UNIVERSITY OF WATERLOO

*** Solutions *** CS 349 Midterm Exam Fall 2013

Course Title: User Interfaces
Sections: All
Instructor: Byron Weber Becker

Exam Date: 23-October-2013
Time: 4:30 - 5:50p.m.
Duration: 80 minutes
Pages: 10

Exam Type: Closed Book
Permitted Aids: None

Cheating is an academic offense. Your signature on this exam indicates that you understand and agree to the University's policies regarding cheating on exams.

Signature: _____

Last Name: _____

Given Name: _____

Student ID: ___ ___ ___ ___ ___ ___ ___ ___

Instructions:

1. Place answers in this booklet. Use the last page if additional space is required.
2. Before you begin, make certain that you have 10 pages.
3. There are 6 questions on the exam. Answer all questions and each part of each question. The marks assigned to each question are shown at the beginning of the question. Use this information to organize your time effectively.
4. Questions will not be interpreted. Proctors will confirm or deny errors only. If you consider the wording of a question to be ambiguous, state your assumptions clearly and proceed to answer the question to the best of your ability.
5. Documentation is always appreciated, but is not required unless explicitly requested.

This exam has been made public in two forms. The other one is as printed for the exam. It's a good one to look at first to test your understanding. This one has sample solutions as well as the instructor's post-exam comments.

Really, sit down in a quiet space for 80 minutes without aids and *do* the other exam before you look at this one.

Q #	Marks Available	Marks Earned	Grader Initials
1	11		
2	14		
3	16		
4	17		
5	08		
6	07		
Total	73		

Mean (without zeros): 61.9%
Median (without zeros): 63.7%
Max: 86.9%
109 students writing.
The exam as a whole was too long. Only several students left before the end of the allotted time.
See the end for overall mark adjustments.

Q1 average was 75%.

Name: _____

1. [11] Events

a. [08] In response to a student question, the following event loop pseudocode was posted to Piazza:

```
while (true)
  if (events) handle one
  if (time to paid) paint
  if (no events) sleep a while
```

Analyze the following variants. In each briefly describe conditions where the program does not behave as desired and what that undesired behaviour is.

`while (true)`
`  if (events)`
`  handle one`
`  if (time to paint)`
`  paint`
`  sleep a while`
	If events arrive quickly, the program will lag behind due to sleeping after each event.

`while (true)`
`  while (events)`
`  handle one`
`  if (time to paint)`
`  paint`
`  if (no events)`
`  sleep a while`
	If events arrive quickly, the screen won't be repainted on a timely basis because the events must all be handled before repainting.

`while (true)`
`  if (events)`
`  handle one`
`  else if (time to paint)`
`  paint`
`  else if (no events)`
`  sleep a while`
	If events arrive quickly, the screen won't be repainted on a timely basis because the events must all be handled before repainting.

`while (true)`
`  if (events)`
`  handle one`
`  else if (time to paint)`
`  paint`
	The program consumes all available CPU cycles, whether there are events to process or not..

b. [03] In XWindows, the pseudocode `if (events) handle one` is implemented with XPending and XNextEvent. What is the purpose of each of these functions? In what kinds of programs would we use one (which one?) without the other?

XPending indicates how many events are in the event queue. It does not block.
XNextEvent gets the next event from the queue, blocking if the queue is empty.

Because XPending doesn't actually get an event, we can't use only that one. We could use XNextEvent by itself in cases where the program does nothing between events. For example, no animation.

Students were distracted by the marking scheme. It should have been discussed before the exam. I wanted to make clear that there was a penalty for guessing. I think the confusion is that students often think about "earning" marks while markers often think about deducting marks. Yes, leaving it all blank would have "earned" ½ a mark.

2. [14] XWindows and Drawing

a. [05] XWindows: Mark each statement with "T" (true) or "F" (false).

[Marking: -1 for each incorrect T/F answer and -1/2 for each blank answer to a floor of 0.]

4th one (window mgr owns..) was judged to be really tricky; only deducted ½ if wrong

__*F* The Base Window System is multi-threaded to handle multiple windows.
__ *F* The Base Window System resides on the XWindows client.
__ *F* Network speed has increased enough since the mid-1980's that programmers can pretty much ignore it.
__ *F* The window manager is a separate application that "owns" the title bar of each window; the application using the window "owns" everything else.
__*T* X is great for UI researchers because it's really easy to change the window manager to obtain a different look and feel.
__ *T* X requires programmers to select which events their programs receive with a set of predefined flags that are ORed together.
__ *T* A graphics context includes settings related to background colours, line widths, line joining policies, and how newly drawn bits are combined with bits already on the screen.
__ *F* Because there is only one graphics context, programmers need a disciplined policy in how to use it to avoid sending extraneous commands across the network.
__ *F* The Xwindows client sends events to the server; the server sends drawing commands to the client.

b. [04] Two performance-enhancing techniques we discussed were clipping and double buffering. Succinctly explain the problem each is designed to solve and a high-level description of how it solves it.

Q2 average was 71.5%.

Double buffering reduces the flickering we see on the screen due to partial repaints. It maintains an off-screen buffer where all the drawing is done. That buffer is then copied to the screen in a single operation.

Clipping allows the system to only update the part of the screen that was damaged while allowing the programmer to simply repaint everything. System routines ignore all drawing instructions that fall outside of the clipping area, potentially saving lots of CPU cycles.

c. [03] Compare and contrast the RGB and HSB colour models. Be sure to indicate what each letter in the name represents.

RGB = Red/Green/Blue – how much of each colour is produced. Commonly used for computer graphics because it corresponds well to how monitors actually produce colours.

HSV = Hue/Saturation/Value (or Brightness). Hue represents the wavelength (and hence "colour") of the colour. Saturation is how much of the colour there is (vibrant red vs. washed out pink). Value/Brightness is how much light is reflected.

Marking:
1 letter meanings
1 RGB explanation
1 HSV explanation

Name: _____

d. [02] What is the Critical Flicker Frequency? What are appropriate values for humans?

The frequency where our perception of intermittent light changes from flickering to continuous light.

Appropriate values depend on a bunch of things, but 30-60Hz is a good range.

Question 3 was far too "BS-able" (except part (a)).
The exam as a whole has a lot of material that depends on memory. Given the nature of the material, I think that's unavoidable. In a lot of ways it's more similar to an Arts course than many of our other CS courses.
One of the reasons I'm making this exam public is so students can better prepare themselves for that aspect of future exams.
Some students appeared to be surprised that history was something worth testing. What surprises me is how many students think that history has no value and isn't worth knowing – unless the "history" is reviewing historical exams!

3. [16] Misc.

a. [03] History: M

[Marking: -1 for each incorrect T/F answer and -1/2 for each blank answer to a floor of 0.]

Q3 average was 74.0%

__*T* Ivan Sutherland created "Sketchpad" which included a constraint solver so one could say, for example, "make these lines parallel."
__*F* Ivan Sutherland invented the mouse to work with his Sketchpad system.
__*T* Doug Engelbart demonstrated a system that allowed people to collaborate across a network.
__*T* Doug Engelbart demonstrated a system that used copy & paste, hypertext, and maps.
__*T* All of the true statements, above, occurred during the 1960's.

b. [02] What's the difference between "interaction" and "interface"?

The interface is the place where a system presents information to the user and the user can communicate intent.

Interaction is the behavior that unfolds over time in the interface.

Many students only mentioned the system presenting information, missing the communication of intent.

c. [02] Compare and contrast "cognitive conscious" and "cognitive unconscious".

Cognitive conscious is our higher-level thought processes. It can solve complicated problems, but can only focus on one thing at a time. Handles decisions, novel situations, etc.

Cognitive unconscious are those activities that we're not generally aware of while we do them – walking, breathing, etc. It's concerned with repetition, expected events, non-branching tasks. It has a huge capacity for small problems.

d. [03] With respect to user interfaces, what is a desirable consequence of "automatic actions"? What is an undesirable consequence? Give examples.

Automatic actions allow us to get lots of work done without thinking about it. A touch typist, for example, doesn't think about which keys to press.

Actions we do repeatedly become automatic. That's not good for dismissing warning messages, for example, where automatically dismissing a "do you really want to delete the file" could have irreversible (bad) consequences.

It became clear while marking that many students were guessing what "automatic actions" were. The question should have asked them to define the term. About 90% of the answers mentioned deleting files as a bad consequence (surprise, surprise).

e. [06] Provide a sketch showing a better design for the interface shown on the right. Annotate your sketch to show where you used two different Gestalt principles to improve the design. Separately, give a brief definition of the two principles you used.

Marking:
2 a pleasing diagram that is appropriately labeled with two reasonable Gestalt principles
2 the names of two applicable Gestalt principles
2 appropriate definitions of each named Gestalt principle

Almost everyone came us with a design that was perfectly reasonable and used appropriate Gestalt principles. Knowing which principles there were using and giving appropriate definitions was a different matter.

In the future, I'd like to include more comments in lecture about how each principle shows up in UI design. Figure-Ground (Area), for example, isn't a big one other than the widgets are the figure on the ground of the window. Proximity and Similarity, on the other hand, is a big one. Uniform Connectedness is a big one, but is over-used.

Many students made up a principle of "Alignment". In retrospect, lectures should have said more about where this fits (similarity).

Q4 average was 52.8%.

Name: _____

4. [17] 2D Graphics

Consider a graphics editor with a set of templates for drawing common shapes. There are three predefined templates, a hexagon, a star, and a speech bubble. Each of the templates is defined with respect to the origin, as shown in Figure 2A. The user first positions a copy of each template as shown in Figure 2B. The user then groups the hexagon and speech bubble, rotating them 45° about the centre of the hexagon, (3,4), to arrive at the final drawing, shown in Figure 2C.

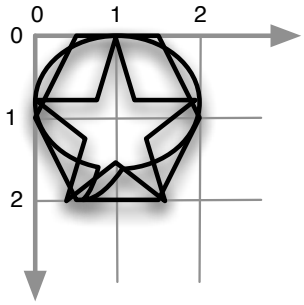


Figure 2A

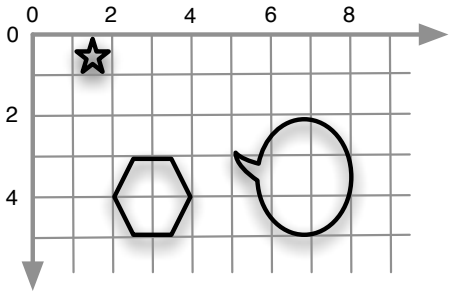


Figure 2B

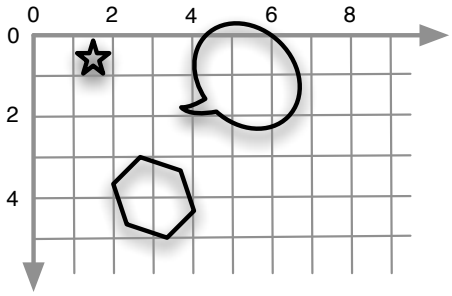


Figure 2C

In the following questions, use $T_{(dx, dy)}$, $R_{(\Theta)}$, and $S_{(sx,sy)}$ to indicate the appropriate translation, rotation, and scaling matrices. Use I for the identity matrix. Use degrees for rotations. Fill in appropriate values for dx, dy, Θ, sx, and sy, of course.

a. [02] What is the affine transformation matrix that $T_{(dx, dy)}$ represents?

$$\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}$$

Marking:
1 correct placement of tx and ty
1 everything else is correct

I thought this one was easy enough that students would have no trouble remembering it. I was wrong.

TAs thought it was a mistake for me to ask for an actual matrix after just saying that such matrices should be represented as T, R, and S.

b. [04] What are the affine transformation matrices, expressed in terms of T, R, and S, for the shapes in Figure 2B?

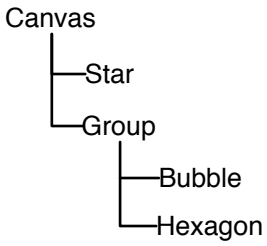
Star: *T(1, 0) * S(0.5, 0.5)*

Hexagon: *T(2, 3)*

Bubble: *T(8, 2) * R(90) * S(3/2, 3/2)*

Many students had difficulties with the order of operations.

c. [02] Assume that the hexagon and speech bubble in Figure 2C are still grouped so they can be operated on together. Draw an appropriate scene graph.



After A2's focus on scene graphs, I thought this would be an easy question. Again, I was wrong.

Markers ignored the presence or absence of the root node ("canvas"). The crucial part was that bubble and hexagon had a common ancestor.

d. [04] Given the description of the user's actions, what would be the transformation matrix for each node of your scene graph to yield the image in Figure 2C?

Canvas: I (this one can be omitted)

*Star: T(1, 0) * S(0.5, 0.5)*

*Group: T(3,4)*R(-45) * T(-3, -4)*

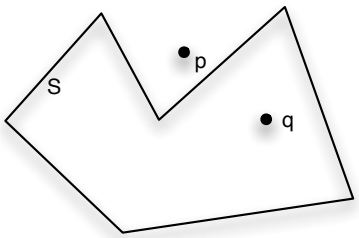
Hexagon: T(2, 3)

*Bubble: T(8, 2) * R(90) * S(3/2, 3/2)*

Marking:
1 Copying answers from part b for star, bubble, and hexagon
3 group

Given that students had difficulty with part (c), it's not surprising that this one was also poorly done.

e. [05] Inside Tests: Describe the algorithm for determining whether points such as p and q are inside or outside of the shape S.



Define a horizontal line through the point and extending past the left edge of the shape's bounding box. Count the number of times the line passes through the shape. If the number is even, the point is outside S. If the number is odd, the shape is inside.

Exception: If the horizontal line passes through a vertex of S, add 2 to the count if the adjoining edges form an V or inverted V; add 1 if the adjoining edges form a sideways V

Marking:
3 first paragraph
2 second paragraph

Name: _____

5. [08] Model-View-Controller

The model classes in the MVC architecture have common code that can be factored out into a superclass. Here is the model for a counter application that extends such a superclass.

```
public class CounterModel extends DefaultModel {
  private int count = 0;
  public void increment()  {  this.count = this.count + 1;  }
  public void reset()      {  this.count = 0;               }
  public int getCount()    {  return this.count;            }
}
```

Q5 average was 34%. Nearly half of the students (50) had a zero for this question. Some of those were probably "earned" via poor answers, but many were simply blank – presumably due to lack of time.

a. [07] Implement **`DefaultModel`**.

```
public class DefaultModel {
    private ArrayList<IView> views = new ArrayList<IView>();
    public void addView(IView v) {
        this.views.add(v);
        this.notify();
    }
    public void notify() {
        for(IView v : this.views) {
            v.updateView();
        }
    }
}
```

It's possible that this was an unfair question, but I don't think so. Student had not yet had an assignment on MVC but had all the lecture content. Based on lecture content, I think students should have understood what the common parts are that can be factored out. Given that understanding, everyone should be able to do the programming.

Marking:
2 list of views
3 addView (header, add to list, notify)
2 notify (exists, loop to notify)

b. [01] Describe any change(s) required to **`CounterModel`**.

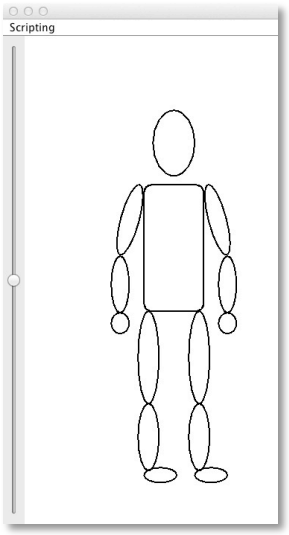
Call notify() at the end of the increment() and reset() methods.

Q6 average was 48.9%.

Of all the questions, this is the one I hesitated most over asking. It asks for a good memory of the terms as well as being able to apply them. In the future, I'd consider giving some of the terms.

6. [07] Direct Manipulation and Instrumental Interaction

One model for evaluating interfaces is "instrumental interaction". Answer the following questions about instrumental interaction with reference to the design for A2 shown on the right. The slider on the left side of the frame is used to control the length of the selected leg. Click and drag a body part to rotate or move it.



a. [02] Instrumental interaction distinguishes between objects and instruments. Define each of these.

Define Object:	Define Instrument:
The thing we're interested in – data and associated attributes. In A2 it was the doll.	*The things we use to manipulate the objects. In the design shown, it would be the slider and/or any handles on the body parts and/or simply areas that we click to drag the body parts.*

b. [05] Instrumental interaction uses three measures to evaluate an interface: degree of indirection, degree of integration, and degree of compatibility. For each of these three measures, (a) provide a brief definition, and (b) use it to evaluate the above interface. Assume the user has a mouse.

Degree of indirection:
Spatial/temporal offset between the instrument and the action on an object.
Typically plotted on a 2D chart with spatial and temporal offsets each forming an access.
For the above interface, spatial is high (bad) but temporal is low (good).

Marking: 2 marks each for the best 2 answers; 1 mark for the not-as-good answer.

Degree of integration:
Ratio of degrees of freedom of the instrument to degrees of freedom of the input device.
Slider: mouse has 2 dof, slider has 1. So that works well (2/1 <= 1).
Rotate and drag: needs 2 dof, mouse has 2 dof. That also works well because 2/2 <= 1.

Degree of compatibility:
Similarity of action on an instrument to the action on the object itself.
Actions on the slider are similar to stretching. So OK. Rotating and dragging are better.

Name: _____

Place continued answers here with appropriate notations both here and where the question is asked.

In recognition of the length of the exam, it will be recorded as out of 66 marks rather than 73. This brings the mean up to 68.5%.