Week 5: Mechanics II
Part 1: When Objects Collide

COMP270: Mathematics for 3D Worlds and Simulations

Objectives

- **Consider** what it means for two objects to collide
- **Implement** methods for detecting whether two objects are in collision

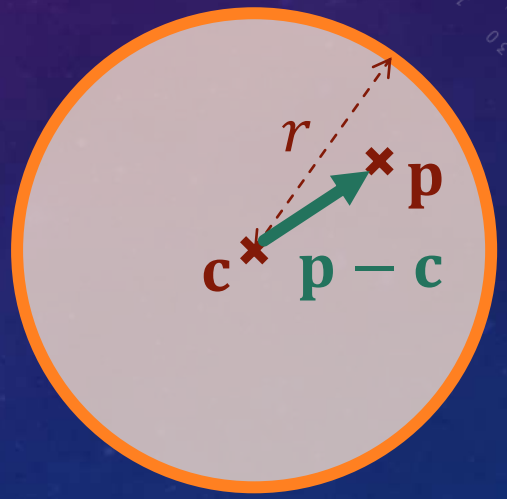
What is a collision?

- A **collision** occurs when two objects occupy the same point in space at the same time (or try to)

Point and circle collision

- Consider a circle with centre $\mathbf{c}$ and radius r
- A point $\mathbf{p}$ is inside the circle if and only if the distance between $\mathbf{p}$ and $\mathbf{c}$ is at most r:

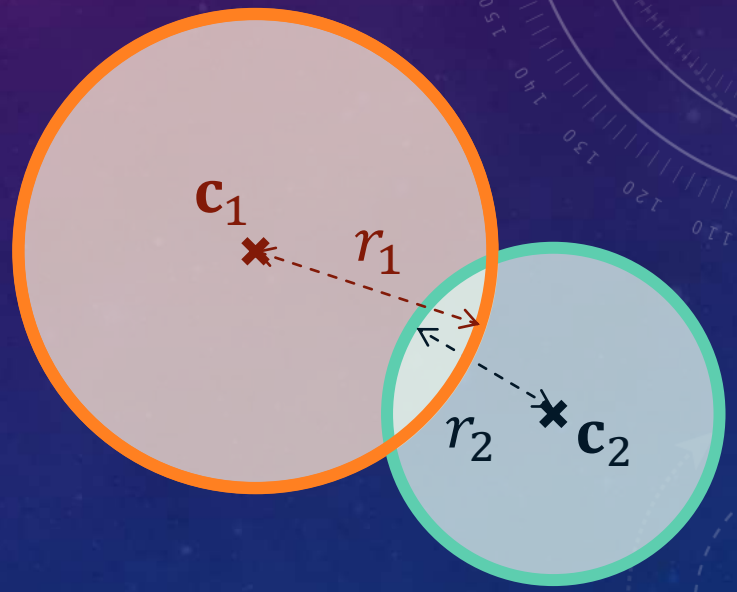
$$\|\mathbf{p} - \mathbf{c}\| \leq r$$



Circle and circle collision

- Consider two circles with centres $\mathbf{c}_1, \mathbf{c}_2$ and radii r_1, r_2
- The circles overlap (collide) if and only if

$$\|\mathbf{c}_1 - \mathbf{c}_2\| \leq r_1 + r_2$$

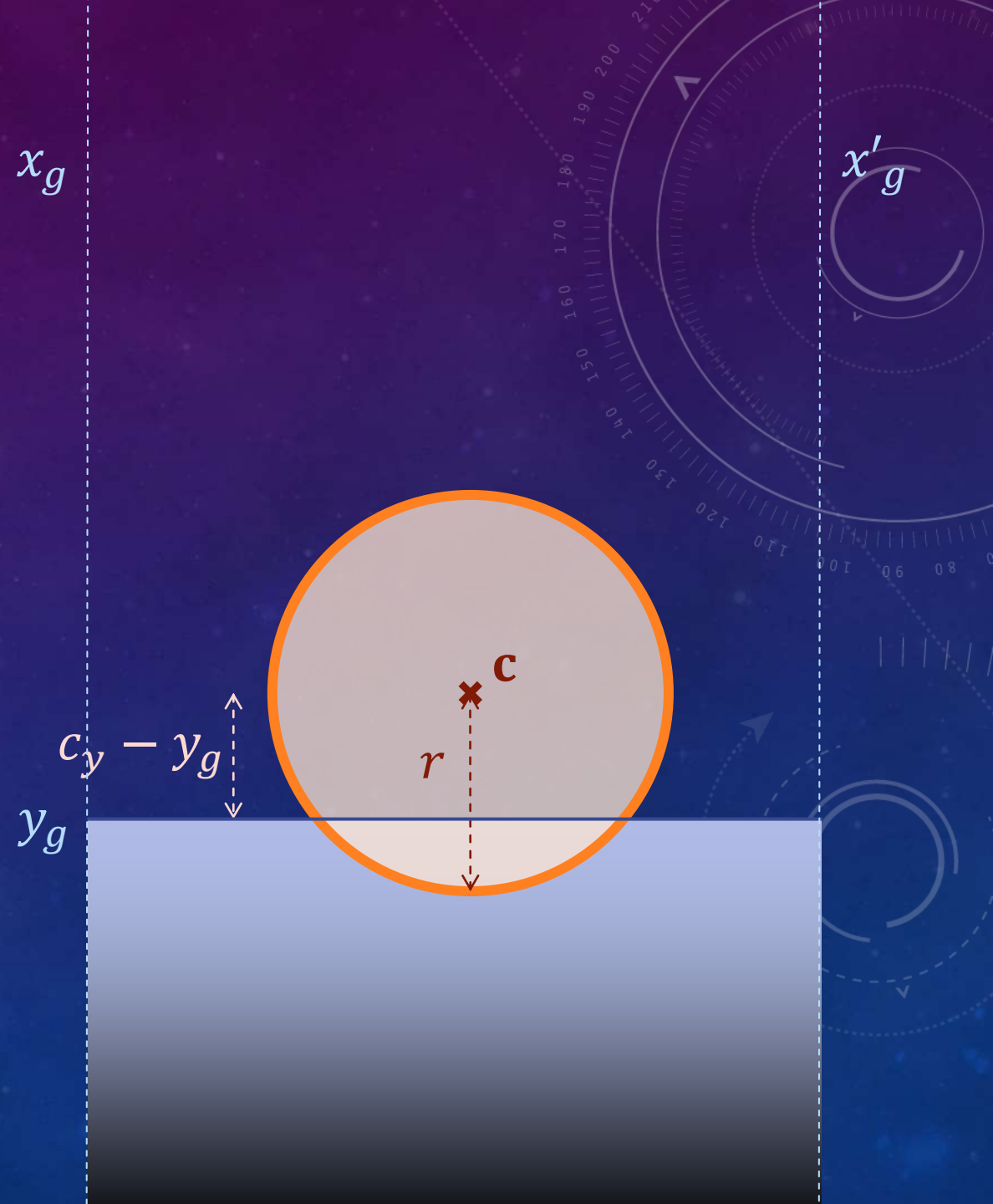


Circle and ground collision

- Consider a circle with centre $\mathbf{c} = \begin{pmatrix} c_x \\ c_y \end{pmatrix}$ and radius r
- Let y_g be the y coordinate of the ground, and let the ground be horizontal
- The circle collides with the ground if and only if

$$c_y - y_g \leq r$$

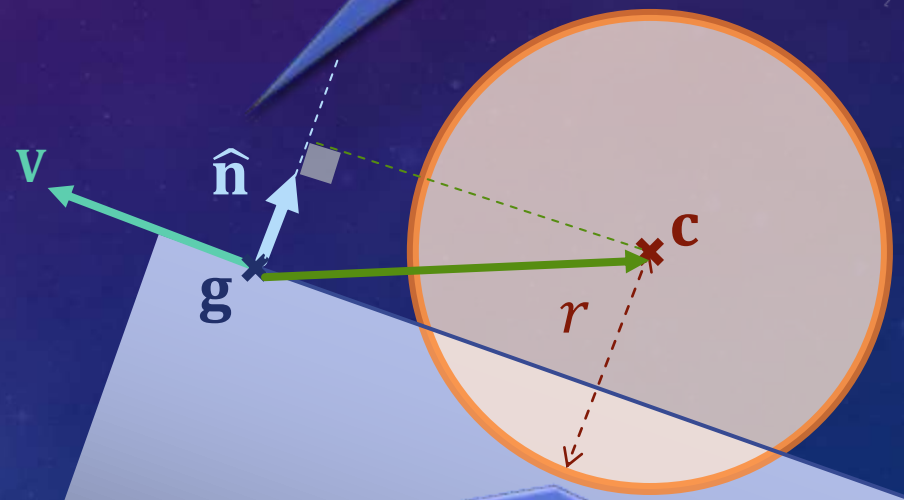
- (and $x_g \leq c_x \leq x'_g$)



Generalised circle and ground collision

- Let $\hat{\mathbf{n}}$ be a **normal vector** (a unit vector perpendicular to the ground)
- Let $\mathbf{g}$ be any point on the ground
- The distance from $\mathbf{c}$ to the ground is the projection of its offset from $\mathbf{g}$ onto the normal:
$$(\mathbf{c} - \mathbf{g}) \cdot \hat{\mathbf{n}}$$
- Therefore the circle collides with the ground if and only if
$$(\mathbf{c} - \mathbf{g}) \cdot \hat{\mathbf{n}} \leq r$$

$$\hat{\mathbf{n}} \cdot \mathbf{v} = \mathbf{0}$$
$$\Rightarrow n_y = -\frac{v_x}{v_y} n_x$$
$$\Rightarrow \mathbf{n} = k \begin{pmatrix} 1 \\ -\frac{v_x}{v_y} \end{pmatrix}$$

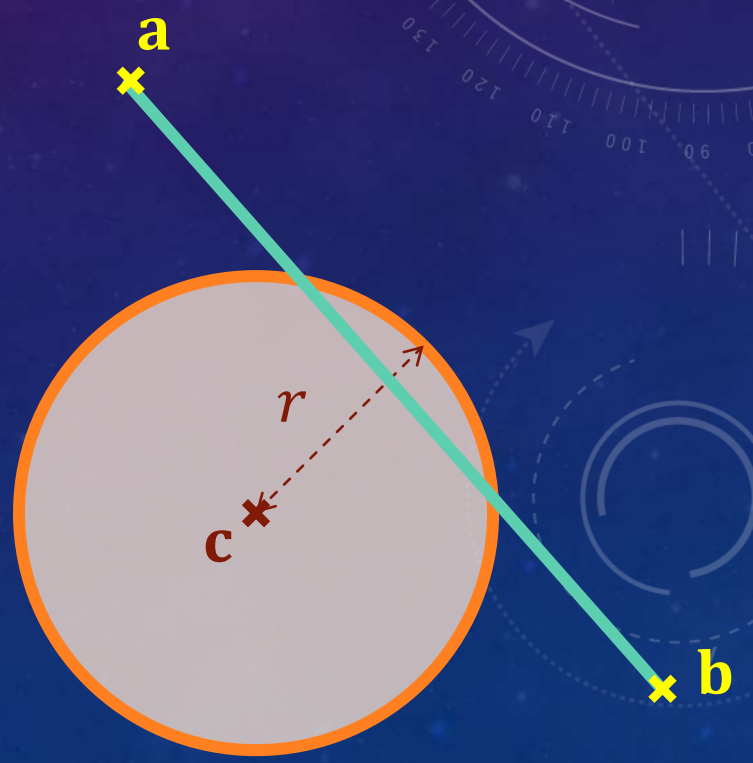


If $\mathbf{c} - \mathbf{g}$ is parallel to $\hat{\mathbf{n}}$:
$$\mathbf{c} - \mathbf{g} = \|\mathbf{c} - \mathbf{g}\| \hat{\mathbf{n}}$$

$$(\mathbf{c} - \mathbf{g}) \cdot \hat{\mathbf{n}} = \|\mathbf{c} - \mathbf{g}\| \hat{\mathbf{n}} \cdot \hat{\mathbf{n}} = \|\mathbf{c} - \mathbf{g}\|$$

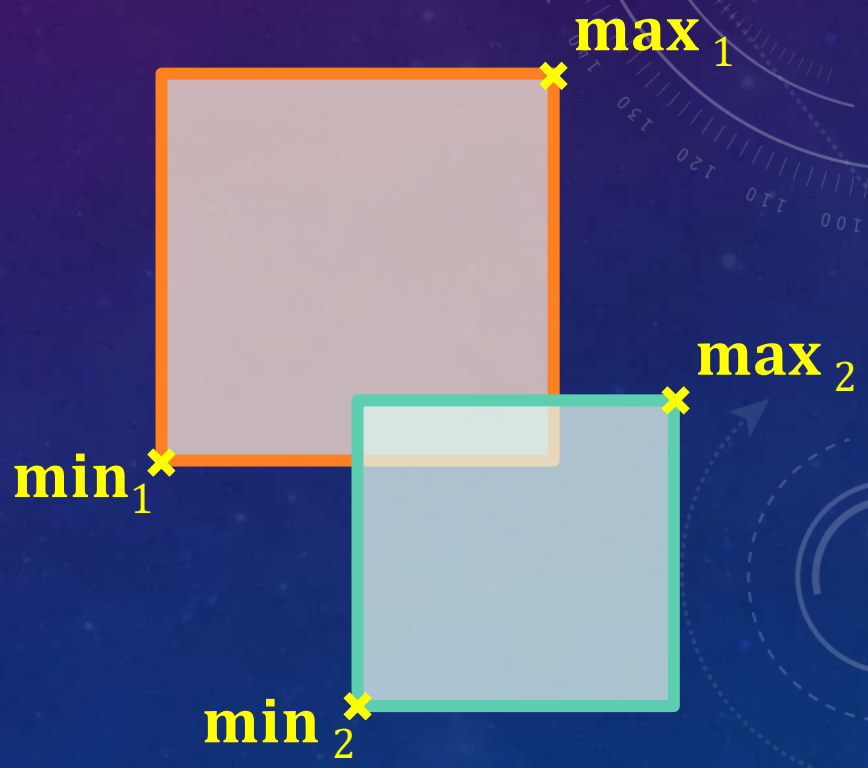
Circle and line segment collision

- Collisions with lines or line segments is the basis of **raycasting**
- Consider a circle with centre $\mathbf{c}$ and radius r, and a line segment from point $\mathbf{a}$ to point $\mathbf{b}$
- The two collide if and only if the shortest distance between $\mathbf{c}$ and the line is $\leq r$



Box and box collision

- Consider two axially aligned boxes (i.e. rectangles) defined by maximum and minimum vertices
- The boxes overlap (collide) if and only if at least one of them has a minimum coordinate not greater than the corresponding maximum coordinate



$$\neg(x_{min_1} > x_{max_2} \vee x_{min_2} > x_{max_1} \vee y_{min_1} > y_{max_2} \vee y_{min_2} > y_{max_1})$$

Box and box collision - code

```
bool boxesCollide(Vector2 box1_min, Vector2 box1_max,
                  Vector2 box2_min, Vector2 box2_max) {
   if (box1_min.x > box2.max.x) return false;
   if (box1_max.x < box2.min.x) return false;
   if (box1_min.y > box2.max.y) return false;
   if (box1_max.y < box2.min.y) return false;
   return true;
}
```

More complex shapes

- General collision detection is beyond the scope of this module
- Algorithms and libraries do exist
- Basic idea: two shapes collide if at least one point (or edge) of one is inside the other