CS148 Midterm

Winter 2009

NAME:

Note: This is exam is closed-book and closed-notes.

The exam consists of 5 questions. Each question is worth 20 points. Please answer all the questions in the space provided, overflowing on to the back of the page if necessary.

This exam has been designed to take 1 hr. However, you have 2 hours to complete the exam.

Computer graphics relies heavily on transformations. Most common are the linear transformations such as rotations, translations, and scales. Recall that OpenGL provides these as:

```c
void GetCenter(float center[3]);
void Draw();
```

For example, calling object->Draw() will draw the object with whatever transformation the library has set. If you want to draw the object rotated by 45 degrees around the Y-axis about its center (e.g., to make it spin), what sequence of commands do you need to perform? Don’t worry about preserving the OpenGL transformation for later code.

```c
float center[3];
object->GetCenter(center);

// NOTE: The center[1] is optional since it won’t be affected by rotation
glTranslatef(center[0], center[1], center[2]);
glRotatef(45.f, 0, 1, 0);
glTranslatef(-center[0], -center[1], -center[2]);
object->Draw();
```

Grading Criteria:

-4 Wrong Order of Ops (Translate(-center); Rotate(); Translate(+center))
-4 Missing –translate
-2 Missing +translate
-10 Only Rotate (since it misses both translates and no order is implied)
1B [10 points]. You have a friend who is good at using a modeling package (a program for creating and editing 3D shapes, like Blender or Maya). They have produced a shape that you would like to display with OpenGL.

When creating the shape, your friend applied some transformation in the modeling program that consists of a translation, a rotation and a scale:

\[
\text{Translate}(tx, 0, tz); \\
\text{Rotate}(ra, 0, 1, 0); \\
\text{Scale}(sx, sy, sz);
\]

The rules of the program specify that these will be composed in a specific order. If \( T \) is the translation, \( R \) the rotation and \( S \) the scale, then the transformation of the whole shape will be TRS.

While you like the translation and scale your friend specified, you really want to remove the rotation. You can’t change the shape directly, though – the TRS transformation is “baked in.” You need to devise an additional transformation \( M \) so that \( MTRS \) is equivalent to just TS.

Give a sequence of OpenGL transformations to apply before drawing your friend’s shape that will achieve this effect. You may use the values \( tx, tz, ra, sx, sy \) and \( sz \) given above if you need.

\[
glTranslatef(tx, 0, tz); \\
glRotatef(-ra, 0, 1, 0); \\
glTranslatef(-tx, 0, -tz); \\
\text{shapeFromYourFriend->Draw();}
\]

Grading Criteria:

-2 Extra Operations
-2 No Rotate

“Correct Inverse”
-4 Wrong Order (\( \text{Translate}(-tx), \text{Rotate}, \text{Translate}(+tx) \))
-2 Missing Translate Back (\( \text{Translate}(tx, 0, tz) \))
-2 Missing Translate To Origin (\( \text{Translate}(-tx, 0, -tz) \))

“Common Incorrect Inverse”
-4 Wrong Inverse (\( \text{Scale}(1/sx, ..), \text{Rotate}(-ra), \text{Scale}(sx, ..) \))
-2 Missing Scale Back (\( \text{Scale}(sx, sy, sz) \));
-2 Missing Scale Down (\( \text{Scale}(1/sx, 1/sy, 1/sz) \))
[20 points] Light and Color

2A [2 points]. In graphics, reflectance and light color is usually specified as \((r, g, b)\) values. For a yellow light \((1, 1, 0)\), give a non-black reflectance that produces a black appearance to all observers under this lighting.

Any blue color \((0, 0, b)\) will do.
-2 Answer wasn’t \((0, 0, b)\) where \(0 < b \leq 1\).

2B [8 points]. In reality, reflectance and light color are actually specified as spectral distributions. In this case, your reflectance would instead be a curve \(R(\lambda)\) and the light color would be \(S(\lambda)\), where \(\lambda\) is the wavelength of light. A particularly interesting light distribution is sky light, which on a sunny day has a distribution close to the 5000K line shown below (the line colors don’t correspond to the perceived color of the light):

![Graph of spectral distributions for different temperatures.](image)

The output for each line is positive everywhere but very small as the wavelength goes towards 0 nm. As in part A, can you provide a non-zero reflectance distribution, \(R(\lambda)\), that appears black under 5000K illumination to any observer? Why or why not? What does this say about the differences in computing color combinations in \((r,g,b)\) instead of full spectral distributions?

No, because appearing black would require that \(R(\lambda) S(\lambda)\) is 0 everywhere (for the integral of their product to work out to be 0). In this case since \(S(\lambda) > 0\) everywhere, \(R(\lambda)\) would have to be \(= 0\) (which is the definition of black). This suggests that RGB operations are not equivalent to real light calculations.

Grading Criteria:
-2 Answered “Yes” by arguing that humans can’t see < 200nm.
-2 Answered “Yes” because of missing the \(S(\lambda) > 0\) everywhere.
-4 No comment on RGB vs Spectral
Comment on some difference of RGB vs Spectral, but not true in general.

2C [2 points]. Assume you perceive luminance to be .3*R + .5*G + .2*B. Given the color (.4, .6, .4) provide another rgb value that you will perceive to have the same luminance.

Luminance(.4, .6, .4) = .3 * .4 + .5 * .6 + .2 * .4 = .5
Any color with luminance = .5 works (e.g. (.5, .5, .5)).

-2 If your answer doesn’t have luminance .5

2D [8 points]. Now assume that your friend has a form of color blindness that makes him perceive luminance to be .4*R + .6*G + 0.0*B. Now choose a single color that both you and your friend will perceive to have the same luminance as (.4, .6, .4).

L_me(.4, .6, .4) = .5 (see above)
L_friend(.4, .6, .4) = .4 * .4 + .6 * .6 = .52

.4 * R + .6 * G = .52
.3 * R + .5 * G + .2 * B = .50

Subtract

.1 * R + .1 * G - .2 * B = .02
Let R = G

.2 * R - .2 * B = .02

Then .4 * R + .6 * B = .52 \( \Rightarrow \) R = .52

.2 * .52 - .2 * B = .02
.104 - .2 * B = .02
.084 = .2 * B
B = .42

Answer (.52, .52, .42) (there are many possible answers though)

Grading Criteria:
-4 If L_me(color) = L_friend(color) = .5 or .52
-6 If L_me(color) = .5, but L_friend(color) != either .5 or .52
-6 If L_friend(color) = .52, but L_me(color) != either .5 or .52
-8 if L_me(color) != either .50 or .52 and L_friend(color) != either .5 or .52
-2 If color had invalid components (e.g. valid is 0 <= R <= 1, etc)
3. [20 points] Input

Interactive programs read input from a variety of devices including mice, keyboards and gamepads. There are two ways of getting data from an input device: events and polling. You have used both in your projects for CS 148, using GLUT and STJoystick.

For the following questions, suppose that instead of GLUT you are using a new library that only lets you read the state of the mouse by polling:

```c
void PollMouse();
int GetMouseX();
int GetMouseY();
bool GetMouseButton(); // true for down, false for up
```

3A [5 points] Your program wants to respond when the mouse button is pressed (changes from up to down) or is released. Using the interface above, how would you detect these events? Where/when would this code need to execute?

**Answer:** Use a global/static or other variable to keep track of the old state of the button. In a GLUT timer callback, call PollMouse() to get the current state and compare the result of GetMouseButton() to the saved result. If the values are different, then an event has occurred – if GetMouseButton() returned true, then the mouse was pressed, if false then the mouse was released. Set the saved-state variable to the new state.

**Pseudocode:**

```c
void Update() {
    static bool prev = false;
    PollMouse();
    bool curr = GetMouseButton();
    if (curr && !prev) handlePressed( GetMouseX(), GetMouseY() );
    if (!curr && prev) handleReleased( GetMouseX(), GetMouseY() );
    prev = curr;
}
```

**Grading:** 1 point each for clearly stating:
- Code should run in application “main loop”, timer callback, display callback or “every frame”
  - No point for “on a regular basis”, “all the time” or a while(true) loop with no explanation of how the rest of the application runs.
- Previous mouse state saved to a variable (global/static/member)
- Call to PollMouse() on every update
- Compare new state to previous state
- Which state maps to pressed event, which to release
Your program draws some rectangular button “widgets” on the screen. The buttons are stored in a list, and those later in the list will draw over (and may overlap) those from earlier in the list.

3B [5 points] When you detect that the mouse button was pressed, how do you determine which button widget (if any) the user clicked?

Answer: Iterate through the list of widgets backwards (from the end to the start). Test if the mouse (x,y) from the click event lies within the bounding volume (rectangle) of the button widget. On the first widget that contains the (x,y) exit/break the loop – that widget was the one clicked.
If iteration completes without a hit, then no widget was clicked.

Grading:
- 3 points for a correct loop structure. Either iteration “backwards” and taking the first hit, or iterating forwards and taking the last hit.
  o One point taken off if it was not clear that the loop should “break” on the first hit
- 2 points for clearly stating that you test if the (x,y) is within the bounds of the widget

In most user interfaces (Windows, MacOS, Flash, …), a button widget does not “fire” immediately when the mouse is pressed down. In fact, these buttons do not fire until the mouse button is released. We can describe this and the other rules for a button’s behavior with a state graph.

A node/state in the graph is based on the state of the mouse: Is the button down? Is the pointer inside the area of the button? The edges/transitions are based on mouse events: button pressed/release, pointer enters/leaves.

3C [10 points] Accurately describe the conditions under which a button should fire, and draw a graph of the states and transitions involved. Clearly identify which transition(s) should cause the button to fire.

Answer: The button should fire when a “pressed” event occurs while the mouse is inside the bounds of the widget and the next “release” event also occurs within the bounds of the widget. This includes the case where the user clicks inside the widget, drags out, drags back in and releases, but not the case where the user clicks outside the widget, drags in and releases.

The canonical graph appeared in the slides for the “Interaction Techniques” lecture:
Grading:
- Two points off for failing to provide a textual description of the conditions for the widget to fire
- Four points off for a graph that was not a finite state machine / DFA, so long as the logic was correct - flowcharts and decision trees, for example.
- One point off for not clearly marking the transition/edge that corresponds to a fire event (we wanted an edge, not a path)
- One point off for showing “fire” as a terminal node in the state machine.
- For the remaining points, we would “simulate” the student’s state machine on some trial input and take off points if it gave an incorrect response. Inputs we tried and expected responses (starting from an Idle state) included:
  - Enter, Press, Release should fire (2 points off if wrong)
  - Press, Enter, Release should not fire (1 point off if wrong)
  - Enter, Press, Leave, Enter, Release should fire (1 point off…)
  - Enter, Press, Leave, Release should not fire (1 point off…)
  - Press, Release should not fire (1 point off…)
4. [20 points] Bézier Curves

(To receive full credit for this problem, please show your calculations or provide explanations of how you obtained your results.)

4A [10 points]. Suppose you have a cubic Bézier curve segment defined by the following four control points:

\[
\begin{align*}
P0 & = (0, 1) \\
P1 & = \left( \frac{4\sqrt{2} - 4}{3}, 1 \right) \\
P2 & = (1, \frac{4\sqrt{2} - 4}{3}) \\
P3 & = (1, 0)
\end{align*}
\]

What are the coordinates of the midpoint, \( M \), of the curve?

Using Chaiken’s midpoint interpolation algorithm (with \( t = \frac{1}{2} \)) on the 4 control points, you can obtain the following:

\[
\begin{align*}
P^1_0 &= \frac{1}{2}(P0 + P1), & P^1_1 &= \frac{1}{2}(P1 + P2), & P^1_2 &= \frac{1}{2}(P2 + P3) \\
P^2_0 &= \frac{1}{2}(P^1_0 + P^1_1), & P^2_1 &= \frac{1}{2}(P^1_1 + P^1_2) \\
M &= P^3_0 = \frac{1}{2}(P^2_0 + P^2_1) &= \left( \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right)
\end{align*}
\]

Alternatively, you can also use the Bernstein polynomials for a cubic Bézier to evaluate \( M \) with a single equation.

What is the tangent (slope) of the curve at point \( M \)?

The tangent vector of the curve, \( T \), can be properly calculated using

\[
T = s(P^2_0 - P^2_1) = s(1, -1),
\]

with \( s \) being some scalar. This is equivalent to a rise-over-run slope of \(-1\). A curve symmetry and continuity argument will also allow you to arrive at the same result. Note that in general, you cannot use the vector \((P2 - P1)\) nor the vector \((P3 - P0)\) to calculate the tangent for a cubic Bézier!

**Grading:** For each of the two above parts, 2 points were deducted for a providing a correct method without calculating the point \( M \) or the tangent, and 1 point deducted for a calculation error. 2 points were deducted for a correct answer obtained using incorrect or unjustified means. Partial credit was given for solutions on the right track, but with errors in the construction or formula.
4B [5 points]. Now suppose instead that you have a Bézier spline consisting of two quadratic Bézier curve segments, defined by the sequences of control points \((Q_0, Q_1, M)\) and \((M, Q_2, Q_3)\), where \(Q_0 = P_0 = (0, 1)\), \(Q_3 = P_3 = (1, 0)\), and point \(M\) is the midpoint result from part 4A. At what coordinates would you have to place the control points \(Q_1\) and \(Q_2\) so that the tangents of the curve at points \(Q_0, M,\) and \(Q_3\) match those of the curve in part 4A?

In order to make the tangents at the endpoints \(Q_0\) and \(Q_3\) match those of the cubic Bézier in 4A, the first thing to notice is that the slope of \(Q_0Q_1\) must be horizontal, and the slope of \(Q_2Q_3\) must be vertical. Finally, for the tangent to be continuous through the two curve segments at \(M\), the points \(Q_1, M,\) and \(Q_2\) must all be collinear.

Solving for \(Q_1\) and \(Q_2\) using the above constrains, you would obtain

\[
Q_1 = (\sqrt{2} - 1, 1) \quad \text{and} \quad Q_2 = (1, \sqrt{2} - 1).
\]

**Grading:** 2 points were deducted for providing a correct method without calculating the coordinates of the control points, and 1 point deducted for a calculation error. No deductions were made if you started with an incorrect midpoint from 4A. Partial credit was given to incorrect solutions that still realized some of the constraints were necessary, and that provided control points that partially met the constraints.

4C [5 points]. Is the curve in part 4A the same as the curve in part 4B? Briefly justify your answer.

A single cubic Bézier curve does not trace out the same path as the two quadratic Bézier segments, despite sharing the same endpoints, midpoint, and slopes at those points!

Perhaps the simplest way to see this is to actually compute another corresponding point on both curves (other than \(P_0, M,\) or \(P_3\)) – you will see that the points are slightly different. You can also make a mathematical argument that the degree 3 parametrization of the cubic is different than the degree 2 parameterization that results from two quadratics.

**Grading:** 2 points were deducted for answering correctly, but providing an explanation that was imprecise or incorrect. Partial credit was given to solutions answering incorrectly, but still made a good, rigorous argument about all the constraints being the same in both curves for the 3 known points. This was indeed a tricky question, and if the correct solution were determined via democracy, then it seems the consensus would be that these Bézier curves are the same.

5A [2 points]. What is the difference between a point and a vector?

A point is a position in space. A vector is defined by both a magnitude and a direction, but is not tied to any particular position.

Grading Criteria:
-1 if a difference was not stated (i.e. only vector defined, but no mention of a point)
-1 if answer mentioned that a vector has a location/position

5B [6 points]. For each operation below, write down whether the result is a point, a vector, or if the operation is invalid.

- Point – point = vector
- Point + vector = point
- Scalar * vector = vector
- Scalar * point = invalid
- Vector - vector = vector
- Point + point = invalid

Grading Criteria:
+1 for each operation, all or nothing

5C [2 points]. Curves can often be represented by either implicit equations or parametric equations (or both). What is the difference between an implicit curve equation and a parametric curve equation?

An implicit curve has the form \( f(x,y) = 0 \). Plugging coordinates into \( f(x,y) \) will yield 0 if the point is on the curve, <0 if the point is inside the curve, and >0 if the point is outside the curve. However, for a parametric curve, the coordinates of points along the curve are defined as a function of some parameter/index, like \( t \) (i.e. \( x = g_x(t), \ y = g_y(t) \)). You can step through \( t \) in order to get back points along the curve.

Grading Criteria:
Full points if a reasonable difference was given.
-1 if definitions were swapped
-1 if one definition was given, but the other definition or mention of the difference was missing
5D [10 points]. Your friend, Sally, is making a 2D bubble-popping game and needs your help with the implementation. Bubbles are drawn as circles that move across the screen. The player pops the bubbles by clicking inside or on the borders of the bubbles.

Fill in the method templates below with some pseudocode for how you might draw a bubble and how you might determine if a bubble has been popped. You are given the center point and radius of the bubble and the point where a player has clicked (Hint: Think implicit and parametric curves)

```cpp
void drawCircle(Point center, float radius)
{
    glPushMatrix();
    Anything that indicates parameterizing along an angle, and drawing the circle by plotting the points (radius*cosΘ+center.x, radius*sinΘ+center.y). The points should be plotted from Θ=0 to 360 degrees (or equivalent radians). Variations like plotting (cosΘ, sinΘ) and then using glScale and glTranslate to first scale the circle to the radius and then translating to the center point also works.

    Example:
    glBegin(GL_POLYGON);
    float increment = 0.01;
    for (float t=pi; t<=2pi; t+=increment)
    {
        glVertex2f(radius*cos(t)+center.x, radius*sin(t)+center.y)
    }
    glEnd();
    glPopMatrix();
}

bool isPopped(Point center, float radius, Point mouse)
{
    The bubble is popped if (mouse.x-center.x)² + (mouse.y-center.y)² - radius² <= 0.

    Example:
    return ((mouse.x-center.x)² + (mouse.y-center.y)² - radius² <= 0)
}
```
Grading Criteria:

The methods were worth 5 points each:

For drawCircle:
-1 if did not take into account center point of the circle (drawing at origin, instead of center)
-1 if did not take into account radius, or draws the wrong radius
-1 for swapping the order of glScale and glTranslate (if used)
-1 if (different) method works but is inefficient (i.e. iterating through pixels in bounding box and testing to see if it’s on the curve)

For full points, there needed to be enough detail about how to plot the points on the circle (figuring out the coordinates or specifying the transformations needed). If the method does not draw a circle or if it assumed a built-in circle function, then:
+1 for considering translating to the center point
+1 for considering radius
+1 for considering parameterizing along an angle

Note: Answers using glRotate were accepted. However, as a note, glRotate technically cannot be called inside of glBegin and glEnd.

For isPopped:
-1 if did not take into account center point of the circle
-1 if did not take into account radius, or missed a radius \(^2\) if it was needed
-1 if the border of the circle is not taken into account
-1 if (different) method works but is inefficient

For full points, there needed to be enough detail about how to calculate if a point is inside or on the border of the circle. If there is not enough detail or the method does not work (i.e. just testing bounding box):

+1 for considering center point of circle
+1 for considering radius