University of Colorado
Boulder

Introduction to Virtual Reality

Other Rotation Representations

Professor **Dan Szafir**

Computer Science & ATLAS Institute
University of Colorado Boulder

Goal: specify geometry, positions, translations, rotations, etc. in 3D space

Rotations with Transformation Matrix

What axis do you want to rotate around?

X-axis

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Y-axis

$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

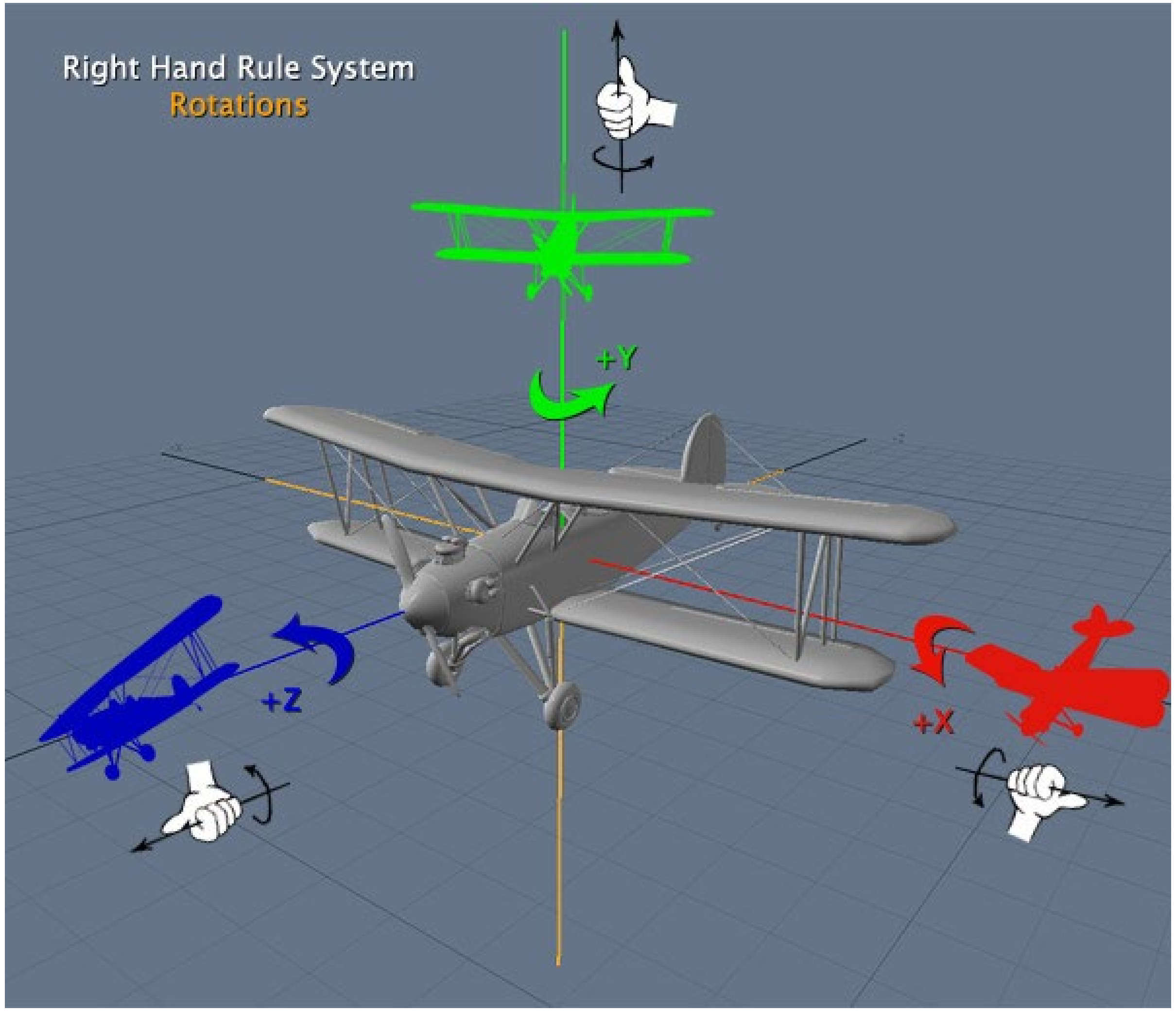
Z-axis

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Which way is positive?

Normally:

Use the right-hand rule



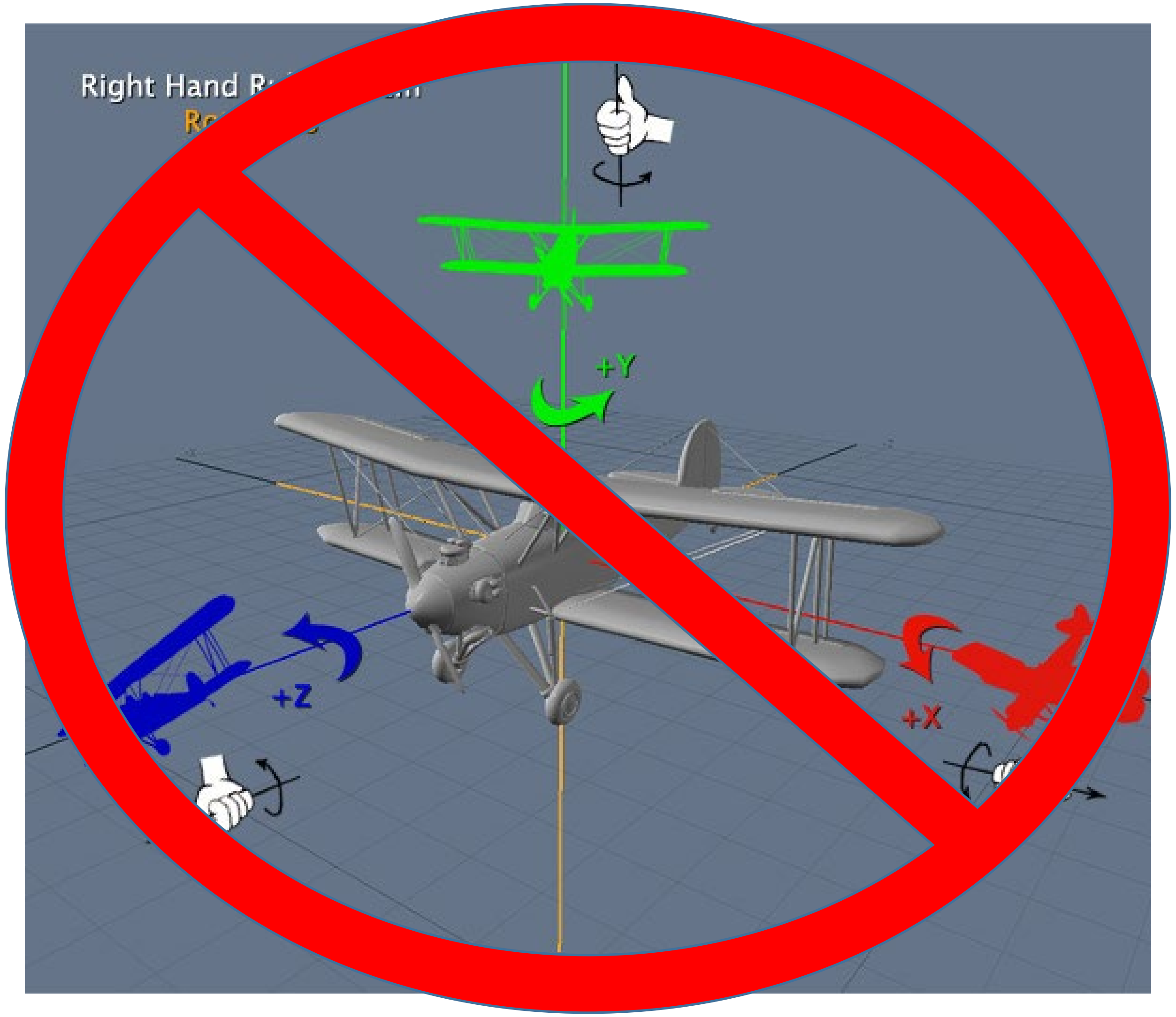
Which way is positive?

Normally:

Use the right-hand rule

Unity:

Left-hand rule



Other Rotation Representations

Euler angles, Axis/Angle, Quaternions,

Limitations of Rotation Matrix

Rotation matrix is over-parameterized (9 parameters instead of 3)

Also hard to visualize

Can we have a minimum representation?

Euler Angles

Product of 3 consecutive rotations around a pre-defined axis

E.g., X – Y – Z (or roll – pitch – yaw)

Rotate 30˚ on X, the 10˚ on Y, the 5˚ on Z

Converting Euler to Rotation Matrix

$$R_X R_Y R_Z$$

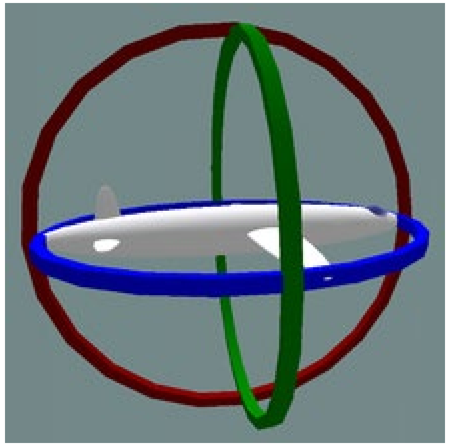
$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

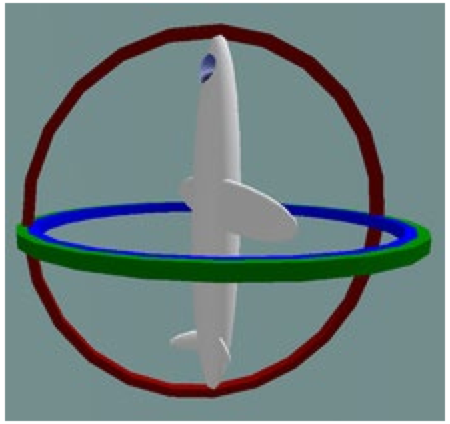
$$= \begin{pmatrix} \cos B & 0 & \sin B \\ \sin A \sin B & \cos A & -\sin A \cos B \\ -\cos A \sin B & \sin A & \cos A \cos B \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos B \cos C & -\cos B \sin C & \sin B \\ \sin A \sin B \cos C + \cos A \sin C & -\sin A \sin B \sin C + \cos A \cos C & -\sin A \cos B \\ -\cos A \sin B \cos C + \sin A \sin C & \cos A \sin B \sin C + \sin A \cos C & \cos A \cos B \end{pmatrix}$$

Ugly! Plus we need a different one for every possible axis ordering (e.g., X – Y – Z, vs X – Z – Y …)

Euler Angles





Advantages

- Minimal representation (3 parameters)
- Easy to understand

Disadvantages

- Many alternatives (X-Y-Z, Z-Y-X, etc.)
- Difficult to concatenate
- Singularities (gimbal lock)

Unity uses Z – X – Y

https://youtu.be/rrUCBOlJdt4?t=120

Axis/Angle

Rotation = rotation axis and angle

4 parameters:

- 3D vector specifying axis
- Size of angle to rotate around axis

OR

3 parameters

- 3D vector specifying axis (vector magnitude is used as angle)
- Not unique (every 2*PI you circle around again)

Can convert to and from rotation matrix

- See Rodriguez' formula and inverse

Axis/Angle

Advantages

- Minimal representation (3 parameters)
- Simple derivations

Disadvantages

- Difficult to concatenate
- Slow conversion to rotation matrix

Quaternions

4 dimensional vector representing rotation

$$Q = xi + yj + zk + w$$

Advantages:

- Multiplication, inversion, and rotations are efficient
- e.g., concatenation only requires multiplication, addition, and subtraction
- No gimbal lock

Can convert to rotation matrix, axis/angle

Used by Unity internally

Quaternions: History

Discovered by Sir William Rowan Hamilton in 1843

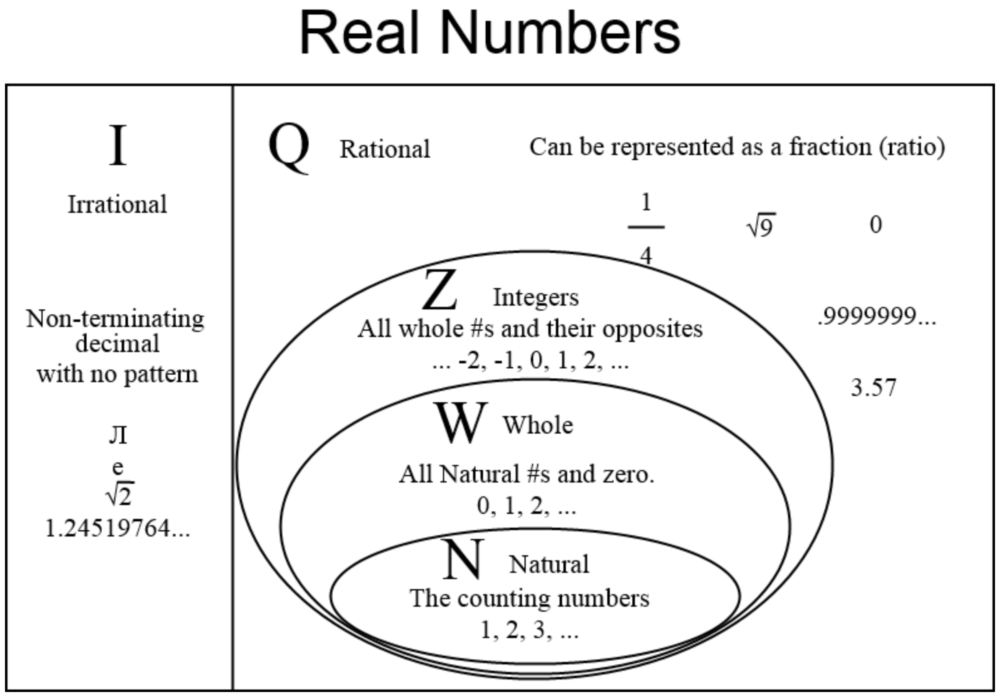
- Attempting to construct an algebra for three dimensions
- Instead realized how to construct an algebra for four!

Mathematical graffiti: $i^2 = j^2 = k^2 = ijk = -1$

Number Sets

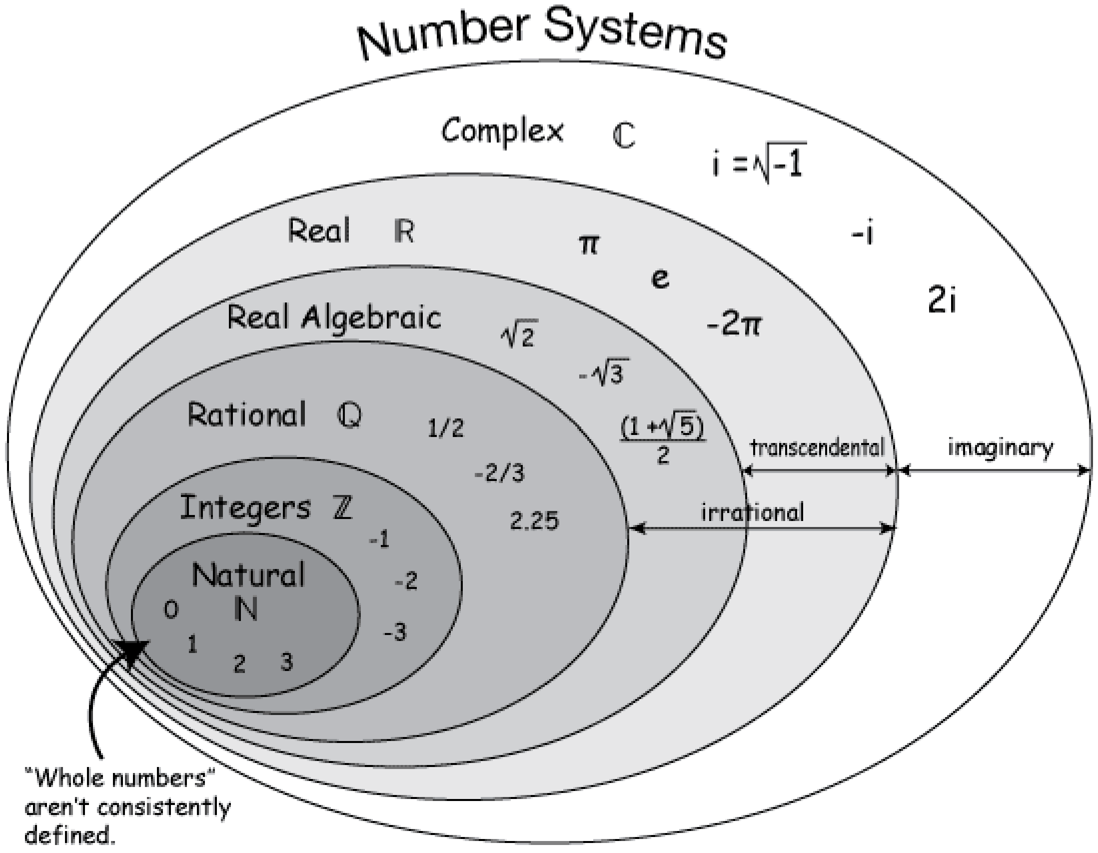
Common number sets:

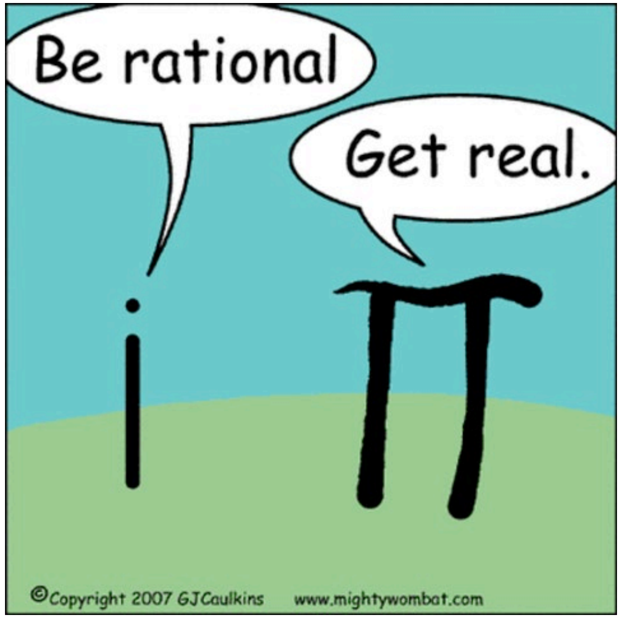
- $\mathbb{N}$: **Natural** – positive whole numbers (maybe 0) – e.g., 0,1,2,3…
- $\mathbb{Z}$: **Integer** – whole numbers – e.g., -2, -1, 0, 1, 2…
- $\mathbb{Q}$: **Rational** – numbers that can be expressed as a fraction – e.g., -4, ½, etc.
- $\mathbb{R}$: **Real** – all rational and irrational numbers – e.g., $\pi, -\sqrt{2}$



Complex Number System

ℂ: Add **imaginary numbers** to the set of real numbers





Imaginary Numbers

Invented to solve certain equations that had no solutions

Example: $x^2 + 1 = 0$

Form: $i^2 = -1$

$\mathbb{C}: z = a + bi \qquad a, b \in \mathbb{R}, \qquad i^2 = -1$

i.e., complex numbers are the sum of a real number and an imaginary number

Operations with Imaginary Numbers

Addition: $(a_1 + b_1 i) + (a_2 + b_2 i) = (a_1 + a_2) + (b_1 + b_2)i$

Subtraction: $(a_1 + b_1 i) - (a_2 + b_2 i) = (a_1 - a_2) + (b_1 - b_2)i$

Scalar multiplication: $\lambda(a + bi) = \lambda a + \lambda bi$

Operations with Imaginary Numbers

Product of complex numbers:

$$\begin{aligned} z_1 &= (a_1 + b_1 i) \\ z_2 &= (a_2 + b_2 i) \\ z_1 z_2 &= (a_1 + b_1 i)(a_2 + b_2 i) \\ &= a_1 a_2 + a_1 b_2 i + b_1 a_2 i + b_1 b_2 i^2 \\ &= (a_1 a_2 - b_1 b_2) + (a_1 b_2 + b_1 a_2) i \end{aligned}$$

Square:

$$\begin{aligned} z &= (a + bi) \\ z^2 &= (a + bi)(a + bi) \\ &= (a^2 - b^2) + 2abi \end{aligned}$$

Operations with Imaginary Numbers

Powers:

$$\begin{aligned}
i^0 &= & & & 1 \\
i^1 &= & & & i \\
i^2 &= & & & -1 \\
i^3 &= ii^2 &= & & -i \\
i^4 &= i^2 i^2 &= & & 1 \\
i^5 &= ii^4 &= & & i \\
i^6 &= ii^5 &= i^2 &= & -1
\end{aligned}$$

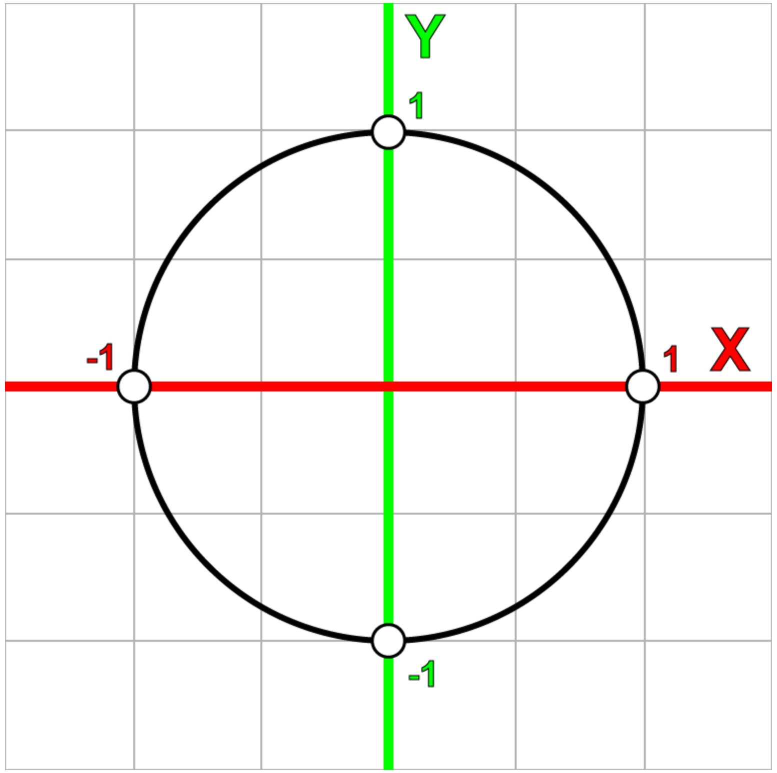
$$\begin{aligned}
i^0 &= 1 \\
i^{-1} &= -i \\
i^{-2} &= -1 \\
i^{-3} &= i \\
i^{-4} &= 1 \\
i^{-5} &= -i \\
i^{-6} &= -1
\end{aligned}$$

What pattern do you see?

What does this remind you of?

Rotations

(x, y, -x, -y, x, …): rotate a point 90 degrees on 2D plane

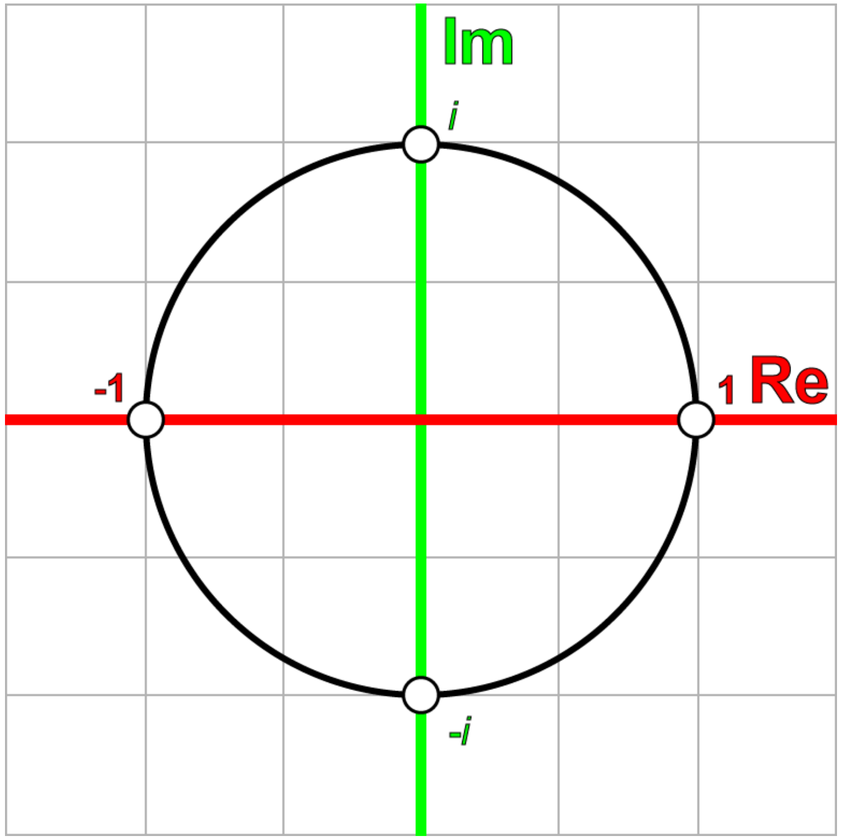


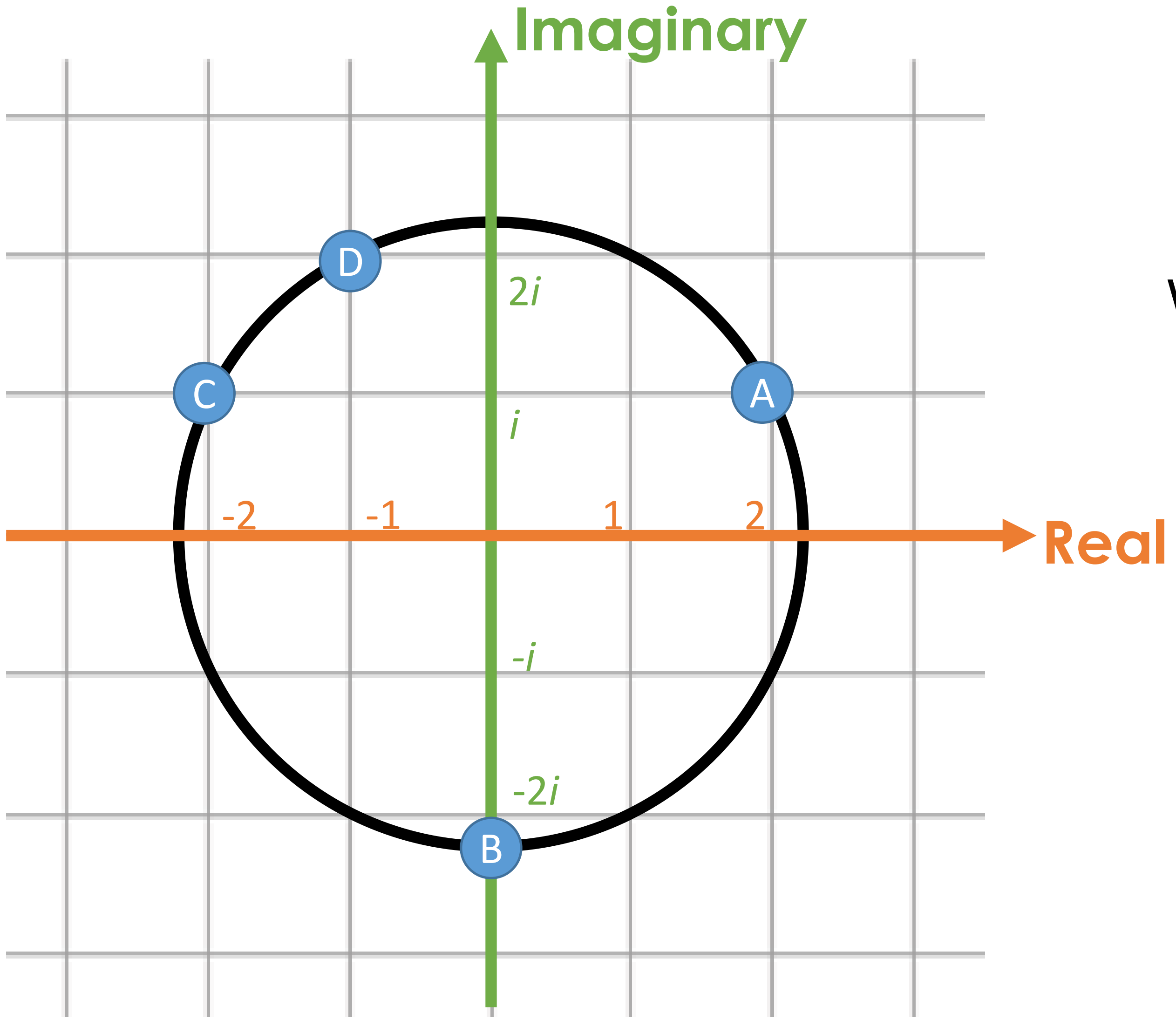
Complex Plane

Can map complex numbers onto a 2D grid

- Horizontal axis = real component
- Vertical axis = imaginary component

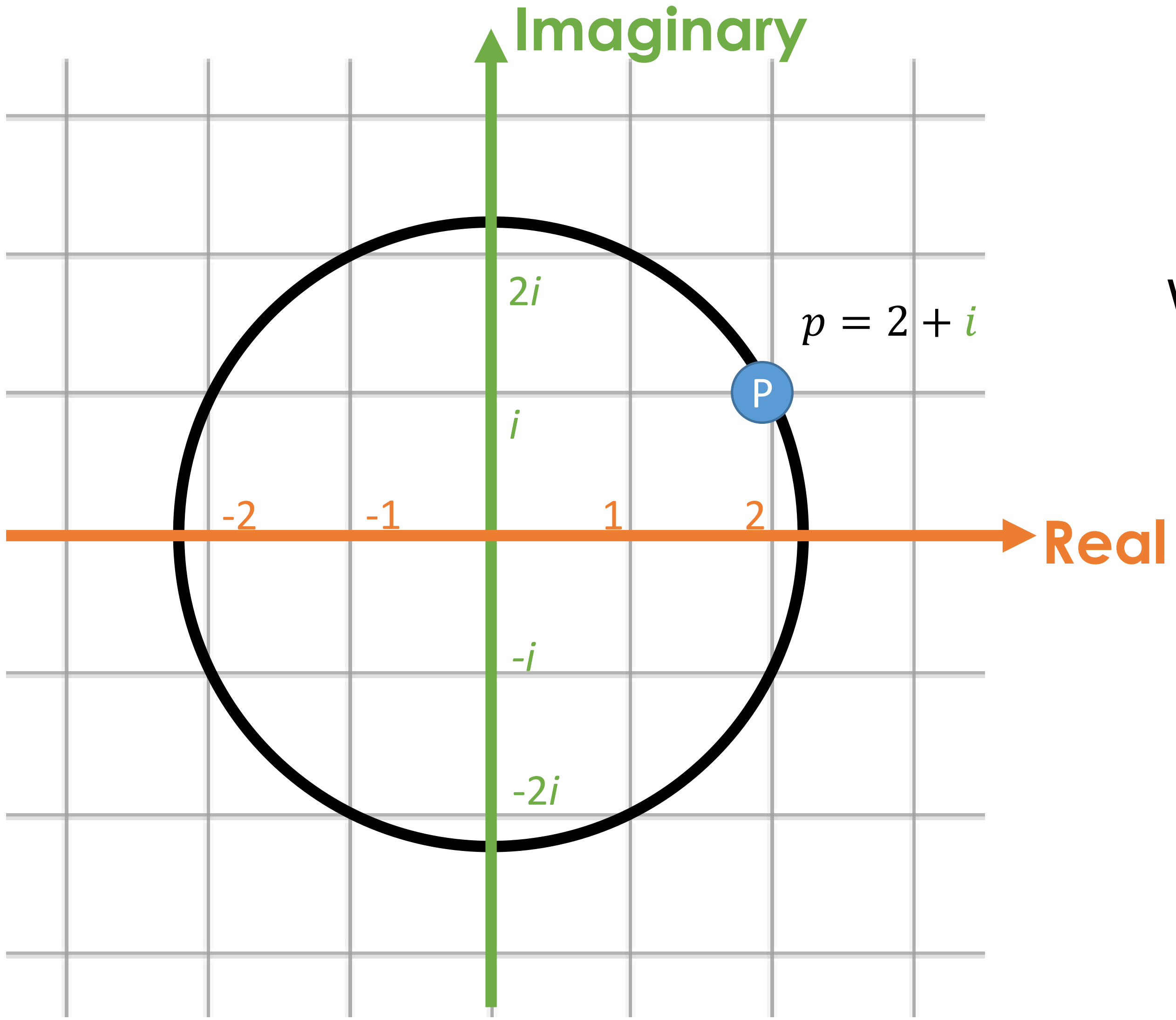
Multiplying a complex number by *i* rotates through this plane at 90 degree increments





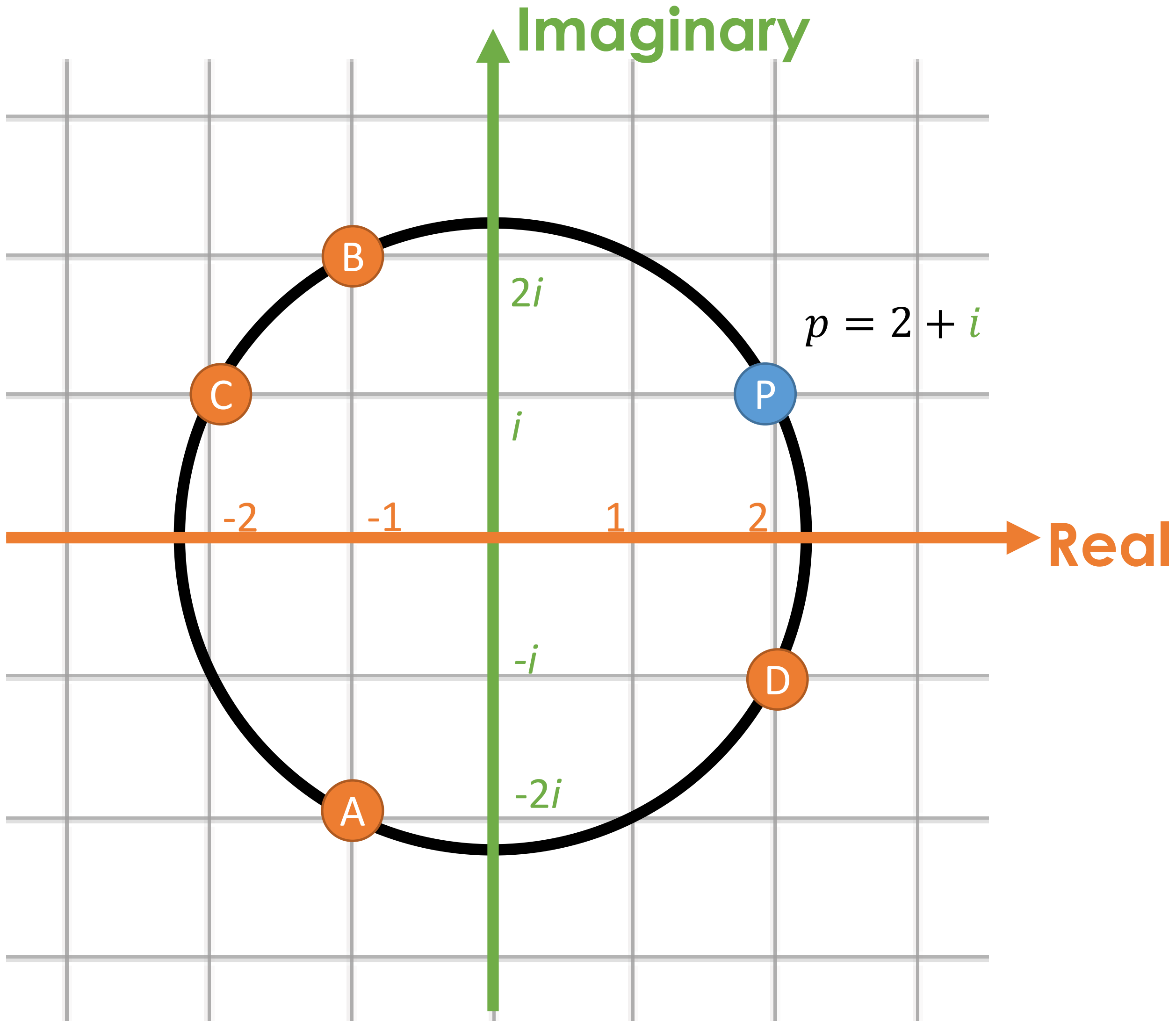
Point $p = 2 + i$

Where would you draw point p?



Point $p = 2 + i$

Where would you draw point p?

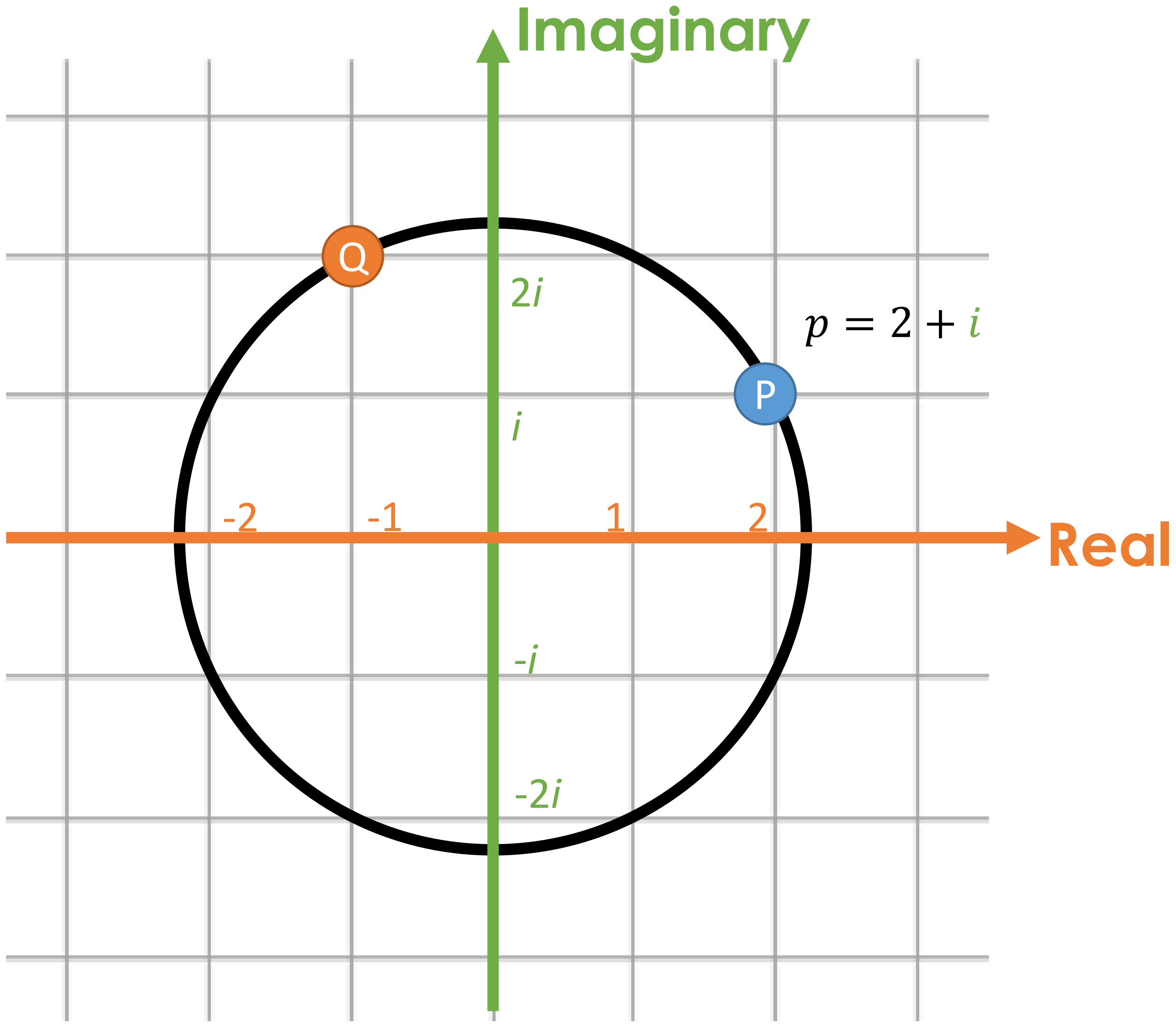


Point $p = 2 + i$

Let's multiply p by i:
Point $q = pi$

$$
\begin{aligned}
&= (2+i)i \\
&= 2i + i^2 \\
&= -1 + 2i
\end{aligned}
$$

Where is point q?

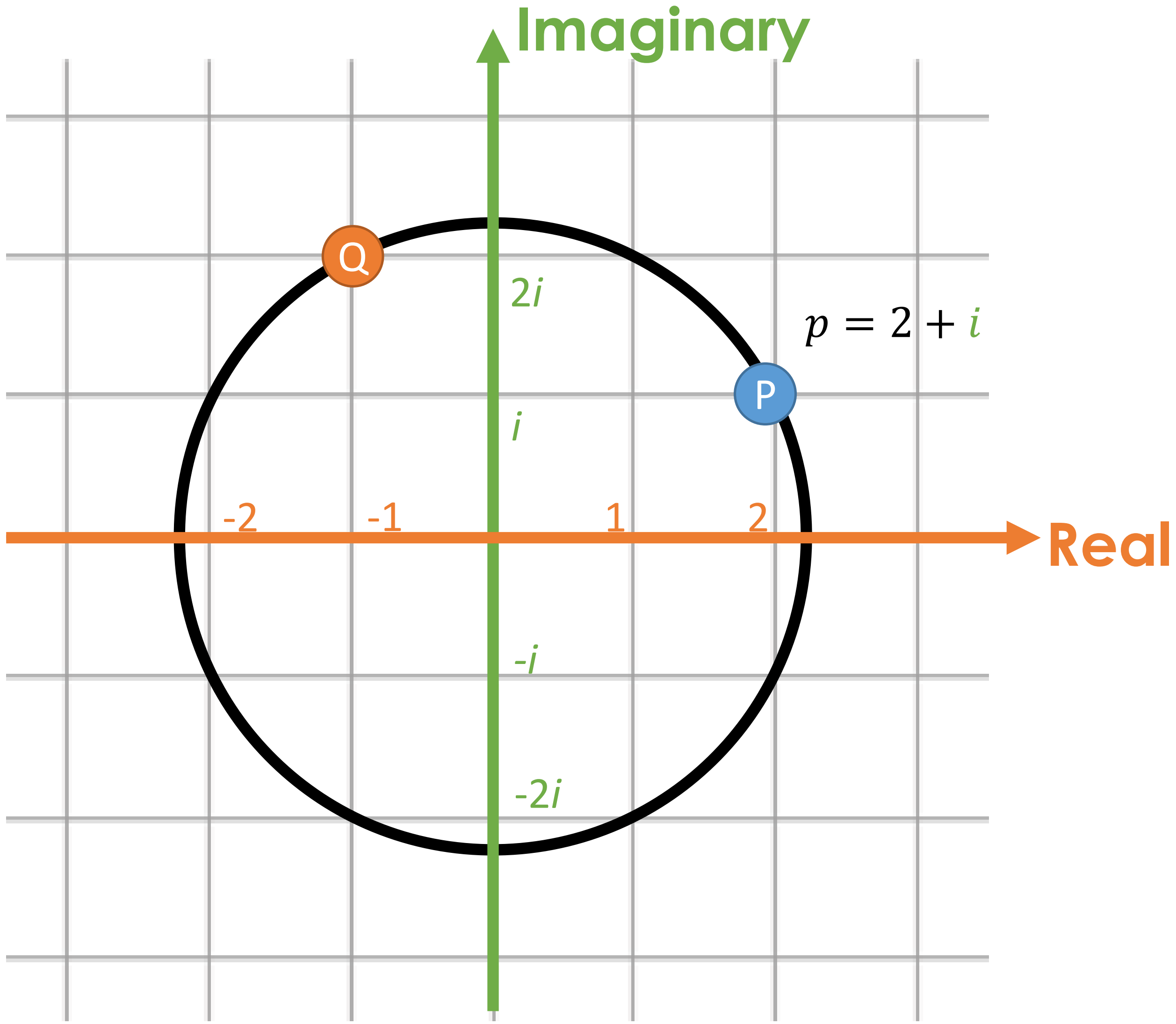


Point $p = 2 + i$

Let's multiply p by i:
Point $q = pi$

$$\begin{aligned} &= (2+i)i \\ &= 2i + i^2 \\ &= -1 + 2i \end{aligned}$$

Where is point q?



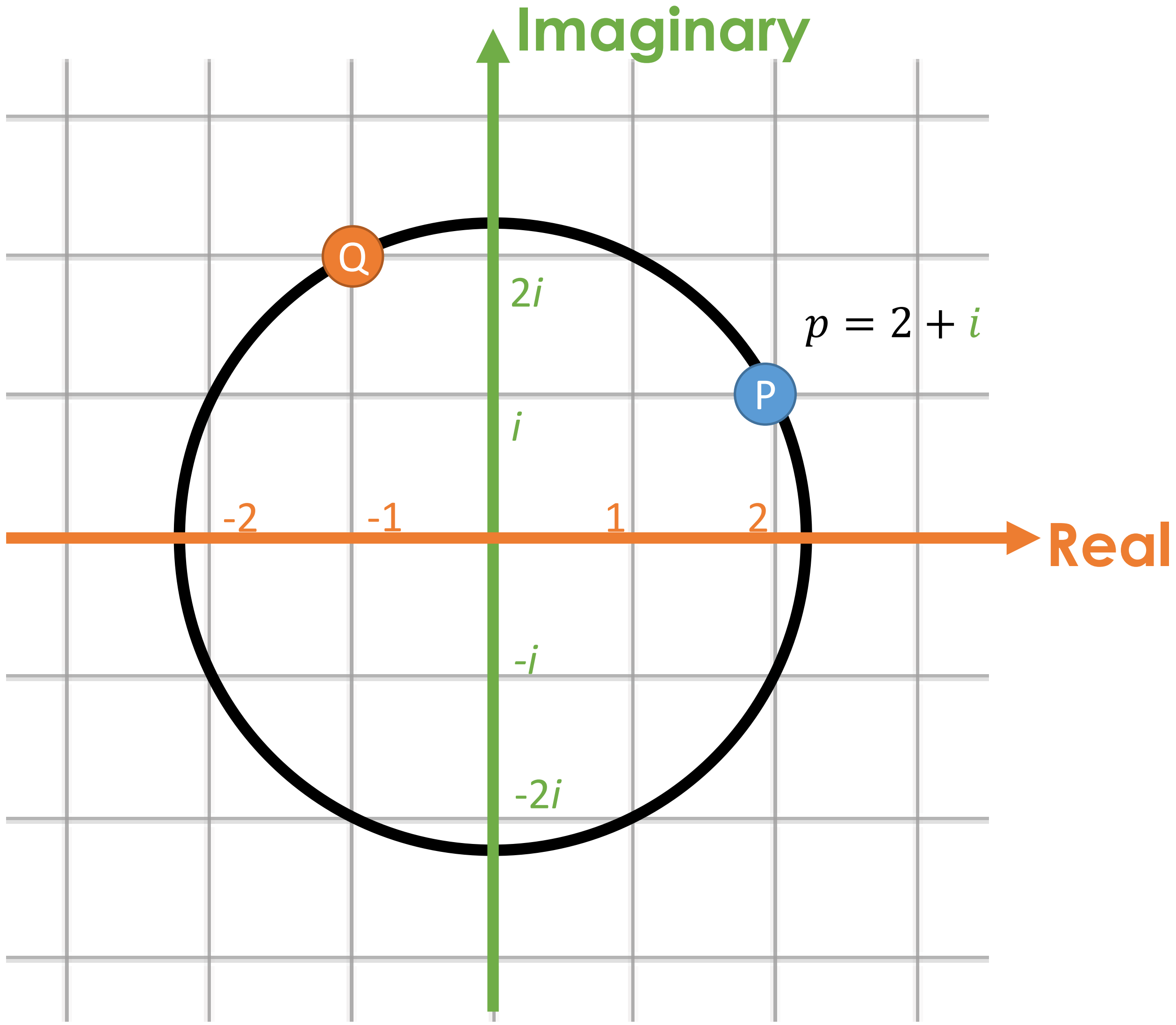
Point $p = 2 + i$
Point $q = -1 + 2i$

Let's multiply q by i:
Point $r = qi$

What is the formula for r?

A) $r = -1 + 2i$
B) $r = 2 + i$
C) $r = -1 - i$
D) $r = -2 - i$

Where is point r?



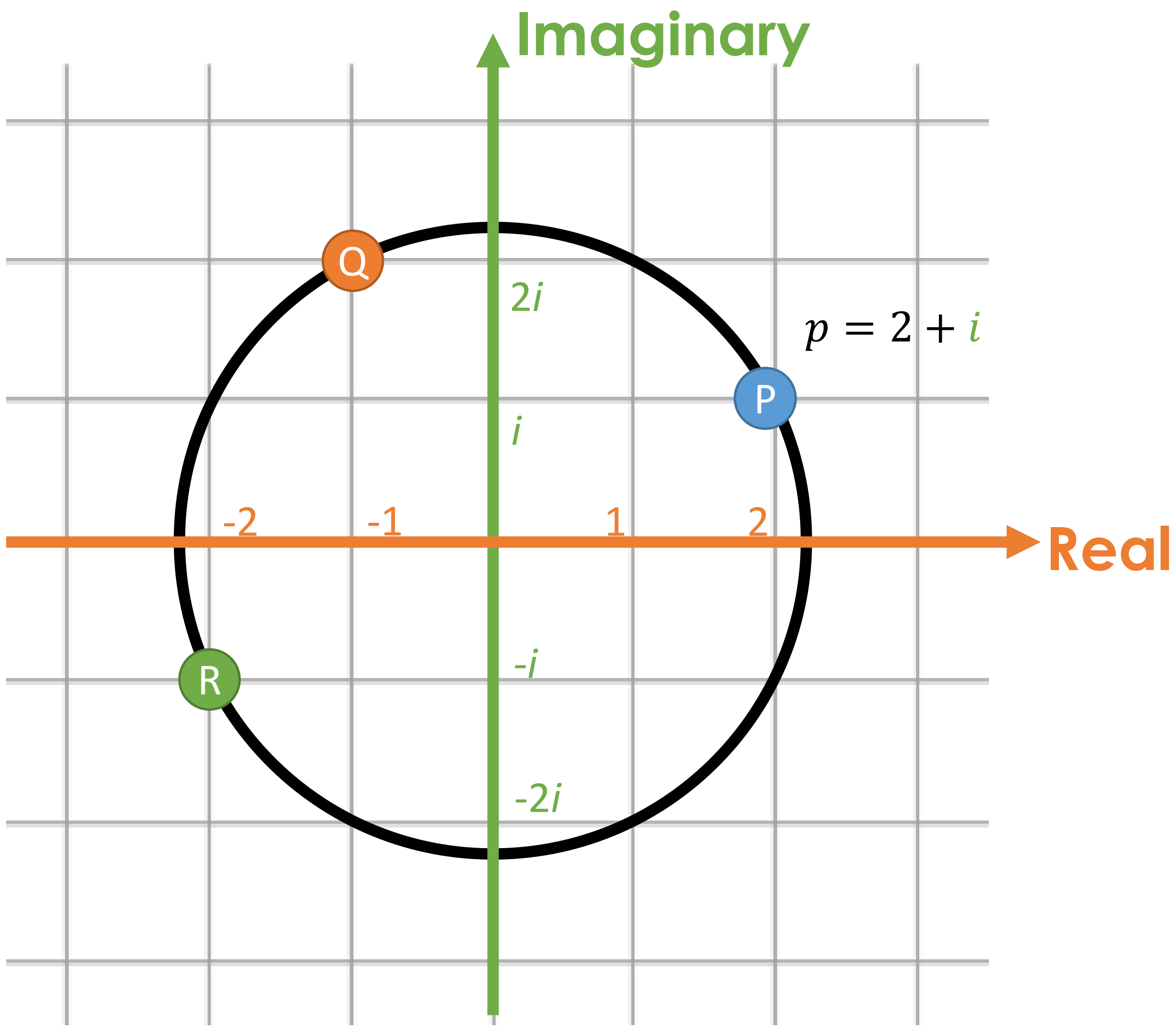
Point $p = 2 + i$
Point $q = -1 + 2i$

Let's multiply q by i:
Point $r = qi$

What is the formula for r?

$$\begin{aligned} r &= qi \\ &= (-1+2i)i \\ &= -i + 2i^2 \\ &= -2 - i \end{aligned}$$

Where is point r?



Point $p = 2 + i$

Point $q = -1 + 2i$

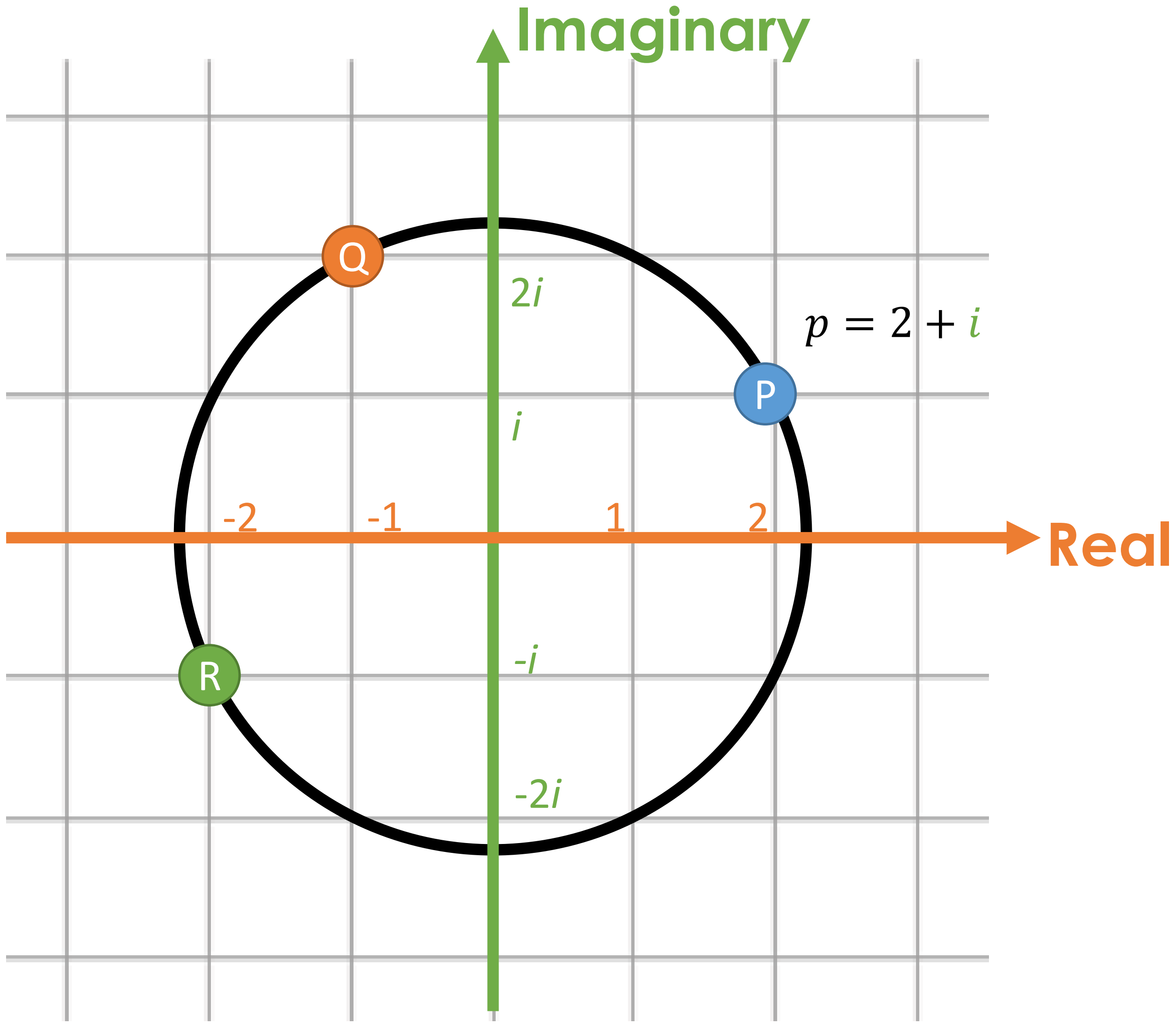
Let's multiply q by i:

Point $r = qi$

What is the formula for r?

$$\begin{aligned} r &= qi \\ &= (-1 + 2i)i \\ &= -i + 2i^2 \\ &= -2 - i \end{aligned}$$

Where is point r?



Point $p = 2 + i$
Point $q = -1 + 2i$
Point $r = -2 - i$

Let's multiply *r* by *i*:
Point $s = ri$

What is the formula for *s*?

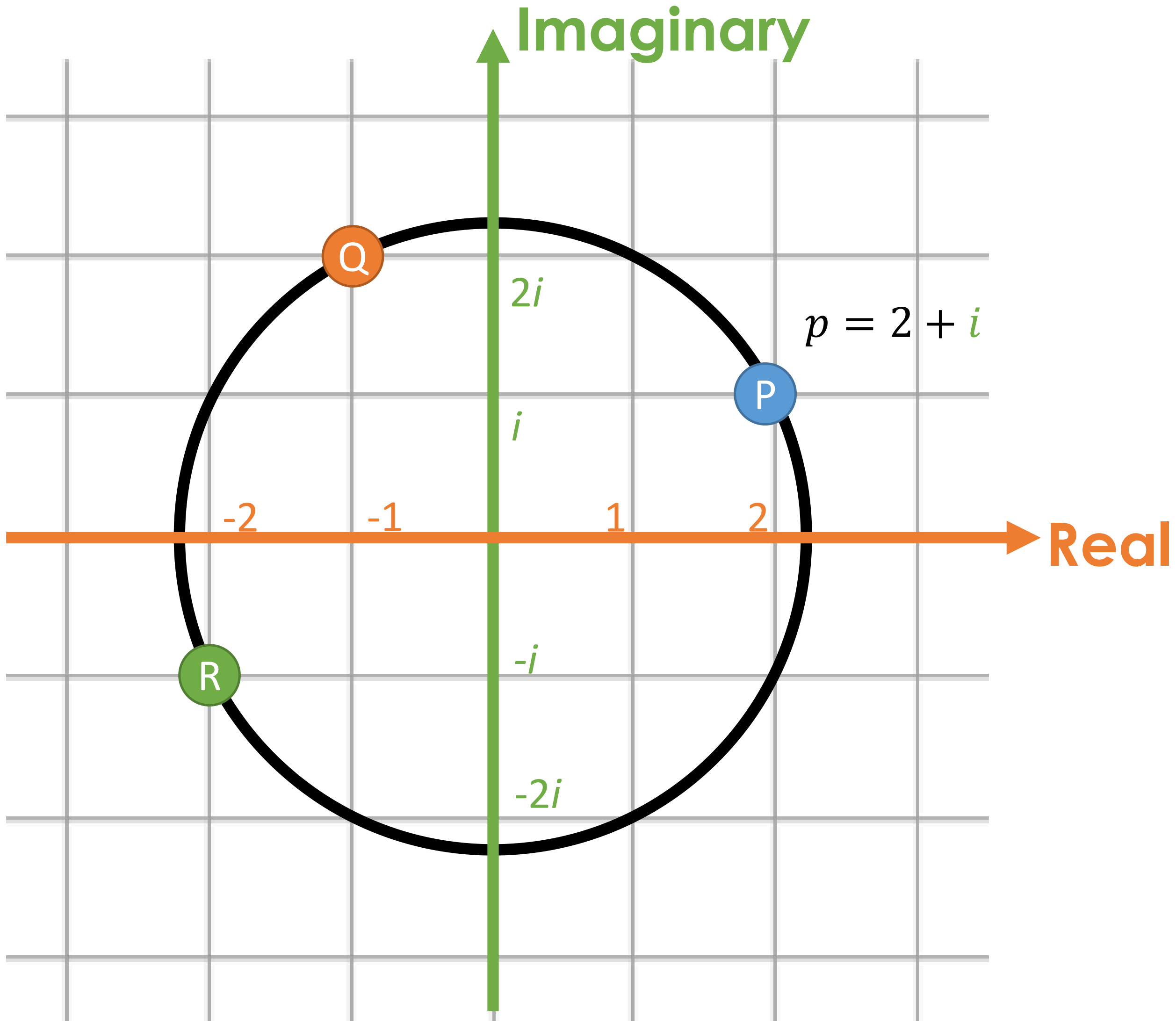
$A)\ s = -1 + 2i$

$B)\ s = 1 + i$

$C)\ s = 1 - i$

$D)\ s = 1 - 2i$

Where is point *s*?



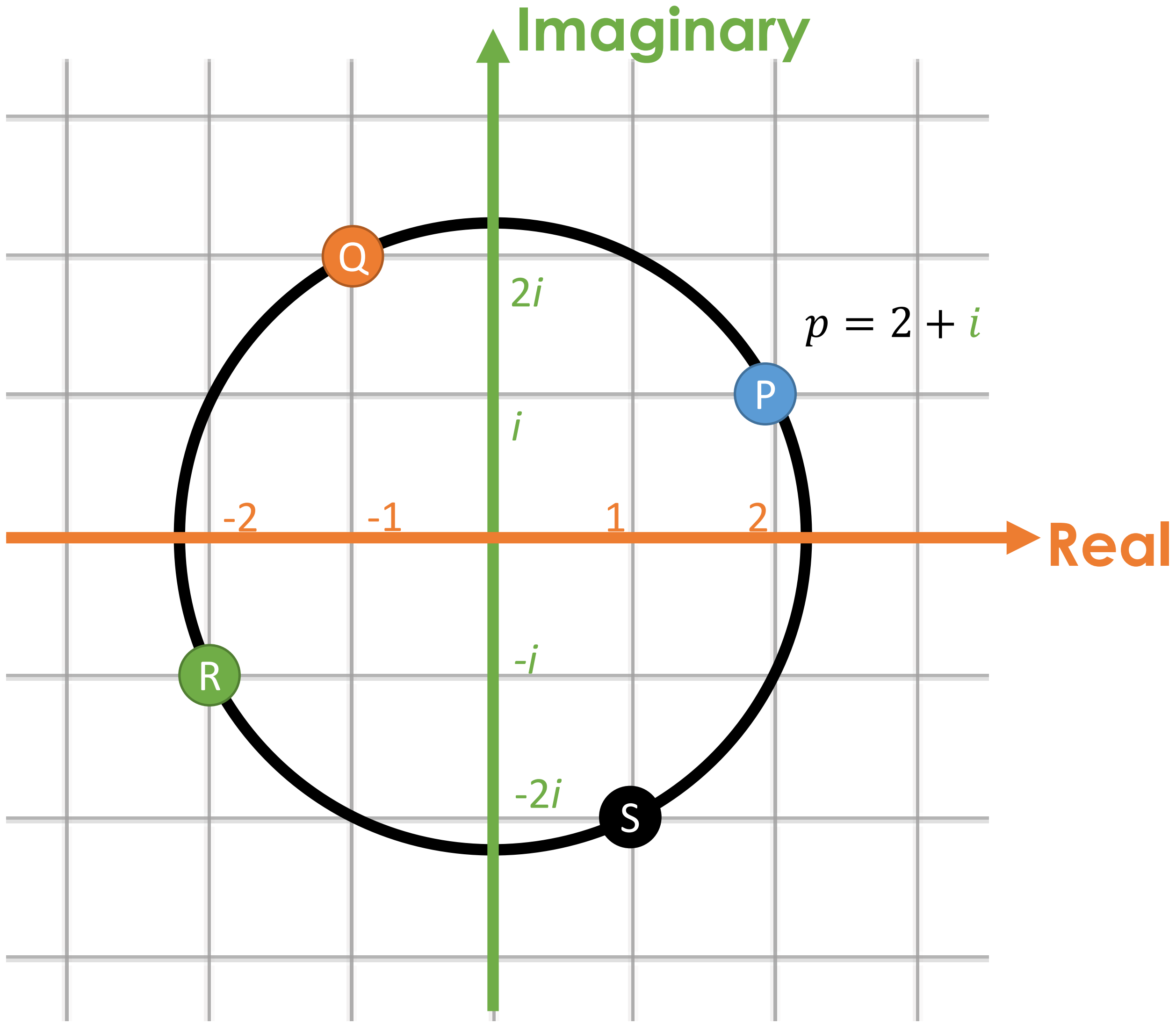
Point $p = 2 + i$
Point $q = -1 + 2i$
Point $r = -2 - i$

Let's multiply *r* by *i*:
Point $s = ri$

What is the formula for *s*?

$$\begin{aligned} s &= ri \\ &= (-2 - i)i \\ &= -2i - i^2 \\ &= 1 - 2i \end{aligned}$$

Where is point *s*?



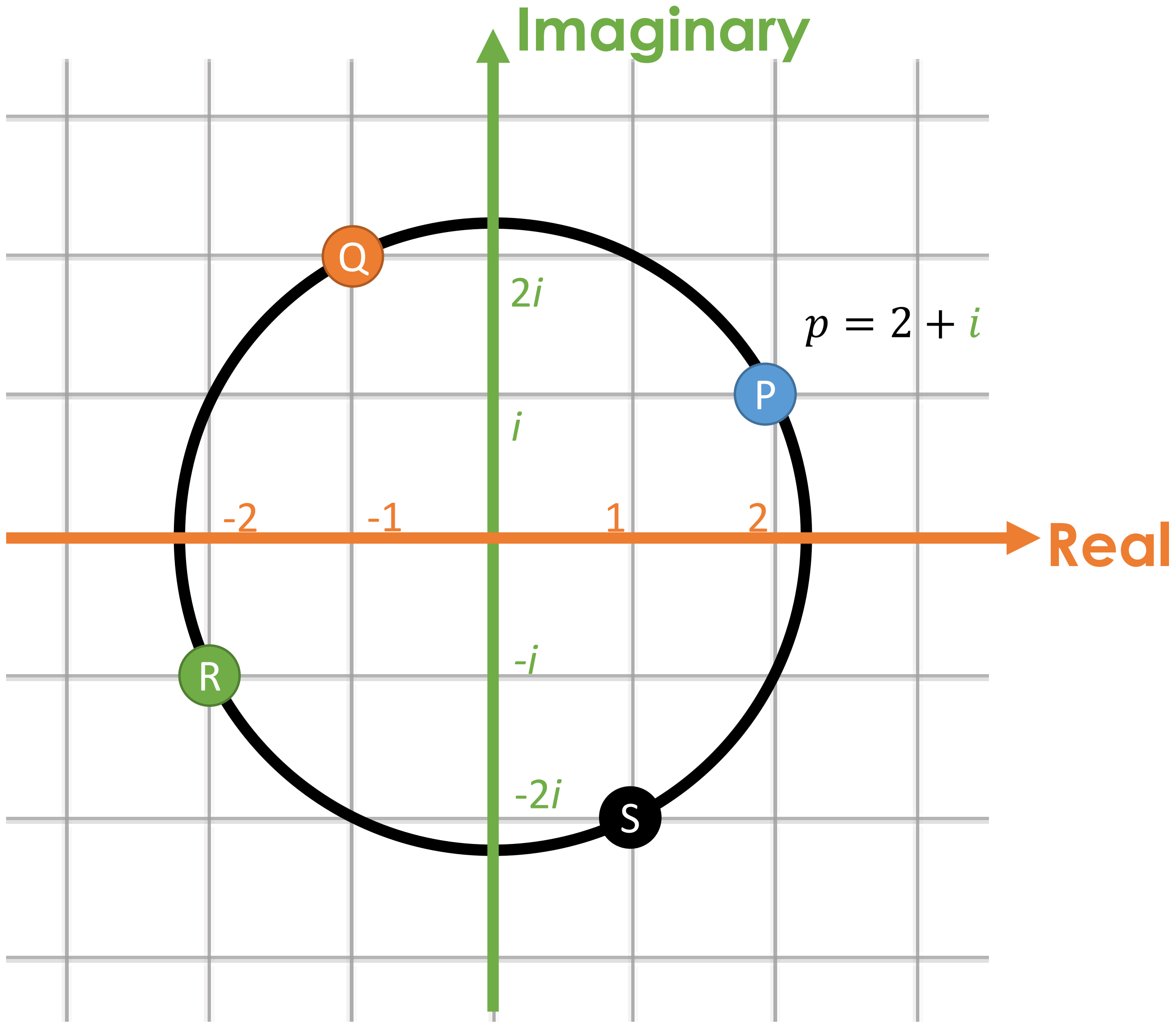
Point $p = 2 + i$
Point $q = -1 + 2i$
Point $r = -2 - i$

Let's multiply *r* by *i*:
Point $s = ri$

What is the formula for *s*?

$$\begin{aligned} s &= ri \\ &= (-2 - i)i \\ &= -2i - i^2 \\ &= 1 - 2i \end{aligned}$$

Where is point *s*?



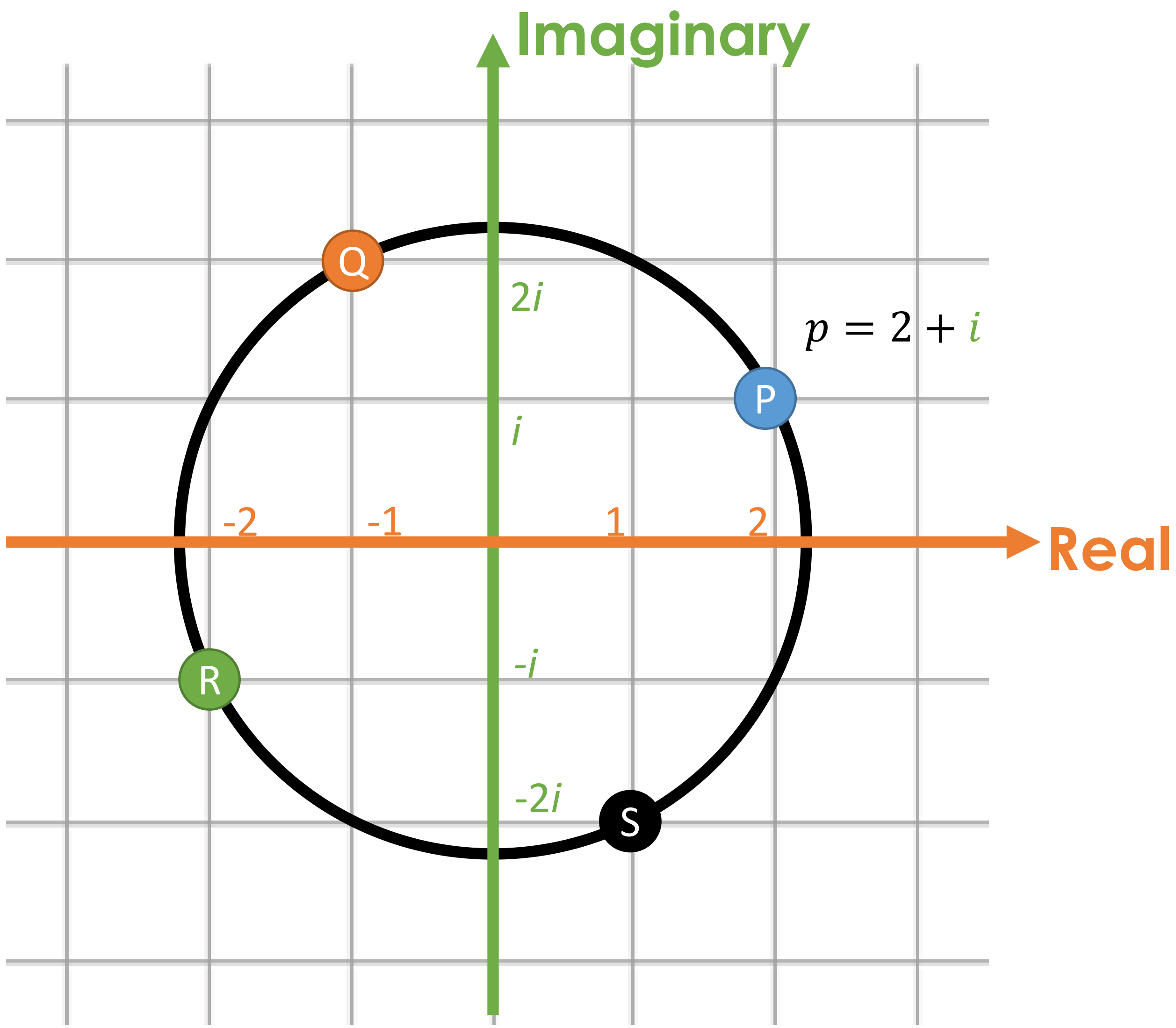
Point $p = 2 + i$
Point $q = -1 + 2i$
Point $r = -2 - i$
Point $s = 1 - 2i$

Let's multiply *s* by *i*:
Point $t = si$

What is the formula for *t*?

$A)\ t = 1 + 2i$
$B)\ t = 1 + i$
$C)\ t = 2 + i$
$D)\ t = -1 + 2i$

Where is point *t*?



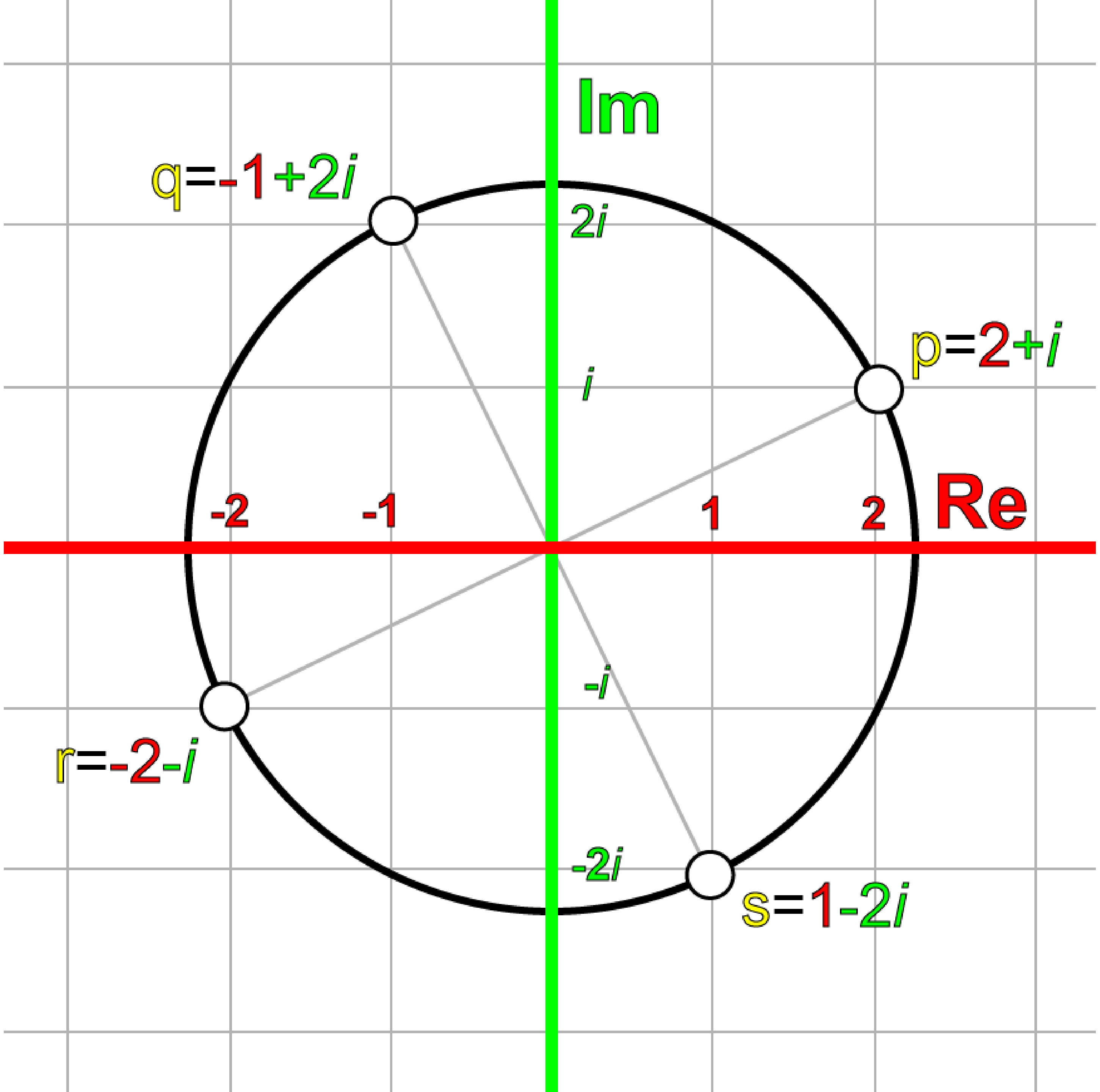
Point $p = 2 + i$
Point $q = -1 + 2i$
Point $r = -2 - i$
Point $s = 1 - 2i$

Let's multiply *s* by *i*:
Point $t = si$

What is the formula for *t*?

$$\begin{aligned} t &= si \\ &= (1-2i)i \\ &= i - 2i^2 \\ &= 2 + i \end{aligned}$$

Where is point *t*?



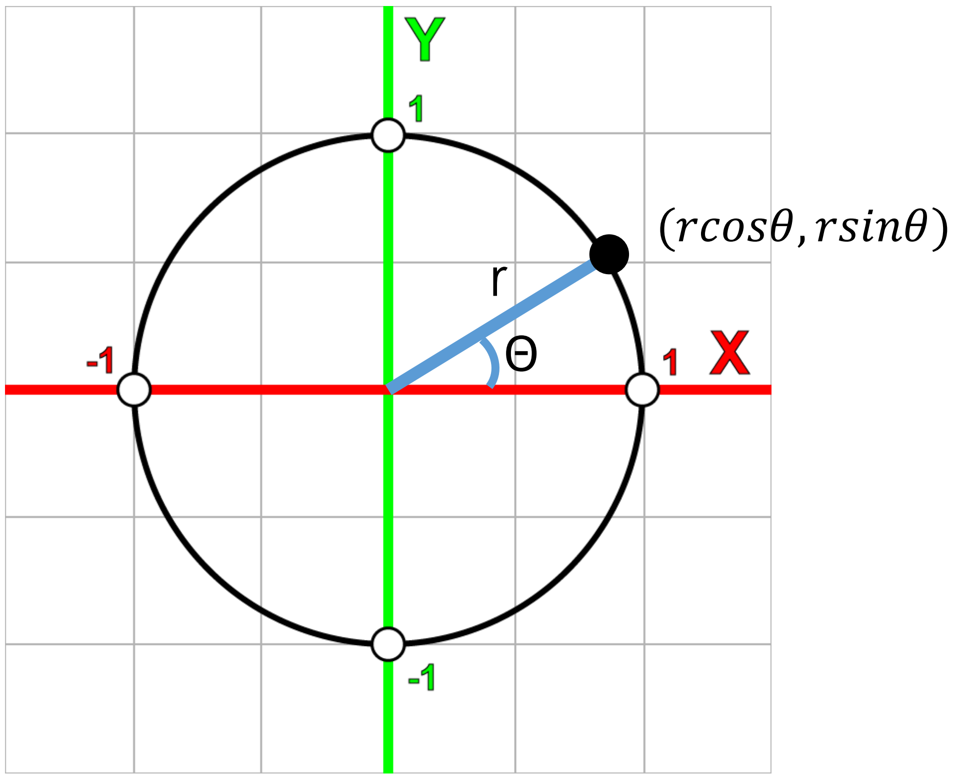
Multiplying a complex number by *i* rotates through this plane at 90 degree increments

How would we rotate clock-wise?

What if we wanted arbitrary rotation angles?

Arbitrary Rotations

Cartesian Plane: use polar coordinates

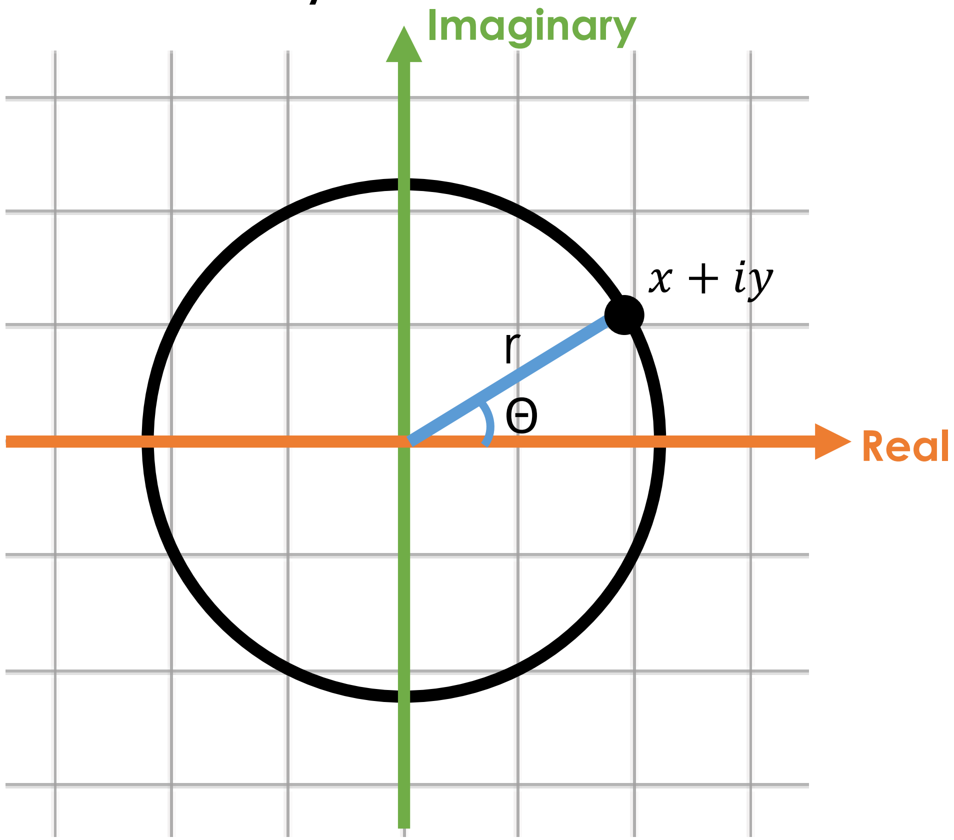


Polar coordinates:

$$x = rcos\theta$$
$$y = rsin\theta$$

Arbitrary Rotations

Complex Plane: define a "rotor" *q* (same idea as polar coords)



Point P = $x + iy = rcos\theta + irsin\theta$

Rotor $q = cos\theta + isin\theta$

Rotated point P' = pq

Multiplying a point by a rotor:

$$
\begin{aligned}
p &= a + bi \\
q &= \cos\theta + i\sin\theta \\
pq &= (a+bi)(\cos\theta + i\sin\theta) \\
a' + b'i &= a\cos\theta - b\sin\theta + (a\sin\theta + b\cos\theta)i
\end{aligned}
$$

An Aside

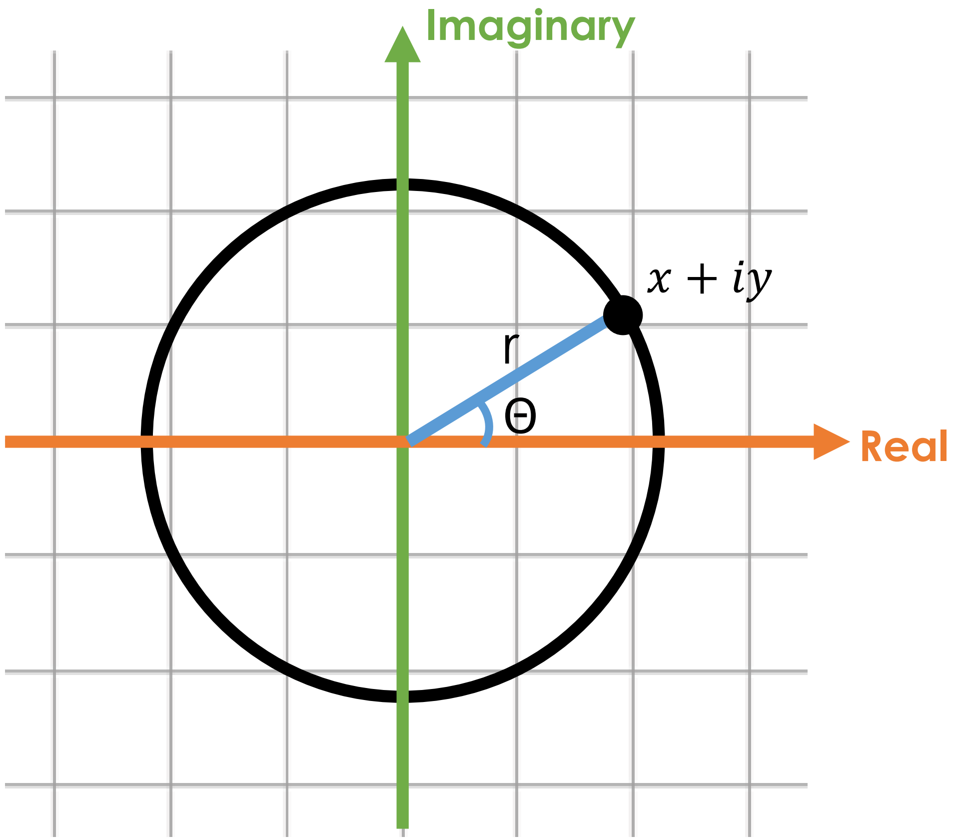
Point P = $x + iy = rcos\theta + irsin\theta$

Euler's formula: $e^{ix} = cosx + isinx$

$\therefore P = re^{i\theta}$

Where is $e^{\pm i\pi/2}$?

Euler's identity: $e^{i\pi} + 1 = 0$



Stretch!

Stand up

Use your left hand

Real axis points forward

Imaginary axis points up

Show:

- 1
- $2i$
- $1+i$
- $e^{-i\pi/3}$

Test your knowledge

If $re^{i\theta}$ is multiplied by i, the corresponding vector is:

A) Reflected about the x-axis

B) Reflected about the y-axis

C) Rotated 90 degrees counterclockwise

D) Rotated 90 degrees clockwise

Test your knowledge 2

$$(re^{i\theta})(se^{i\alpha})$$

If $re^{i\theta}$ is multiplied by $se^{i\alpha}$, the corresponding vector is?

Get with a partner, write **big** on a blank sheet of paper (or the back of your plicker card). We will hold up answers when finished.

$$\left(re^{i\theta}\right)\left(se^{i\alpha}\right) = rse^{i(\theta+\alpha)}$$

In other words

- Multiplying by *i* = rotate counterclockwise by 90 degrees
- Multiplying by $se^{i\alpha}$ rotates counterclockwise by α and scales by *s*

Extending to 3D Space: Quaternions

Add 2 additional imaginary numbers to our number system

- *i, j, k* are all roots of -1

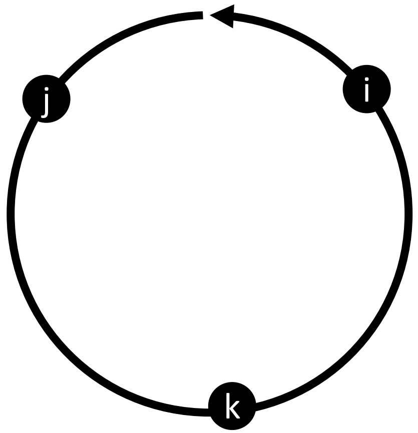
Quaternion general form: $q = s + xi + yj + zk$ where

- $s, x, y, z \ \in \mathbb{R}$
- i.e., *s, x, y*, and *z* are just normal numbers (*x,y,z* correspond to our normal *x,y,z* axis)
- Hamilton's equation: $i^2 = j^2 = k^2 = ijk = -1$

Quaternions

Anti-commutative:

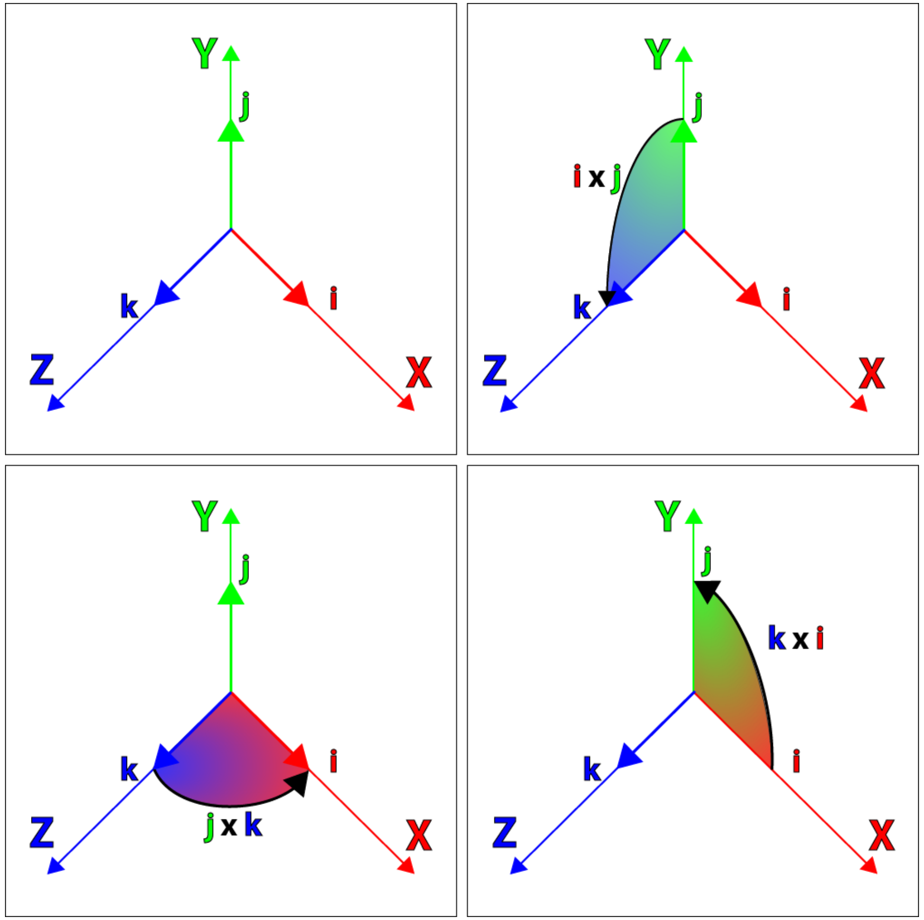
$$\begin{array}{lll} ij = k & jk = i & ki = j \\ ji = -k & kj = -i & ik = -j \end{array}$$

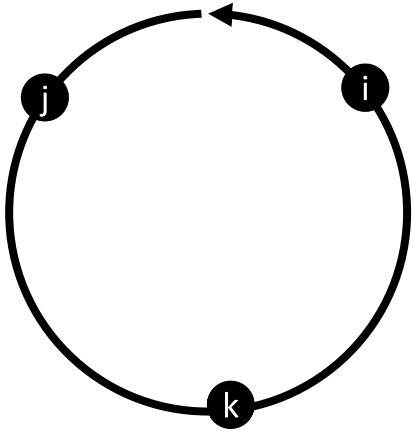


Note similarity to Cartesian cross product:

$$\begin{array}{lll} \mathbf{x} \times \mathbf{y} = \mathbf{z} & \mathbf{y} \times \mathbf{z} = \mathbf{x} & \mathbf{z} \times \mathbf{x} = \mathbf{y} \\ \mathbf{y} \times \mathbf{x} = -\mathbf{z} & \mathbf{z} \times \mathbf{y} = -\mathbf{x} & \mathbf{x} \times \mathbf{z} = -\mathbf{y} \end{array}$$

Visualizing i,j,k





What happens if we multiply a quaternion by *i*?

$q = s + xi + yj + zk$
$iq = ?$
$qi = ?$

$$\begin{array}{ccc} ij = k & jk = i & ki = j \\ ji = -k & kj = -i & ik = -j \end{array}$$

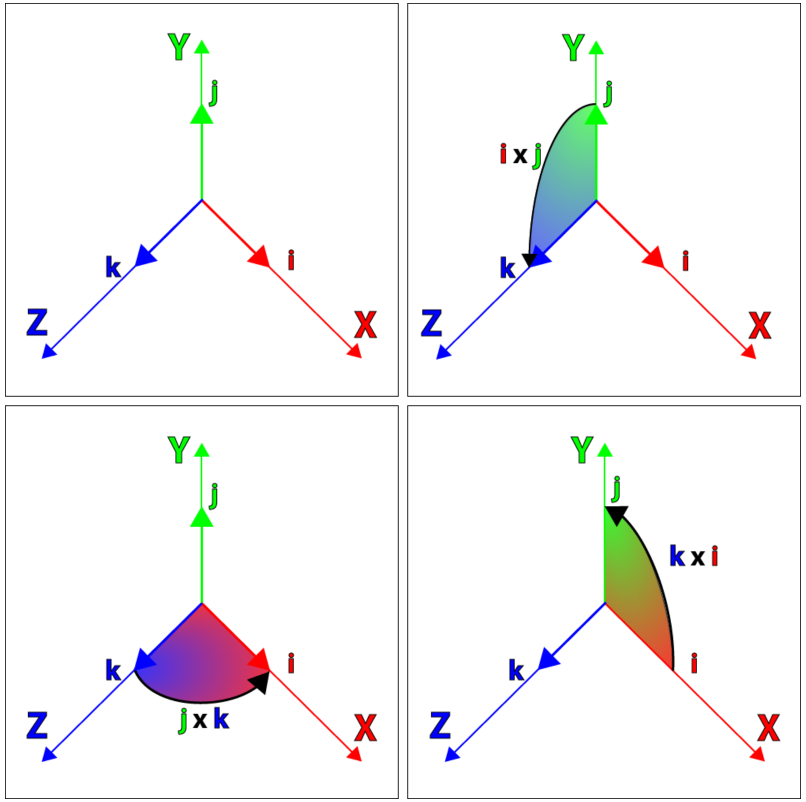
What happens if we multiply a quaternion by *i*?

$$q = s + xi + yj + zk$$
$$iq = si - x + yk - zj$$
$$qi = si - x - yk + zj$$

$$ij = k \quad jk = i \quad ki = j$$
$$ji = -k \quad kj = -i \quad ik = -j$$

$$iqi = -s - xi + yj + zk$$
$$-iqi = s + xi - yj - zk$$

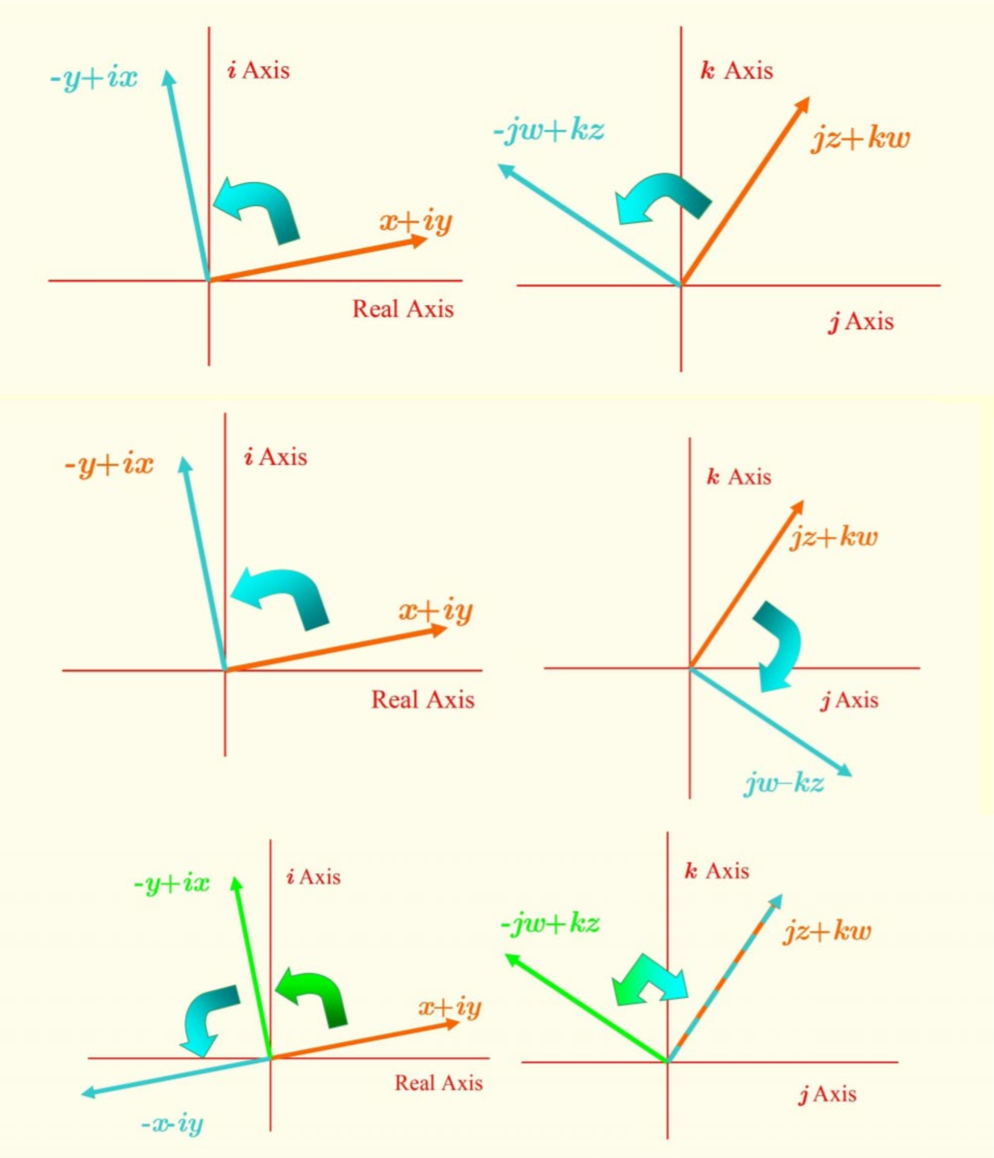
(rotation in the *jk-plane*)



Visualization

-iqi rotates by Θ "about i" (in the *jk*-plane)

Similar methods for rotating through the other planes



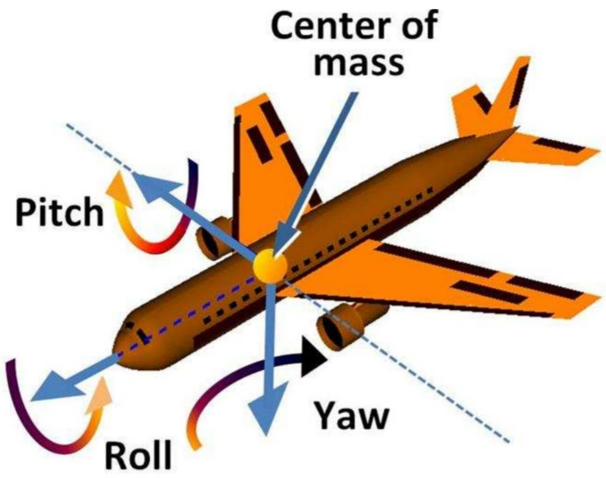
Visualization

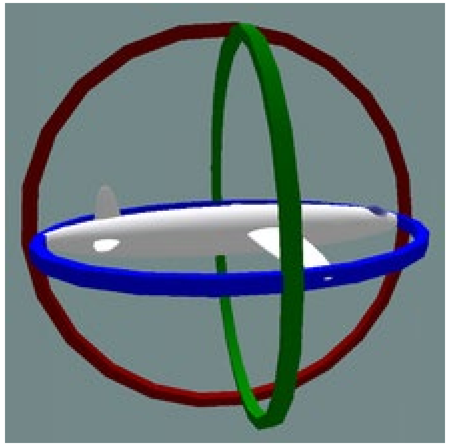
http://quaternions.online/

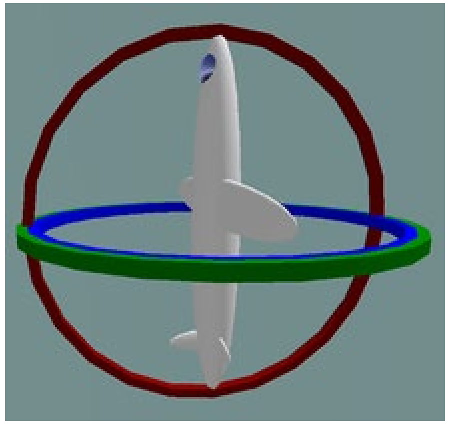
Why are quaternions helpful?

Interpolation!

Can avoid gimbal lock







SLERP

Spherical **L**inear Int**erp**olation

- Allows us to smoothly interpolate between two orientations with ease in/out
- q_1 = starting orientation
- q_2 = end orientation
- t = time (how fast the interpolation happens) – ranges [0,1]
- p = start point
- p' = end point

LERP: $p' = p_1 + t(p_2 - p_1)$

SLERP: same idea, but we are interpolating across the 4D sphere formed by the quaternion

Unity Demo

For more info: https://docs.unity3d.com/Manual/QuaternionAndEulerRotationsInUnity.html

THANKS!

Professor **Dan Szafir**

Computer Science & ATLAS Institute
University of Colorado Boulder