



Mash Club - Coding and Blogging - Session 1

2simple

1 Session 1

Coding skills to master and concepts to understand:

- The screen layout and language used.
- Adding Backgrounds and Objects in design view.
- Creating code by dragging blocks.
- Sequencing code
- Collision detection
- Responding to key presses or swiping
- Keeping score
- Using a timer
- Conditionals ('If' statements)
- Testing and improving code

Game to Develop: Escape the Dragon

Session 1

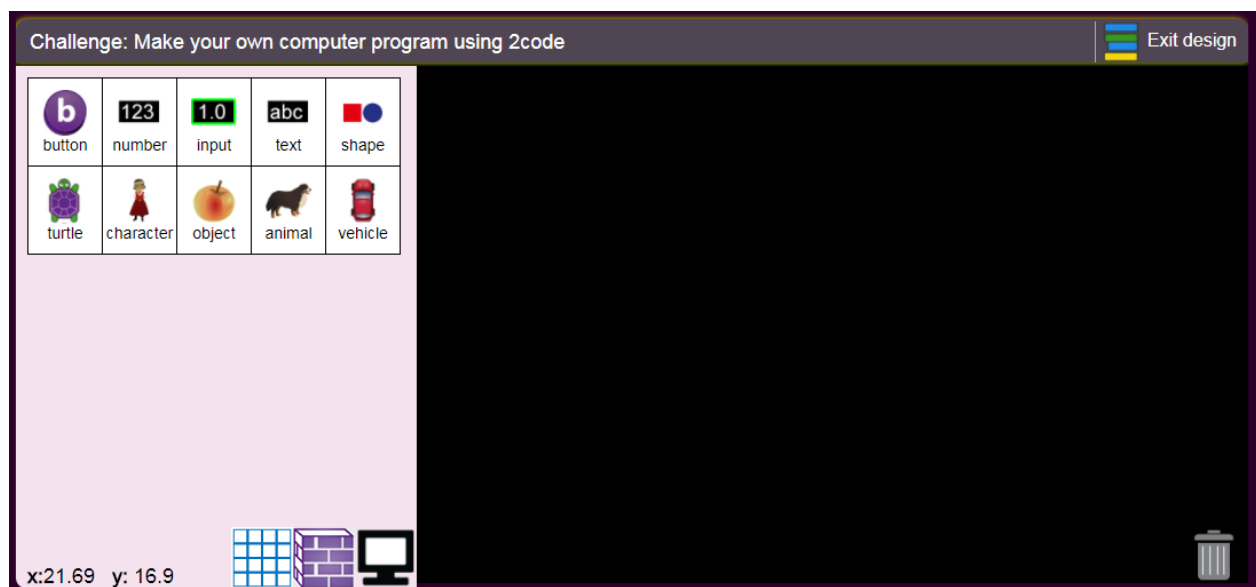


In this game, the dragon flies around shooting fire, the player (the wizard) needs to avoid being set alight!

1. The first thing to do is to login to Purple Mash and Open 2Code free code Gorilla level. Your teacher should show you how to do this and help you to get to the right place.
2. The screen that opens initially is the **coding screen** which we will be looking at in more detail in a








minute. For now, click on the **Design** button to switch to **design view**. This is where the look of the program that you are coding is designed.



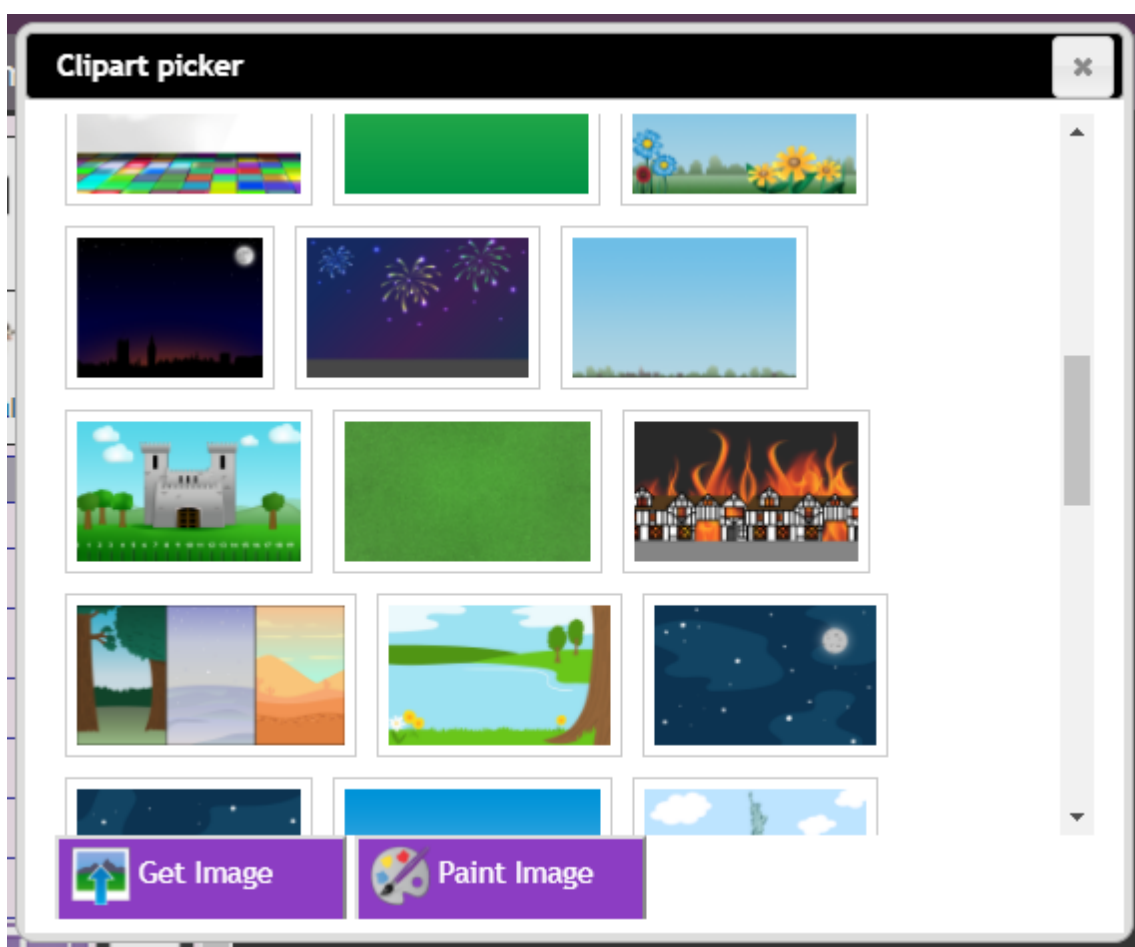
3. The left-hand side of the screen contains the available **objects**, these are like the characters in the program. The right-hand side is the **stage** where you design the look of your program.



4. Click on the , this will open a table which shows the **properties** of the background.

Property	Value
type	background
name	background
 colour	
 image	
Grid size	4










5. Click on the image which is currently a question mark symbol. This opens the Clipart Picker with some ready-made backgrounds.



6. Scroll down until you find the castle image and click on this, now you have a background for your

program!

7. Now to add your dragon: drag an animal **object** onto the **stage**. Notice that at the left-hand side of the screen under the types of objects is a property table which is all about the animal that you have just added. You can drag the animal anywhere you want on the screen, this will be its starting position.

Property	Value
type	animal
name	myAnimal1
 x	4.552
 y	4.953
 allow off screen	No
 rotation style	Side to side
 angle	0
 speed	0
 scale	70
 image	
 friction	0
 show/hide	show
Copy	







Notice that when you drag it around, the

 x	4.552
 y	4.953

number

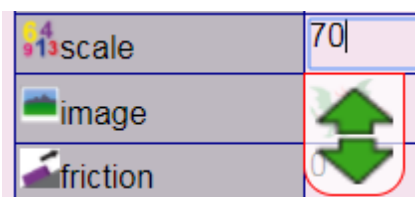
change. You can click on the x or y properties and change them. Do this and see what happens?

 x	9
 y	4.953
 allow off screen	

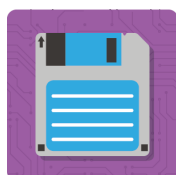
8. At the moment it is a dog, so let's change this. Click on the image, it should open the Clipart picker.



9. The dragon can be found by clicking where it says 'animals' at the top, then clicking on 'fairy tales'. Choose a dragon image that you like.
10. You might end up with a dragon that is far too big. To change the size, click on the scale **property** and change this. The dragon in the example has scale= 30.

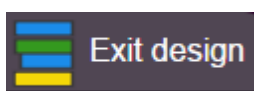


11. Click on the name **property** 'myAnimal1' and change it to 'dragon'.
12. Now we need a wizard to fight the dragon. Drag in a character **object**. See if you can change the image to a wizard and change the name to 'wizard'.



Save: Now that you've designed your screen, save your work so that you don't lose it. Click on the save button at the top and give your game a name, maybe 'Escape the Dragon'? This will save it into your work folder.

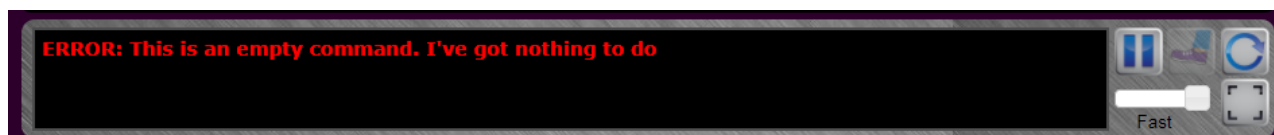
13. Now to begin coding. Exit the design view and return to the code view by clicking



14. Click on the Play button at the top of the screen . What happens?



Nothing happens, why? – look at debug window on the bottom right for a clue.

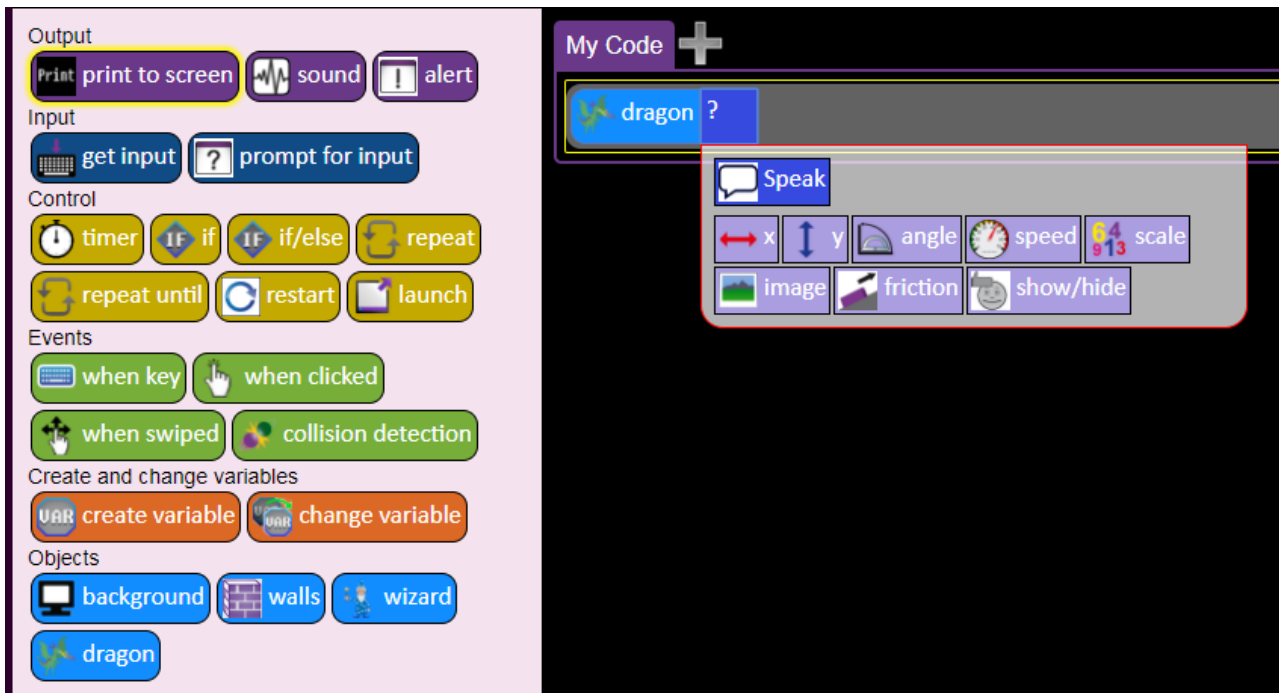


When you are coding, the debug tools will help you to get a deeper understanding of what your computer is doing and also work out why things aren't working as expected. In this case, nothing happens because you haven't written any code yet so the program has nothing to do!

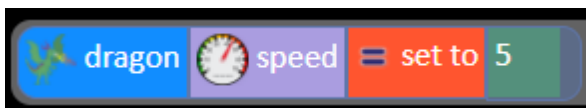


15. Press the stop button

16. Look at the coding view. On the left-hand side are all of the available **code blocks**. At the bottom are all the **objects** including the dragon and the wizard. Find the little image of the dragon and drag it into the coding area.



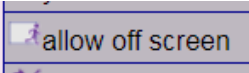
17. Click on the box that says  and set the speed to 5.



18. Press the Play button and see what this does. Press stop.



Go to the design view again and click on the dragon. Investigate what difference changing

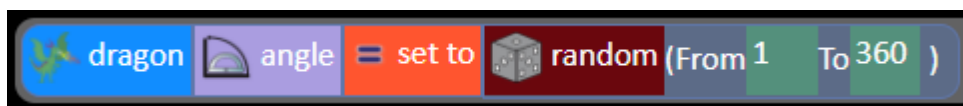
the property  makes by changing it and pressing Play. **Make sure that you set it back to 'No' once you have investigated.**

19. Your program is now doing something but it's not very interesting, let's start making it better. Exit the design view and have a look at the coding view. Find the little image of the dragon again and drag it into the coding area below the speed **block**.

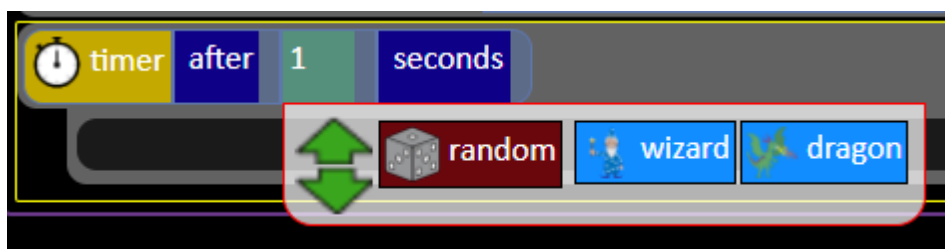
20. Click on the box that says . Set the dragon's angle to 5.



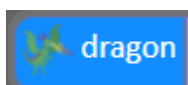
21. Click Play to see what this does.
22. Still a bit boring? Now set the angle to random between 1 and 360. Do this by clicking where you entered the number 5 and choosing random, then enter the numbers as shown below.



23. Test this out a few times. Each time the dragon should fly at a different angle; you will probably think it's still not really exciting so let's do a bit of clever coding to make the dragon fly around in a more interesting way.
24. Ideally we want the dragon to keep changing direction. One way to do this is to use a timer.
25. Drag a timer control into the coding window.



26. Click on the word **after** and change it to **every**. Then change the number of seconds to 2.
27. Drag the line of code about the dragon moving inside the timer by holding your mouse down on



the **dragon**, keeping it pressed down, move the mouse to just below the word



and an orange outline should appear, then release the mouse button (or your finger if using a touchscreen). It should look like this:



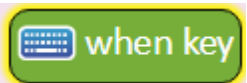
28. Now press play and your dragon should move in a more interesting way.

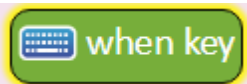


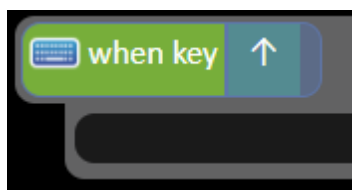
Try changing how often the timer runs by altering the number of seconds. What effect does this have? Make sure you change it back to something sensible at the end of your investigating.

29. Now we have the dragon moving, he keeps catching the poor wizard! Let's get the wizard to move when you use the arrow keys on your computer or when you swipe on a tablet. The code is different depending upon what you are using so choose the right instructions below. **If you want your code to work on both types of devices then you will need to include all of the code for both:**

If you are using a computer without a touchscreen

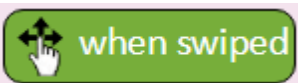
Find the  command on the left and drag it under the timer (not inside the timer).

Click in the darker green box next to the words  and then press the ↑ key on your keyboard so your code looks like this:

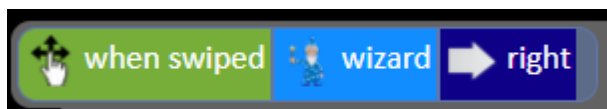


Create the following code inside the **when key** block:

If you are using a touchscreen device

Find the  command on the left and drag it under the timer (not inside the timer).

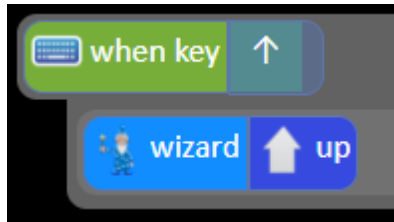
Select the following options for your **when swiped** block:



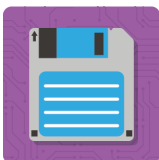
Create the following code inside the **when swiped** block:



Next drag in the same commands again but use the ←, → and ↓ choices. Your code should look like this:



Next drag in the same commands again but use the ←, → and ↓ keys. Your code should look like this:



Save and

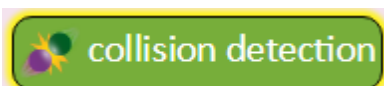


test your program.

Can you make the wizard move when you press keys or swipe?

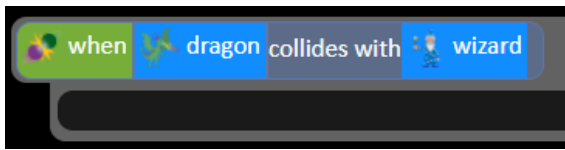
30. Now we need something to happen if the dragon catches the wizard. For this, you are going


to use **collision detection**. Drag a

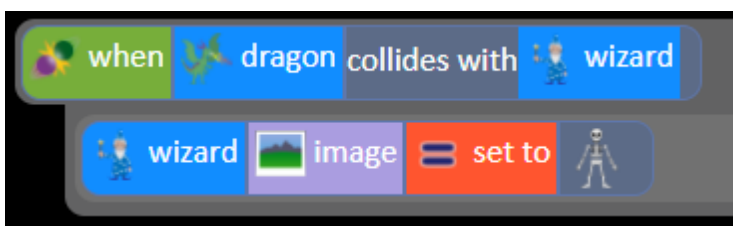


block into your code and

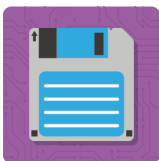
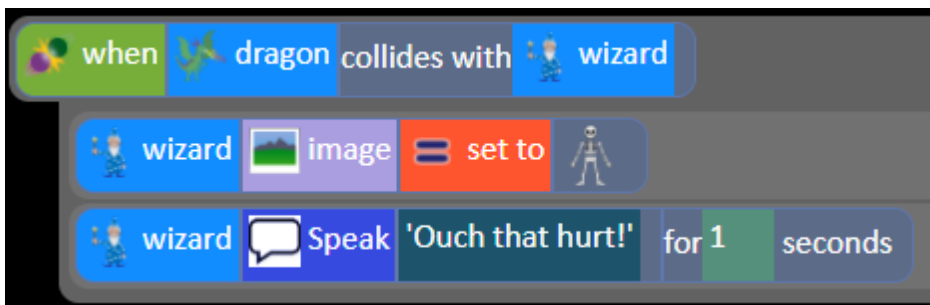
select the following choices:



31. The wizard is going to change into a skeleton. Add this code inside the **collision detection block** by dragging the **wizard object** from the left-hand side inside the collision block. From the options that pop-up, select image. Click on the  to open the clipart picker. You can find the skeleton in the haunted house category.



32. Let's make him say 'Ouch that hurt!' by adding the following by dragging the wizard object in again and selecting different options:



Save and



test your program.



Does the wizard change into a skeleton when he is caught? What do you think we need to do next?

33. The main problem now is that the wizard changes to a skeleton but then stays like this. Let's make him turn back to himself after a couple of seconds using a timer. Change your code to look like this:



Save and











test your program.

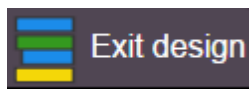


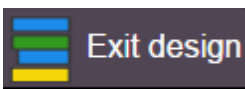
What do you think we need to do next to make a better game?

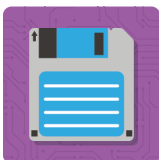
34. Let's add a score for the game so that the longer the wizard keeps away from the dragon, the more

points he gets. Go to the  view and drag in a number object . Put it wherever you'd like the score displayed on the screen. This will be the score so rename it 'score' in the property table:

Property	Value
type	number
name	score
 value	0
 text size	18
 text colour	
 border	No
 x	10
 y	0



35. Exit the design view . Add the following code at the beginning of your program. It will add 1 point every second:





Save and

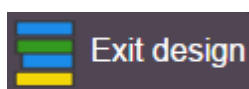


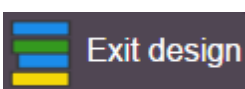
test your program.

36. Now this game could just go on forever and nothing happens to the score when the wizard gets caught. Lets give the wizard some lives so that the game can end when he loses all his lives. First



go into the  view and drag in another number object . Put it wherever you'd like the number of lives displayed. You could change the text colour property so the text is red to stand out if you wish. Rename it 'lives' in the property table.



37. Exit the design view . Add the following code at the beginning of your program to set the number of lives to 3 at the start of the game.



38. Change the collision detection code so it looks like this:



Which line in the code above makes the wizard lose a life?



Save and

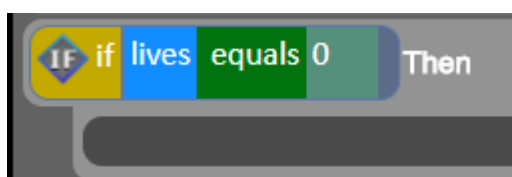


test your program.

39. Now the wizard is losing lives but the game still doesn't end. You need to write some code so that when the number of lives is zero, the game ends. To do this you need to add an **'if' statement**.

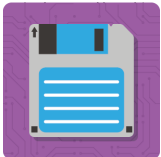
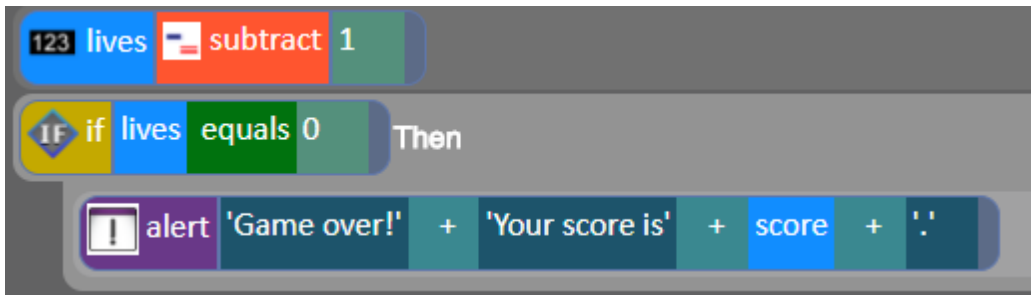


Drag an **if** block onto the line after the wizard loses a life. Use the options that you are shown to make your code look like this:



40. One way of sending a message to the player is to use an **alert**. This is in the **Output** section of the

coding blocks. Add the following alert (the dot at the end is to make a full stop).



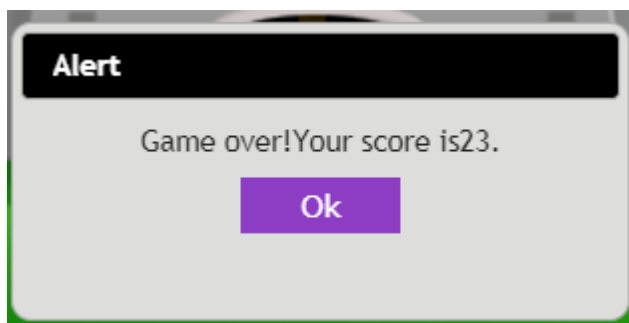
Save and



test your program.

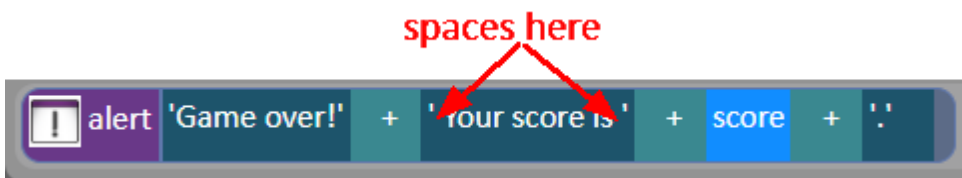
41. When you test, you should notice the following '**bugs**' which we will solve in the next few steps:

- The alert box doesn't have space in the right places:



- Sometimes the wizard loses more than 1 life at a time because the dragon is still colliding with him after he's just lost a life.

42. To solve the alert box issue, you need to add some spaces into the text in your code:



43. To solve the loss of too many lives problem, we can make the dragon hide for a bit after the wizard loses a life. Add the following code into the collision detection block:



Save and



test your program.

You now have your first finished game!

For the time being, don't worry about the fact that the game goes on once you click OK, this is your first session after all!



If you've got this far, try some of the following Challenges to stretch you coding ability. They are not in any order, just pick one that you fancy?

Challenge 1

Add a title by adding a Text object in design view. You will need to write some code to make it move out of the way once the game starts or to make it hide. Explore the different actions that the text object has in code view and decide what is best.

Challenge 2

Change the game so when the wizard and dragon collide, it's the dragon who loses; what could he turn into?

Challenge 3

Change the scoring, maybe the wizard loses points when he gets caught? Maybe if he survives for 30 seconds, he gets bonus points? Maybe if he survives for long enough, the dragon gets tired and turns

into a jelly! Making the wizard the winner. This is your game now so decide how you would like to adapt the scoring and see if you can code it.

Challenge 4

Spend some time exploring other options in 2Code; can you add something to personalise your game