# Project 97

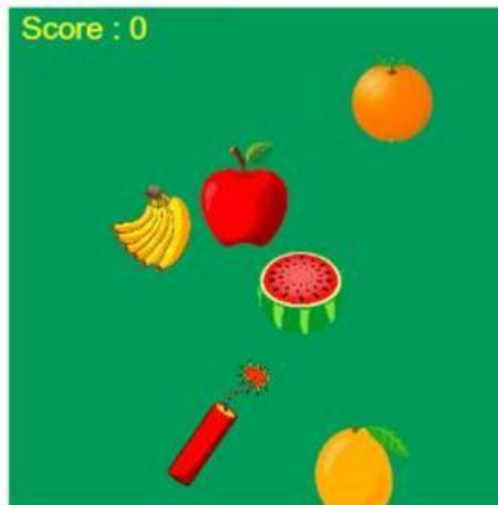**DP** EDUCATION

**Coding School**

Score : 0

How It Works — Start Here

Run — See the App

Built on Code Studio ▲

❖ Let's create the Fruit Crush Game.

❖ In this game, when you click on the fruit, points should be added, and if you click on the bomb, the game will be over.



❖ Let's first create the required sprites for this.

❖ Create the banana related sprite and set its animation and scale as below.

```
var banana = createSprite(200, 200);
banana.setAnimation("banana.png_1");
banana.scale = 0.3;
```

❖ Set the starting position of all sprites to be created as 200,200.

❖ All the images required for these sprites are included in the animation tab and are contained in the project given to you. Click on how it works and open it.

❖ In this way, create sprites for other fruits as follows.

❖ Create the sprite for Apple as follows.

```
var apple = createSprite(200, 200);
apple.setAnimation("Apple.png_1");
apple.scale = 0.3;
```

❖ Create the sprite like this for the watermelon.

```
var watermelon = createSprite(200, 200);
watermelon.setAnimation("watermelon.png_1");
watermelon.scale = 0.3;
```

❖ Create the sprite like this for the oranges.

```
var orange = createSprite(200, 200) →;
orange.setAnimation(▼ "Orange.png_1");
orange.scale = 0.4;
```

❖ Finally, let's create the sprite of the bomb moving here and there on the screen.

```
var dynamite = createSprite(200, 200) →;
dynamite.setAnimation(▼ "Dynamite.png_1");
dynamite.scale = 0.3;
```

❖ Create the sprite shown as game over as follows.

```
var gameOver = createSprite(200, 200) →;
gameOver.setAnimation(▼ "game_over.png_1");
gameOver.visible = false;
```

❖ Define variables as follows to calculate the time to change the motion of each fruit and bomb and to store the score.

```
var countForBanana = 0;
var countForApple = 0;
var countForWatermelon = 0;
var countForMango = 0;
var countForOrange = 0;
var countForDynamite = 0;
var score = 0;
```

❖ After creating in this way, let's first set a background color in the draw function.

```
function draw() {→
    background(rgb(57, 153, 86) →);
```

❖ Within it, every time the draw function is called, code in this way to increase the value of the above fruit and bomb variables by one.

```
function draw() {
  background( rgb( 57, 153, 86 ) );
  countForBanana    = countForBanana    + 1;
  countForApple     = countForApple     + 1;
  countForWatermelon = countForWatermelon + 1;
  countForMango     = countForMango     + 1;
  countForOrange    = countForOrange    + 1;
  countForDynamite  = countForDynamite  + 1;
```

❖ Now let's code using the variable created for the above banana to continuously change the position of the banana at a certain time.

```
if( countForBanana == 25 )
{
    banana.x = randomNumber(60, 380);
    banana.y = randomNumber(60, 380);
    countForBanana = 0;
}
```

❖ Whenever the value of countForBanana Variable is 25, the x and y position of the banana sprite will change randomly.
When it is 25, create the value of the countForBanana variable to be 0 again

❖ Just like bananas, code the same way as above to change the position of the apple in a certain time.

```
if( countForApple == 25 )
{
    apple.x = randomNumber(60, 380);
    apple.y = randomNumber(60, 380);
    countForApple = 0;
}
```

❖ Code to change the position like this for watermelons as well.

```
if( countForWatermelon == 35 )
{
    watermelon.x = randomNumber(60, 380);
    watermelon.y = randomNumber(60, 380);
    countForWatermelon = 0;
}
```

❖ Code to change the position like this for the mango.

```
if( countForMango == 25 )
{
    mango.x = randomNumber(60, 380);
    mango.y = randomNumber(60, 380);
    countForMango = 0;
}
```

❖ Code to change position like this for oranges as well.

```
if( countForOrange == 25 )
{
    orange.x = randomNumber(60, 380);
    orange.y = randomNumber(60, 380);
    countForOrange = 0;
}
```

❖ In the same way, create the change of position of the bomb.

```
if( countForDynamite == 10 )
{
    dynamite.x = randomNumber(60, 380);
    dynamite.y = randomNumber(60, 380);
    countForDynamite = 0;
}
```

❖ To change the time of each item's position change, it is possible to increase or decrease the corresponding value of the corresponding variable.

❖ When you click on the fruit, the sprites will disappear and the score will be added and create a sound to play.

```
if ( mousePressedOver(banana) )
{
    playSound( ▼ "sound://category_achievements/lighthearted_bonus_objective_2.mp3", ▼ false); ─
    banana.destroy();
    score = score + 10;
}
```

❖ Above is the code related to what happens when you click on the banana. Create code for other fruits in the same way.

❖ The code related to what happens when you click on the apple is as follows.

```
if ( mousePressedOver(apple) )
{
    playSound(▼ "sound://category_achievements/lighthearted_bonus_objective_2.mp3", ▼ false); ⊣
    apple.destroy();
    score = score + 10;
}
```

❖ Code what happens when you click on the watermelon as follows.

```
if ( mousePressedOver(watermelon) )
{
    playSound(▼ "sound://category_achievements/lighthearted_bonus_objective_2.mp3", ▼ false); ⊣
if ( mousePressedOver(mango) )
{
    playSound(▼ "sound://category_achievements/lighthearted_bonus_objective_2.mp3", ▼ false); ⊣
    mango.destroy();
    score = score + 10;
}
```

❖ Code as follows to create what happens when you click on the mango.
❖ Code as below to create what happens when you click on oranges.

```
if ( mousePressedOver(orange) )
{
    playSound(▼ "sound://category_achievements/lighthearted_bonus_objective_2.mp3", ▼ false); ⊣
    orange.destroy();
    score = score + 10;
}
```

❖ When you click on the bomb, all the sprites disappear and the game over sprite should be visible.

```
if ( mousePressedOver(dynamite) )
{
    playSound(▼ "sound://category_explosion/8bit_explosion.mp3", ▼ false); ⊣
    banana.destroy();
    apple.destroy();
    watermelon.destroy();
    mango.destroy();
    orange.destroy();
    dynamite.destroy();
    gameOver.visible = true;
}
```

❖ Then let's create the related codes to show the scores as follows.

```
fill( ▼"yellow");
textSize(25);
text("Score : " + score, 10, 25); →
drawSprites();
```

❖ The scores stored in the score variable are shown as above. finally drawSprite(); Call the function.