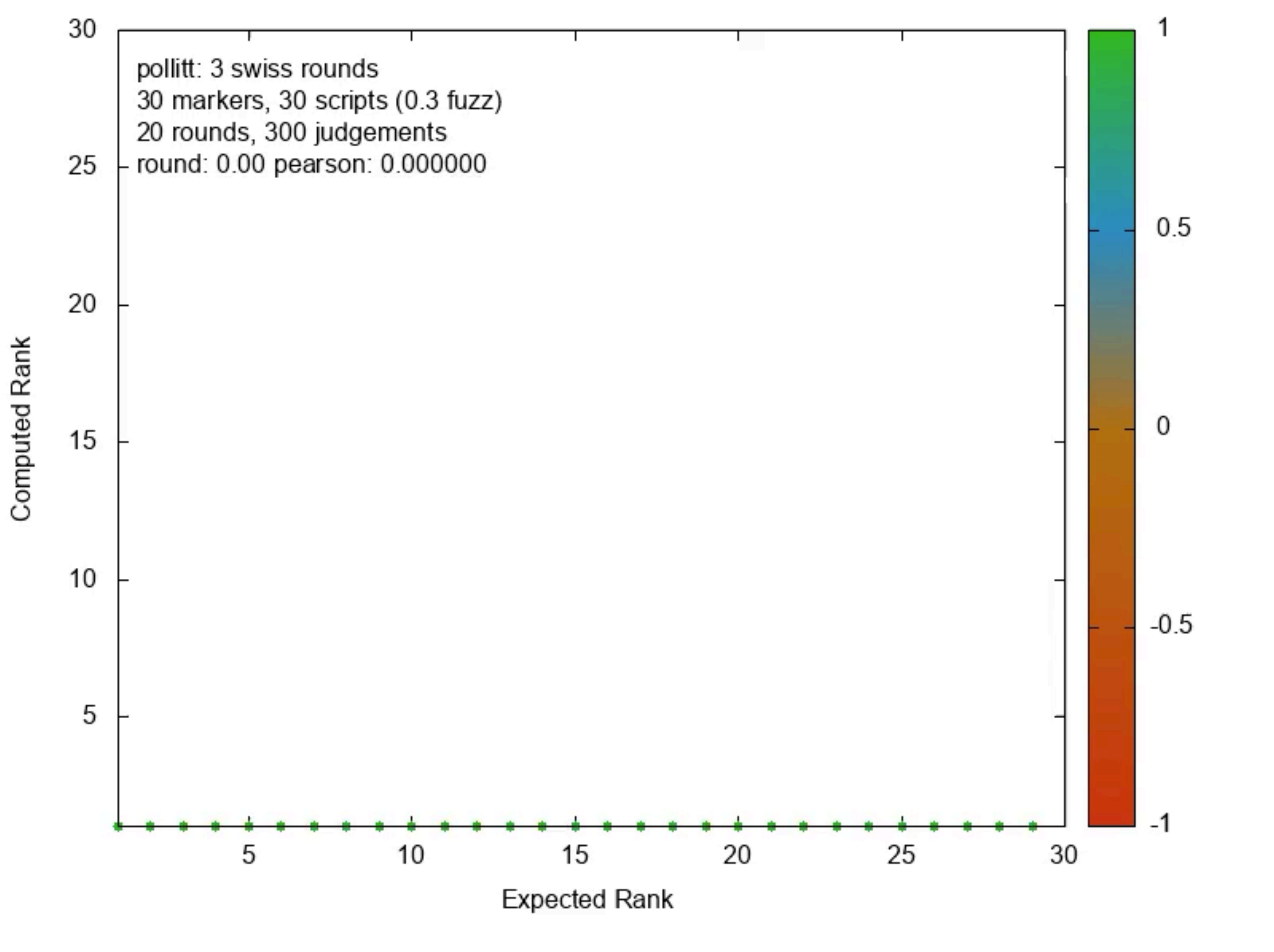
Vets



Simulation

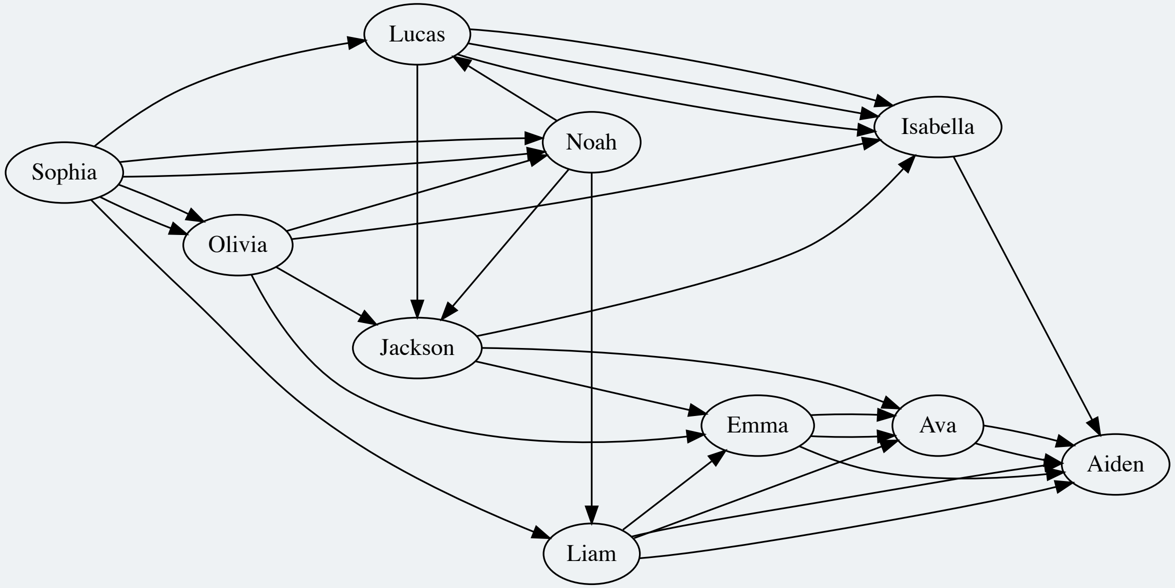

pollitt: 3 swiss rounds
30 markers, 30 scripts (no fuzz)
20 rounds, 300 judgements
round: 0.00 pearson: 0.000000
Computed Rank
Expected Rank
30
25
20
15
10
5
5
10
15
20
25
30
1
0.5
0
-0.5
-1

Simulation with “fuzz”

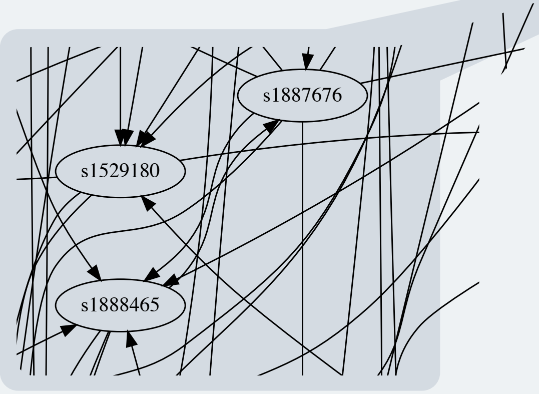


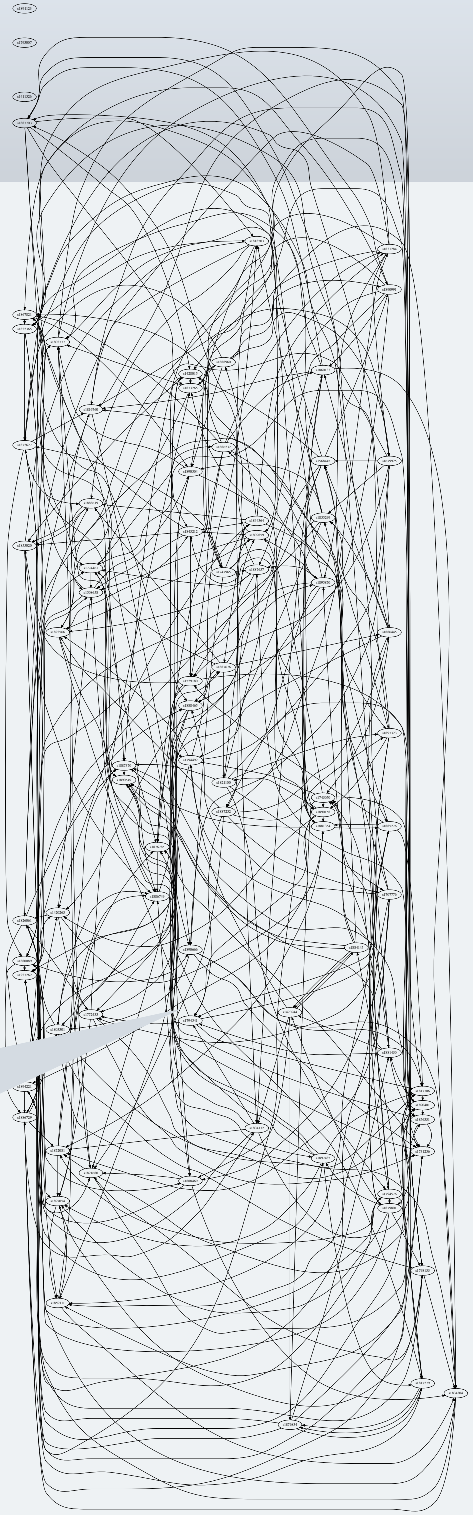
Judgements

Simulated judgements
10 scripts, 30 judgements



“Real” judgements
80 scripts, 350 judgements





Effect of “fuzz” factor

	Fuzz factor					
Difference in mark	0.0	0.1	0.2	0.3	0.4	0.5
0	0.0	50.0	50.0	50.0	50.0	50.0
1	0.0	41.7	45.8	47.2	47.9	48.3
2	0.0	33.9	41.7	44.5	45.8	46.7
3	0.0	26.9	37.8	41.7	43.8	45.0
5	0.0	15.9	30.3	36.5	39.7	41.7
10	0.0	3.4	15.9	24.8	30.3	33.9
20	0.0	0.1	3.4	9.8	15.9	20.9
30	0.0	0.0	0.7	3.4	7.6	11.9

Table shows probability (%) of the simulator generating a “wrong” result

Conclusions

Peer ACJ seems to have potential for ...

- encouraging students to see a range of approaches
- improving assessment literacy
- providing at least an aid to assessment which scales very well
- understanding differences between staff and student perspectives

Issues ...

- more work needed on the algorithms, interfaces & experiments
  - anyone with expertise on such algorithms ?
  - an interesting Phd project ?
- conflicting requirements for assessment & literacy
  - E.g. seeing too many pairs which were too similar was unhelpful

I am continuing to work on this "in the background”

- building an experimental framework
- interest @ Glasgow (used it for allocating conference reviews)

References

Bloxham 2016

- *Let's stop the pretence of consistent marking: exploring the multiple limitations of assessment criteria*
  Sue Bloxham, Birgit den-Outer and Jane Hudson and Margaret Price
  Assessment & Evaluation in Higher Education, 2016

Sadler 2009

- *Indeterminacy in the use of preset criteria for assessment and grading*
  D. Royce Sadler
  Assessment & Evaluation in Higher Education, 2009

Pollitt 2012

- *The method of Adaptive Comparative Judgement*
  Alastair Pollitt
  Assessment in Education: Principles, Policy & Practice, 2012

Hardy 2016

- *Ask, Answer, Assess: Peer learning from student-generated content*
  Judy Hardy, Ross Galloway, Susan Rhind, Karon McBride, Kirsty Hughes and Robyn Donnelly
  Higher Education Academy, 2016