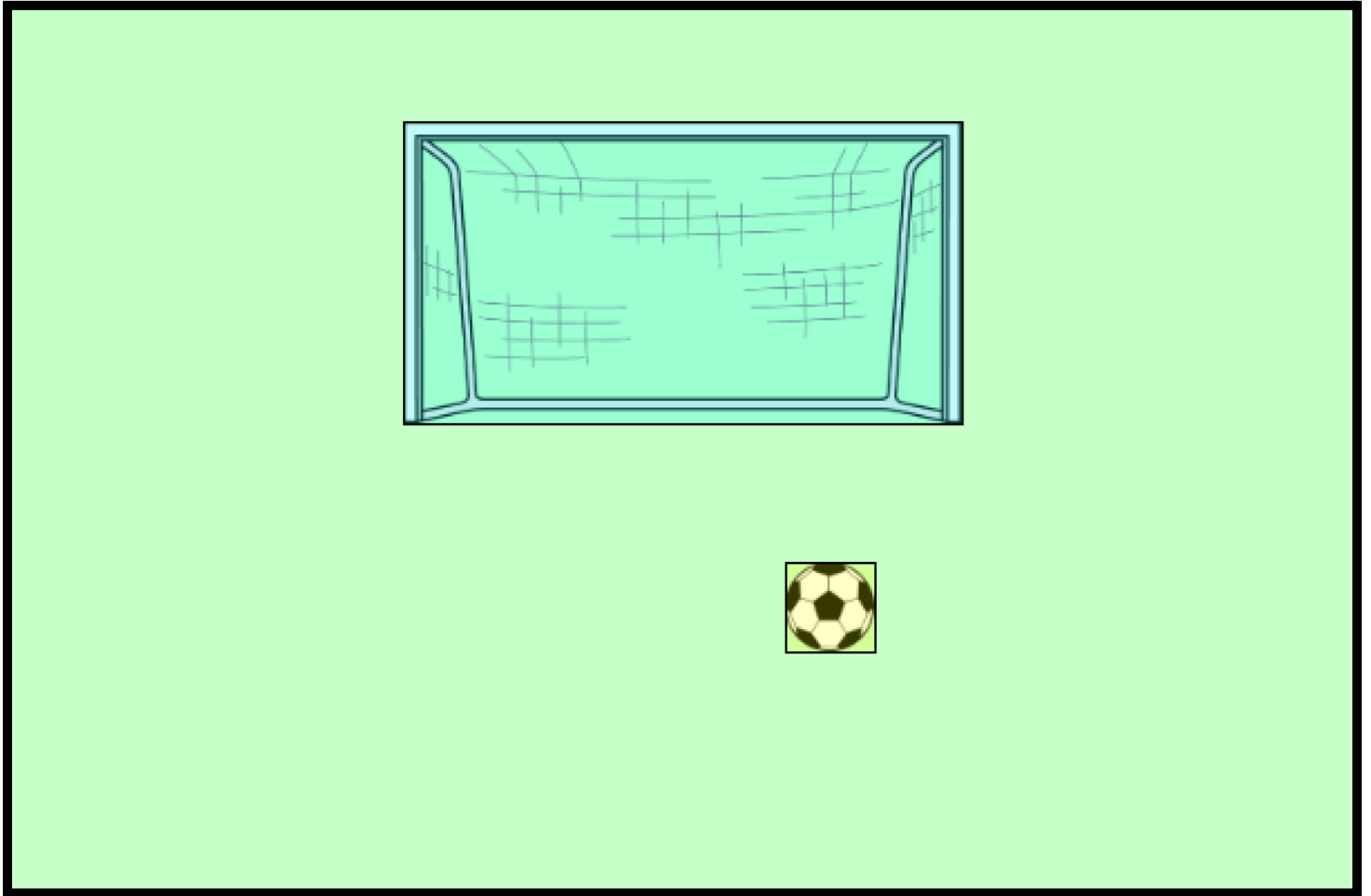


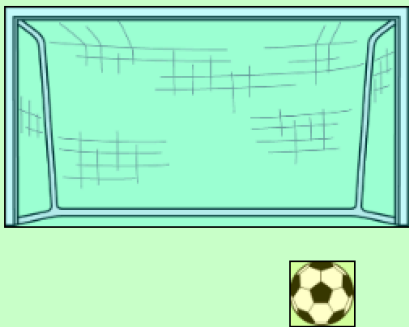
Introduction to Collision Detection

How can we detect a goal using Processing?

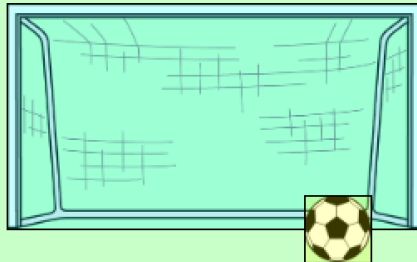


What do we consider a goal?

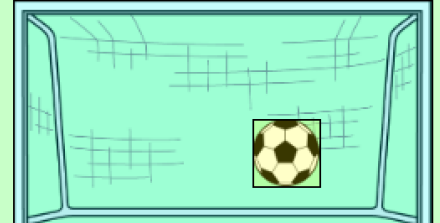
Not a goal ✗



Goal! ✓



More accurate goal! ✓



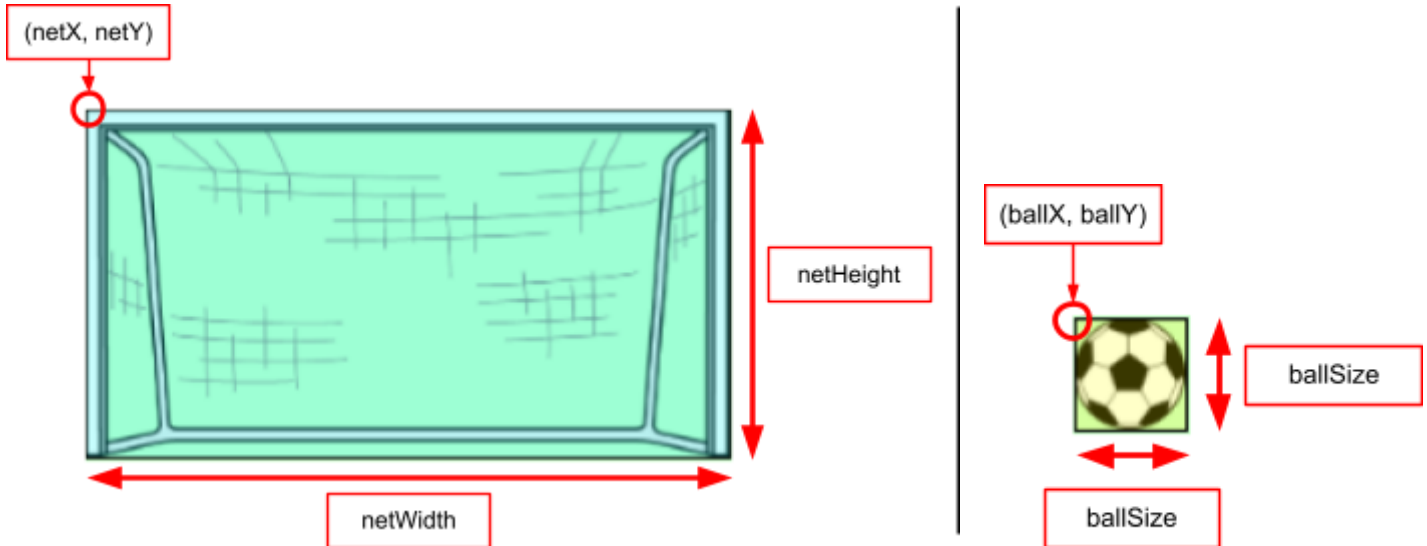
Ball is fully in the net!

How can we detect that the soccer ball is inside the soccer net?

We should first recognize what **variables** we're using for the **positions** and **dimensions** of these objects:

```
int netX = 175;  
int netY = 50;           // Position of net  
int netWidth = 250;      // Width of net  
int netHeight = 135;     // Height of net
```

```
int ballX;  
int ballY;               // Position of ball  
int ballSize = 40;       // Diameter of ball
```



Note: For simplicity, we're going to use squares and rectangles as our collision objects, rather than a circle for the ball.

In groups, brainstorm ideas and pseudocode for how you might check if the ball is in the net on your mini whiteboard.

Now, implement your solution in Processing!
Use the starter file posted in Google Classroom.

Once you're correctly detecting a goal, choose:

Give the player feedback for their action!

Choose to implement at least *one* or any of these to let the player know that they scored a goal:

- Randomly change the colour of the background.
- Play a sound.
- Have text appear on screen briefly ("Goal!").
- Play a confetti animation.
- Make up your own player feedback!

How can we make sure that the animation/sound is only played once *when the goal is initially scored*, and not while the ball is sitting in the net?



Implement a counter to display the number of goals!

- Declare an *integer variable* to store the number of goals/points.
- Each time a *new* goal is scored, the counter should be incremented by **1**.

How can we make sure that the counter is only incremented once *when the goal is initially scored*, and not while the ball is sitting in the net?



