Problem 1. Short answer questions. Unless otherwise specified, explanations are not required, but may be provided for partial credit.

(a) Suppose that a Unity game object is declared to be static (by checking the “Static” checkbox in the editor). Which of the following optimizations can Unity perform as a result? (Indicate True or False for each.)

(i) Navigation computation and physics can be optimized (because the object’s position is fixed).
(ii) Fewer method calls are needed, because the methods Update or FixedUpdate are not called on static objects.
(iii) Some global lighting computations can be precomputed.
(iv) Space is saves because all instantiations of a static object refer to the same (shared) game object.

(b) What does it mean when we say that the vector dot product (or generally any inner product) is bilinear? (Express your answer as one or more vector equalities.)

(c) You have a long, thin object (e.g., an arrow) that can be oriented arbitrarily in space. Which of the following collider shapes would NOT be a good choice to represent this object (Select all the apply). Briefly explain your answers.

(i) Axis-aligned bounding box (AABB)
(ii) General (arbitrarily oriented) bounding box
(iii) Bounding sphere
(iv) Capsule

(d) Consider the following two computational tasks that arise in animation processing:

Task I: Given the placement of a skeletal model in a scene and an assignment to its joint angles, determine the position of a point of the model (e.g., the tip of the index finger) relative to the scene’s coordinate system.

Task II: Given the placement of a skeletal model in a scene and the desired position of a given point of the model (e.g., the tip of the index finger should be touching a light switch), determine how to set the joint angles to achieve this desired result.

(i) One of the above tasks is called forward kinematics and the other is called inverse kinematics. Which is which?
(ii) Which of these two tasks is computationally more challenging? Briefly justify your answer.
Problem 2. Your new 3-dimensional game involves frisbee throwing. You need to implement an efficient collider that will (roughly) represent the shape of a flying disk. You have chosen to model the frisbee collider as a simple flat circular disk in three dimensional space. The collider is specified by three parameters: (1) the center point \( p = (p_x, p_y, p_z) \) of the collider, (2) a unit-length normal vector \( \vec{u} = (u_x, u_y, u_z) \) that points in the direction perpendicular to the plane on which the disk lies, and (3) a positive real \( r \) that indicates the radius of the disk (see Fig. 1(a)). The disk has zero thickness.

![Figure 1: Problem 2.](image)

The objective of this problem is to derive a procedure that, given a frisbee collider \( \langle p, \vec{u}, r \rangle \) and a line segment \( \overrightarrow{ab} \), where \( a = (a_x, a_y, a_z) \) and \( b = (b_x, b_y, b_z) \), determines whether the frisbee collider intersects the line segment (see Fig. 1(b)). You may assume that \( a \neq b \) and neither of the points \( a \) or \( b \) lies on the plane that contains the collider.

(a) The first step is to determine the equation of the infinite plane containing the collider disk. A point \( q = (x, y, z) \) lies on the plane if and only if the free vector directed from \( p \) to \( q \) is perpendicular to the vector \( \vec{u} \) (see Fig. 1(c)). Use this fact to derive the equation of the plane. (Hint: The plane equation can be expressed in the form \( \alpha x + \beta y + \gamma z + \delta = 0 \) for some scalars \( \alpha, \beta, \gamma, \) and \( \delta \). Derive the values of these four scalars as a function of the coordinates of \( p \) and \( \vec{u} \).)

(b) We showed in class that any point on the infinite line \( \overrightarrow{ab} \) can be expressed as the affine combination \( (1 - t)a + tb \), for some real \( t \). Using your answer from part (a), derive a procedure (in mathematical notation) for computing the value of \( t \) where the infinite line hits the collider plane. Let’s call this point \( q(t) \) (see Fig. 1(d)). Also, present a test to determine whether \( q(t) \) lies within the line segment \( \overrightarrow{ab} \).

(c) Your answer to (b) should involve division by a quantity that depends on the inputs. Under what conditions (as a function of \( \vec{u}, p, a, \) and/or \( b \)) would the divisor(s) be equal to zero? Does the problem description exclude this possibility? (If not, what additional assumptions need to be added?)

(d) Assuming that \( q(t) \) (from part (b)) exists and lies within the line segment \( \overrightarrow{ab} \), explain how to determine whether \( q(t) \) lies within the collider disk of radius \( r \) (see Fig. 1(d)).

Problem 3. You are implementing a football game, and you want to simulate the process of a quarterback throwing the ball to a pass receiver. The receiver is running across the field at
a fixed speed of $s_p$ feet per second, and the quarterback throws the ball at a fixed speed of $s_q$ feet per second. (You may assume that both of these quantities are positive.) The quarterback needs to adjust the angle at which the ball is thrown (thus, leading the receiver) so that the ball arrives at the same time as the receiver.

To simplify matters, let us do this in the 2-dimensional plane. Assume that the quarterback is located at a point $q = (q_x, q_y)$, and at the instant the ball is thrown the receiver is at point $p = (p_x, p_y)$ directly above $q$. Thus, $p_x = q_x$ and $p_y > q_y$ (see Fig. 2(a)). Let $\ell_q = p_y - q_y$ denote the initial distance between the quarterback and receiver.

![Diagram](image)

**Figure 2:** (a) Problem 3 and (b) Challenge problem.

Derive (in mathematical notation) a procedure, which given $q$, $p$, $s_q$ and $s_p$, outputs the angle $\varphi > 0$ of the direction (relative to the vector from $q$ to $p$) at which the quarterback should throw the ball so that the receiver and ball arrive at the same time in the same place (assuming that they move at their given speeds). You may express $\varphi$ either in radians or degrees.

In order for your solution to exist, what assumptions need to be made about the relationship between $s_q$ and $s_p$?

**Problem 4.** We wish to perform a rotation of $\theta$ degrees about a unit vector $\vec{u} = (u_x, u_y, u_z)$ using a quaternion representation (see Fig. 3). We will apply this rotation to a point $p = (p_x, p_y, p_z)$.

Throughout, when asked to present a quaternion, present it either as a 4-element vector or as a pair consisting of a scalar and a 3-element vector.

(a) As a function of $\vec{u}$ and $\theta$, express this rotation as a unit quaternion $\mathbf{q}$. (You may express $\mathbf{q}$ as a 4-element vector or in the form $(s, v)$, where $s$ is a scalar and $v$ is a vector.)

(b) Express the point $p$ as a pure quaternion, denoted $\mathbf{p}$.

(c) As a function of $\vec{u}$ and $\theta$, express $\mathbf{q}^{-1}$. (Hint: The inverse of a rotation is a rotation by the negation of the angle.)

(d) Let $p'$ be the image of $p$ under this rotation. Explain how to obtain the coordinates of $p'$ in terms of quaternion operations on $\mathbf{q}$ and $\mathbf{p}$. (I am looking for a couple of short formulas. You do not need to expand out all the terms of the quaternion multiplication.)

**Problem 5.** Consider a skeletal model of an arm holding a sword in 2-dimensional space. Suppose that the bind pose is as shown in Fig. 4(a), with the arm and sword extending horizontally to the right of the shoulder. The shoulder, elbow, hand, and tip of sword coordinate frames are called $a$, $b$, $c$, and $d$, respectively. It is 6 units from the shoulder to the elbow, 7 units from the elbow to the hand, and 8 units from the hand to the tip of the sword.
(a) Following the naming convention for the local pose transformations (given in Lecture 9) express the following local pose transformations a $3 \times 3$ homogeneous matrices. (In all cases assume the arrangement shown in Fig. 4(a).)

(i) $T_{[c \leftarrow d]}$, which translates coordinates in the sword tip frame to the hand frame.
(ii) $T_{[b \leftarrow c]}$, which translates coordinates in the hand frame to the elbow frame.
(iii) $T_{[a \leftarrow b]}$, which translates coordinates in the elbow frame to the shoulder frame.

For example, the transformation $T_{[c \leftarrow d]}$ should transform the column vector denoting the tip of the sword relative to the tip-of-sword frame coordinate (as the origin) to its representation relative to the hand frame coordinates (as lying 8 units along the $x$-axis). That is,

$$T_{[c \leftarrow d]} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 8 \\ 0 \\ 1 \end{pmatrix}.$$

(b) Show that by multiplying these matrices together in the proper order, we obtain a matrix $T_{[a \leftarrow d]}$ that maps a point in the tip-of-sword frame to the shoulder frame. For example, because the tip lies 21 units to the right of the should, we have

$$T_{[a \leftarrow d]} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 21 \\ 0 \\ 1 \end{pmatrix}.$$
(c) Give the following inverse local pose transformations:

(i) \( T_{[d\leftarrow c]} \), which translates coordinates in the hand frame to the sword tip frame.

(ii) \( T_{[c\leftarrow b]} \), which translates coordinates in the elbow frame to the hand frame.

(iii) \( T_{[b\leftarrow a]} \), which translates coordinates in the shoulder frame to the elbow frame.

(Hint: You can exploit the simple structure of the matrices in part (a) to avoid the need for general matrix inversion.)

(d) Suppose that we apply a rotation by angle \( \theta_b \) about the elbow and \( \theta_c \) about the hand. (These are both \( 90^\circ = \pi/2 \) in Fig. 4(b), but they can be any angle, positive or negative, in general.) Assume that Rot(\( \theta \)) denotes a 3 \times 3 rotation matrix, that is

\[
\text{Rot}(\theta) = \begin{pmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{pmatrix}.
\]

Let’s assume that all points are represented in the shoulder frame. Following the example in Lecture 9, derive a matrix (which you may express as the product of a sequence of matrices) that maps a point representing the tip of the sword in the bind pose to its rotated position. For example, in the particular case where \( \theta_b = \theta_c = 90^\circ \), this would map the vector \((21, 0, 1)\) to \((-2, 7, 1)\). (Your answer should work for any values of \( \theta_b \) and \( \theta_c \).)

Explain how you derived your answer.

**Challenge Problem 1.** Challenge problems count for extra credit points. These additional points are factored in only after the final cutoffs have been set, and can only increase your final grade.

Generalize Problem 3 so that, rather than starting immediately above \( q \), the point \( p \) starts at a distance of \( \ell_p \) to the left of the vertical line passing through \( q \) (see Fig. 2(b)). Let \( \ell_q = p_y - q_y \) denote the vertical distance between \( q \) and \( p \)'s line of travel. Now, \( \varphi \) is defined relative to the vector between \( q \) and \( p \). (If you prefer, you can compute \( \varphi \) relative to the vertical, as before. Just be sure to explain which you are doing.)

**Challenge Problem 2.** Work through Problem 4, but where \( \vec{u} = (0, 0, 1) \), \( \theta = 90^\circ \), and \( p = (1, 0, 2) \). In addition to parts (a)–(d), also add a part (e) where you explicitly compute the coordinates of \( p' \) after the rotation.