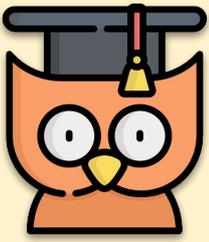


# Game Project - Obstacle Dash



Learning Outcomes:

- Write code to make a game where the player makes a rectangle jump in the air to avoid obstacles.

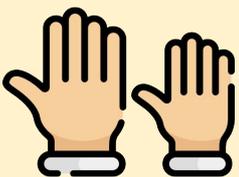
A large, stylized title graphic for the game 'Obstacle Dash'. The text is in a bold, red, bubbly font with a green outline, set against a background with a purple-to-teal gradient.

# OBSTACLE DASH



Revision topics before starting:

- ✓ Jumping
- ✓ Variables
- ✓ **If** Statements
- ✓ Booleans



**REMEMBER:** Put up your hand. We love to help!



## Making a Jumping Game

Now that we have the all important jumping game mechanic down we are going to put it into practice and design a game around the concept. The game will involve a player rectangle having to jump over incoming rectangle obstacles to score points. If the player touches off one of the obstacles then the game is over!



### Tasks

1

Draw the **ground** and the player rectangle and get it to jump when the player presses the “*SPACEBAR*” key. You may copy your code from the Jumping lesson if you wish.

2

Draw a rectangle to be the **obstacle**. The obstacle should start off the screen and travel from **right to left** across the screen.

3

If the player manages to jump over it then the obstacle should **disappear** off the left-hand side of the screen and **loop** back to the right-hand side of the screen.

4

Write the code to do **collision detection** between the player rectangle and obstacle rectangle. You can simplify this step and perform circular collision detection between the rectangles:

```
if (dist(playerX, playerY, obstacleX, obstacleY) < playerRadius + obstacleRadius){  
    // Collision Code  
}
```



### Expert Tip

Use `rectMode(CENTER)`; so the origin of the rectangles is in the centre and not the top-left corner!

5

Have a **score** that increases every time you successfully jump over an obstacle.

If the player collides with the obstacle then the score should get set back to **zero**.



Now would be a great time to save your sketch if you haven't already



### Challenge 1

Improve the graphics of your game by **adding in images** to use for the player, the obstacles and the background. Decide on a concept for your game and choose appropriate images.



## Challenge 2

When a collision between the player and the obstacle has been detected, set the game state to **game over**.

To do this you must declare a **Boolean** variable, initialize it to **false** and set it to **true** when a collision takes place. See the example below to help you:

```
boolean gameOver;
void setup() {
  size(800,800);
  gameOver = false;
}
void draw() {
  if(gameOver == false) {
    // All of the Game Code
    if (dist(playerX, playerY, obstacleX, obstacleY) < playerRadius + obstacleRadius) {
      gameOver = true; // Collision Detected
    }
  }
  else {
    // Draw a Game Over Screen
  }
}
```