

---

---

# PROGRAMMING IN ALICE

---

## WORKSHOP: CONDITIONAL STATEMENTS

---

### LEARNING OBJECTIVES

---

The learning objectives for this workshop are:

- Students can design and implement programs using conditional statements
- Students can use interactive commands to get input from the user

---

### EXERCISE 1

---

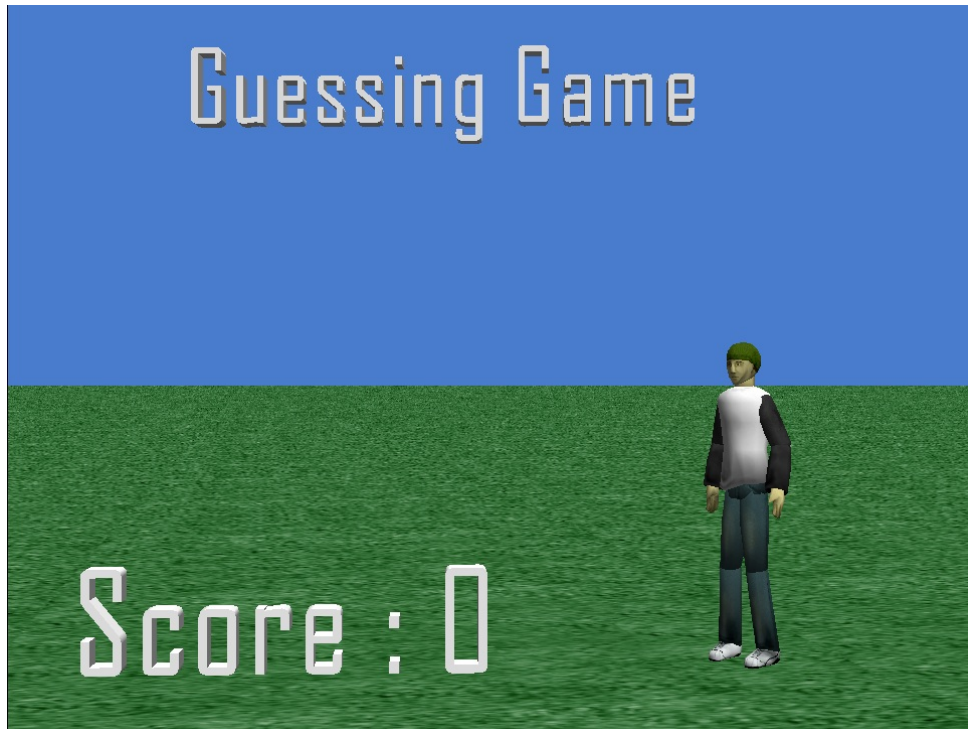
In this activity you will design and develop a quiz game in which the user is asked a series of questions. Answers will be stored in variables, and conditional statements will be used to check whether or not each answer was correct. The program will keep track of how many questions the user answered correctly, updating the score after each question. To assist in the development of the program we provide a number of suggested development steps.

---

#### STEP 1: CREATING THE WORLD

---

Create a new world that includes a Person object (your choice) who will be the quiz master. The world will also require a 3D Text object to be used as a title for the game, and a second 3D Text object to be used to display the current score. Your initial world should look something like the following.



To include the 3D Text, click on **File>Add 3D Text**.

---

## STEP 2: CREATING A SCORE VARIABLE

---

Your program will need to keep track of how many questions the user has answered correctly. To do this, you will need to create a variable call *score* (with type Number) in **my first method**. The score must be initialized to 0.

---

## STEP 3: THE FIRST QUESTION

---

In this step you will ask your first quiz question. This quiz question should have a number as an answer (e.g. you may ask how many states are in Australia). You will need a variable to store the answer – this variable will have type Number. To get the answer from the user, you must use the **ask user for number** function. After the answer is received your program must check whether the answer was correct. You should use an if-then-else statement to do this – use the **a == b** math operator to check that the answer from the user is equal to the expected answer. If the answer is correct, then you must increase the score by 1 and your quiz master should say “Correct”. Otherwise the quiz master should say “Wrong”.

---

## STEP 4: ADDITIONAL QUESTIONS

---

Repeat the above process until you have at least 3 quiz questions. For each quiz question you will need an additional variable to store the answer. You must include at least one yes/no question using the **ask user for yes or no** function. For these kind of questions, the answer variable must be of type Boolean (i.e. it has values of either true or false).

---

You must also include at least one question that accepts a string as an answer, using the **ask user for a string** function. The answer variable for these questions must have type String. Setting up the check in the if statement is a bit tricky in this case – the easiest way to do this is to create the if/else statement, then drag the variable representing the answer into the if condition.

---

### STEP 5: UPDATING THE SCORE

---

Create a new World method (i.e. it should be created within the World object) called **refreshscore**. This method will update the score shown in the 3D text in the bottom left hand corner. This method should take a single parameter, called **score**, which must be a number. The method will update the *text* property of the 3D Text. To do this click on the properties for the 3D Text object and drag the *text* property over to your method code. You will then need to use a combination of the two string functions from the World object to update the string.

You should call the **refreshscore** function after each quiz is answered.

---

### EXTENSION 1: ANIMATING THE QUIZ MASTER

---

Create a new method called **reactToCorrect** for the Quiz Master object (corresponding to whatever person object you created in step 1). The method should perform an animation for the quiz master when the user answers a question correctly. For example it may be as simple as nodding the head, or a more complicated animation, such as jumping up and down in celebration. This method can then be called together with the say “Correct” method whenever a question is answered correctly.

Create a second new method called **reactToWrong** for the Quiz Master object. This is similar to the **reactToCorrect** method, except the quiz master is in this case is disappointed, so may for example shake his/her head.

---

### EXTENSION 2: A PERFECT SCORE

---

At the end of the quiz, if the user has achieved a perfect score the quiz master will be extremely happy. Create a new animation called **celebrate** that will perform some over-the-top animation of the quiz master (you might try get him/her to do the moon walk, or backflips or a star jump). You should call this method in **my first method** if the user has achieved a perfect score at the end of the quiz.