Programming

in

Alice

PRACTICALS

1 Practical 1:

a) Create a new world using the Room blank slate. Set a camera marker and save the project as ProgrammingPrac1.

b) Add a chair from the Household group. For now, place it somewhere in the middle of the room.

c) Create a person – you choose the age and other details

d) Arrange the person so he or she is sitting in a chair

e) Set the person's vehicle to one of the chairs.

f) In the code editor, write instructions to rotate the chair slowly through 360 degrees, both horizontally and vertically

g) Set the room to be see-through, so you are able to watch the rotation even when the person goes underground, then run the program.

h) Make any adjustments needed to the person's positions, and also his body part positions.

i) Once you're happy with it, set the ground back to being opaque, place the chair where you want it, and add a few more items from the Household group to make a room scene.

j) Check your scene in scene layout view to make sure everything is in a good position in 3D. You can resize the objects if needed.

2 Practical 2:

a) Create a new world using a blank slate

b) Create a person – you choose their looks and outfits

c) Think of a short dance sequence (about 3 moves, eg step, turn, kick). Make use of do-together to make arms and legs move at the same time.

d) Make your person carry out the dance, paying attention to what should move together, and what in sequence.

e) Run it to test it

f) Make them carry out the dance sequence 5 times Test the program

3 Practical 3

a) Create a new project using blank slate Grass, and save it as Prac5

b) Create a biplane. Set its width to 2.5, and position it to the right of the screen, facing left.

c) Create a bunny. Position him so he is sitting in the biplane. Set his vehicle to the biplane.

d) Create a scene property named **moreLoops** of data type Text String. Set its initial value to 'Yes'

e) Create a procedure called **bunnyLoop**, which:

- Moves the biplane forward by 2
- Turns it forward in a full loop for a duration of 1
- Moves it forward by 2
- Turns left by 0.5
- Make the bunny say 'Hey, that was cool! Shall I go again?"
- Get a reply from the user into the variable **moreLoops**

© G. Thornhill 2022

f) Edit **myFirstMethod**

g) For a count of 3, do together:

- Insert a comment 'Take off'
- Move the biplane up by 1, with a duration of 0.15
- Move the biplane forward by 1 for a duration of 0.15

h) While **moreLoops** contains the text 'Yes', call the procedure **bunnyLoops**

i) Insert a comment "Landing"

j) For a count of 6, in a do together:

- Move the biplane down by 0.5 for a duration of 0.2
- Move the biplane forward by 0.5 for a duration of 2

k) Move the biplane forward by 1.

l) Test the program and fix any problems. The bunny should keep doing loops until the user no longer answers 'Yes.'

4 Practical 4:

a) Open the program you saved in Practical 2

b) Import the biped class that contains the 'walk' methods. It will be provided by the instructor

c) Using the clipboard, copy the instructions needed for the dance you programmed in Practical 2 to a new procedure named coolDance in the Biped class. Now all bipeds should be able to dance.

d) Export the biped class so you will be able to use it in other projects

e) Open a new project using the blank slate of your choice.

f) Import the biped class that you saved.

g) Create three bipeds standing in a row facing forwards. Line them up so they are all at Z location 0.

h) Write a program that makes them walk forward together, then do the dance together.

5 Practical 5

a) Open the project you saved in Pratical 3

b) Change the coolDance procedure to take a parameter to control the number of times the dance will be repeated.

c) Put the existing code in coolDance into a 'count' block, using the procedure parameter as the counter. Change MyFirstMethod to include this parameter where **coolDance** is called. Test the program to make sure it works.

d) Create a couple more bipeds

e) Choose which ones you would like to do the dance. The others will just stand and watch.

f) Put these bipeds into an array called 'dancers' in MyFirstMethod.

g) Write a new procedure under Scene:

   a. Name it **groupDance**
   b. It should take two parameters: an array of bipeds, and a number of repeats.
   c. Use an each in … together construct to call the **coolDance** procedure of the current item in the array.

h) Change MyFirstMethod so that instead of calling **coolDance** on each biped, it calls the **groupDance** procedure belonging to the scene (this).

i) Test the program. At the same time, get an estimate of how long the program takes to run, either by using a timer/stopwatch, or by counting the seconds.
j) Add a doTogether block into MyFirstMethod, and move the call to **groupDance** inside it.
k) Also in the doTogether, add a playAudio method. Choose one of the audios under the 'soundtrack' folder. Change it to a Custom Audio and change the start and/or stop times so that it plays for the same number of seconds as your program takes to run.
l) Run the program. If the soundtrack plays for too long or too short a time, adjust the start/stop times.

6 Practical 6

a) Open the program you saved in Practical 3
b) Create a very short audio that will play as the bunny makes the loop. You can choose what he says, but make him sound like he's having fun. Experiment with changing his voice
c) Add the audio so it plays as he's doing the loop.
d) Save this program, and open the one you saved in Practical 5
e) Think of something suitable that one of the dancers could shout at one point in the dance – maybe 'Yippee!' or 'Yee-ha'.
f) Record it in Audacity, and play it at the right point in the dance.
g) Record something that one of the characters could say before the dance begins. Maybe 'We are the Cool Dance gang!' Experiment with the Audacity effects to make the voice suitable for the character who will say it.
h) Play it just before the dance begins.

7 Practical 7

a) Create a new project using the Desert blank slate
b) Set a camera marker
c) Add a few props – maybe some rocks and a cactus
d) Move the camera left by 20. Set a camera marker.
e) From the Vehicle > Automobile class, add a Humvee. Set its width to 1.5. Position it so it's facing towards the right of the screen.
f) Select the Humvee and create an object marker.
g) Add a procedure called MoveVehicle that:
    a. Moves the Humvee to the object marker
    b. Move it forward by 30 for a duration of 5
h) Add a time event listener that triggers every 6 seconds. It should call the MoveVehicle procedure.
i) Also add this procedure to the SceneActivated event listener.

8 Practical 8

a) Create a new project using the Snow blank slate
b) Add a few props
c) Add a dragon and a baby dragon. Make them small enough to fit into the scene nicely.
d) Add default model manipulation for the dragon, and an object mover for the baby dragon.
e) Run the program. You should be able to move the dragon with the mouse, and the baby dragon with the keyboard.

© G. Thornhill 2022

9 Practical 9

a) Open the program you saved in Practical 8

b) Search the object gallery for flames. Add two flame objects, and position and size them so they look as if they are coming out of each dragon’s mouth. Set their vehicles to the dragons. Set their opacity to zero.

c) Add a collision listener for collisions between the dragon and the baby dragon. This should call a procedure that:

    a. Sets the opacity of the flames to 1 for a duration of 0

    b. Sets the opacity of the flames back to 0 for a duration of 4

d) Run the program, and make the dragons collide. They should look as if they are spouting flames every time they collide.

© G. Thornhill 2022