# GAME PROGRAMMING & DESIGN – LAB 3

## "Tag" - a game with multiple levels (states)

# I. BACKGROUND

### 1. Introduction:

In a previous lecture we talked about "game state". In particular we suggested that:

1. All games consist of a sequence of states.

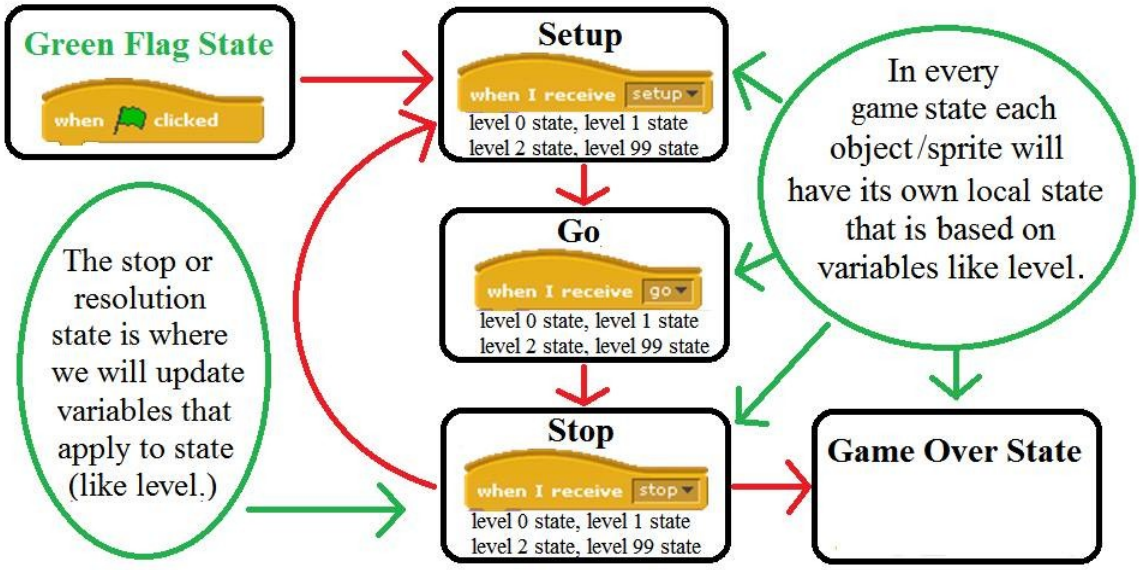
2. These states can be characterized by a combination of visual, audio and/or animation effects, as well as a set of rules that are being applied.

In this lab, you will create a simple game in Scratch using the concept of "game state".

### 2. Game State vs. Object State:

Objects in the game proceed through their own states as well. These states are defined by the behavior and functionality applied at that time.

In this lab you will create several objects that use the game state transition diagram below:




**NOTE:** The behavior of each object during the run, setup and go states might be different based on the level.

# II. PLANNING THE GAME

## 1. Description of Game:

*(Never start a game by sitting down and coding. Every large programming project needs to be planned out, in as much detail as possible BEFORE you start coding. The more time, effort and detail you put into your planning stage, the faster the coding of the game will actually go.)*

The game will be called **Tag.** In the game a cat (the player) will alternately attempt to catch a dog, and escape from a dog, depending on whether or not the cat is "it". The player will control the cat using the arrow keys.

In the first iteration of the game (what this tutorial covers), the player will click the screen in order to advance through the various parts of the game. There is no "score" variable in this iteration of the game.

## 2. Programming Outline

Plan, plan, plan, before you write a single line of code! A programming outline will help you see problems, and help prevent you from making mistakes

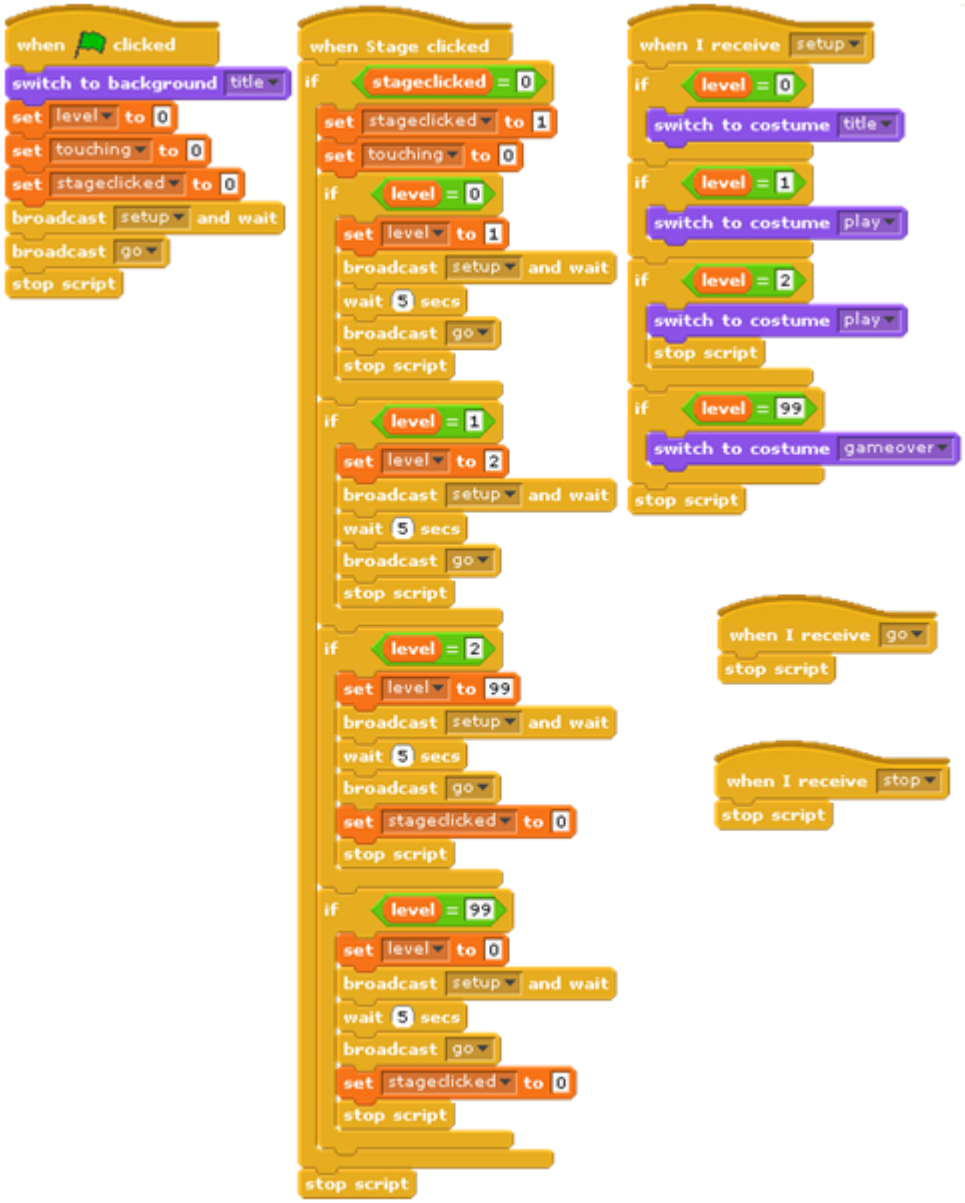
| Object<br>(Name, Description) | Properties<br>(What are the facts about this object? What does the object look like? How many images will you need for it? Where does it start? What are its states (alive, dead, etc.) | Functions<br>(What does this object do? Can it move? Can it change costumes? Can it interact with other objects? Can it interact with the player?) |
|---|---|---|
| **Stage** (the stage itself, which will have a title and a gameplay screen). | Stage will have three backgrounds (Title, Play, and Gameover)<br>Stage will have three variables:<br>level (initially set to 0 and used to manage object state)<br>touching (initially set to '0' for no, and is used to track if dog and cat are touching)<br>stageclicked (initially set to 0) | Stage will show the "title" and "gameover" messages by switching backgrounds. The stage will be in charge of changing the game level, and will broadcast a series of messages including:<br>setup<br>go<br>The stage will also respond to being clicked on. |
| **Cat** (the player) | Will have two costumes so that it can appear to run. | The cat will be controlled by the player, using the arrow keys. If the cat, touches the dog, the cat will broadcast a "stop" message and set the "touching" variable to 0. |
| **Dog** (the opponent AI) | Will have two costumes so that it can appear to run | When the player is "it", the dog will run away from the player. When the player is "not it" the dog will pursue the player. |
| **Text** (an object that has a series of text messages as costumes) | Has a series of text messages as costumes. Messages are:<br>"Click anywhere in screen to start."<br>"You are it. Use arrow keys. Catch that dog."<br>"You are NOT it. Use arrow keys. Run!"<br>"Tag! Click anywhere to continue."<br>"Game Over. Click anywhere to play again." | Each of the messages appears at appropriate times during the games. |

# III. CREATING THE GAME

## 1. Creating the Objects:

By looking at the programming outline, you should be able to see that we need four objects. In SCRATCH these objects are called Sprites.

1. To save time, this game is available as a starting template. You will not get all of the code, but you will get much of the starting requirements. This tutorial assumes that you have already downloaded the template, which is available here:
   a. ..\OtherMaterials\ExampleGames\Ex_Tag_Template.sb
2. Select the stage, by clicking on the **Stage** icon in the Sprite window (lower right of screen).
   a. You should see the following scripts:




   b.
   c. Examine the first four scripts in the scripts window above.
      1. What does each of the following scripts do?
         1. "when green flag clicked"
         2. "when Stage clicked"

3. "when I receive setup"
4. "when I receive go"
5. "when I receive stop"

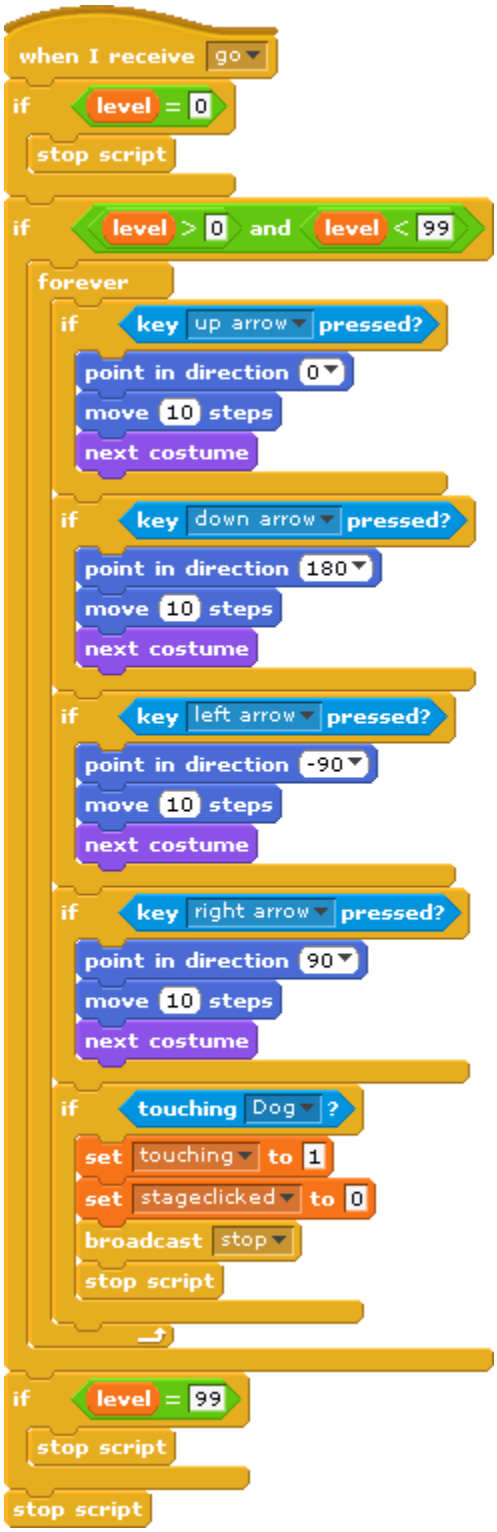
d. **IMPORTANT**: <u>The most important script is the "when Stage clicked" script, because it controls the how the game changes from state to state.</u>

1. Notice that there is a flag variable called "stageclicked". This variable keeps the user from accidently running the script twice (or more) in a row (double-clicking). How does it do this? When does it get reset?
2. The "touching" variable is used by the cat and dog to see if they are overlapping.
3. There is a 5 second wait between the "setup" and "go" broadcasts. This is so that the text messages, telling the player what to do, have a chance to be read. Thus the game is held in the "setup" state for 5 seconds.
4. When the game level is changed to "1`" and "2" the variable "stageclicked" does not get reset to 0. Why do you think that is?

3. Select the cat, by clicking on the cat icon in the Sprite window.
   a. When the green flag is clicked the cat should be hidden. Create that script for the cat.
   b. The cat will always start in the same place, but based on the current level, the cat may be hidden or shown. Create the following script for the cat:


   1.
   2. What does the script above do?

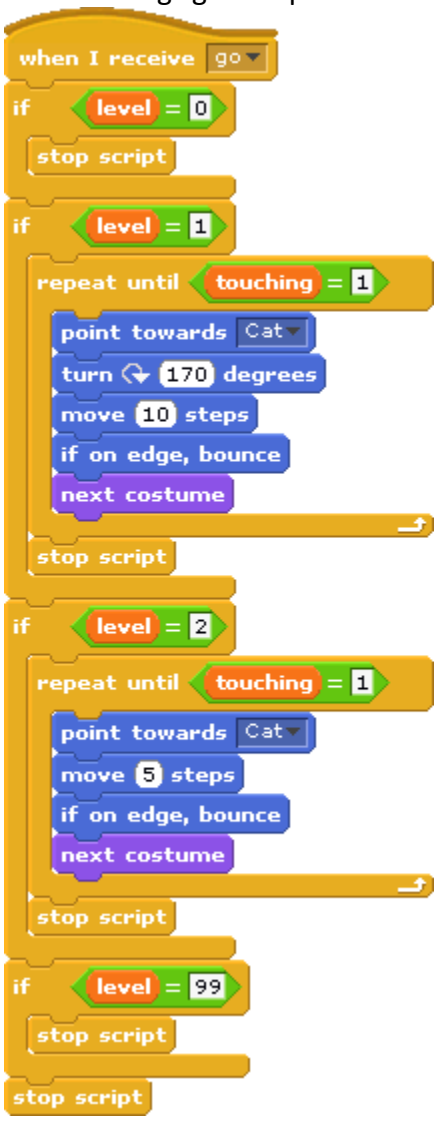
   c. When the cat receives the stop broadcast, he should become hidden. Create that script for the cat.
   d. The only script that still needs to be created is the "when I receive go" script. The cats behavior when it receives the go script will be different based on the level of the game. Create the following script for the cat.




1. 
2. What does this script do?

4. Select the dog, by clicking on the dog icon in the Sprite window.
    a. Like the cat, the dog will have 4 scripts. 3 of these scripts will be almost exactly the same as the cats ("green flag", "setup", "stop").
    b. When the green flag is clicked the dog should be hidden. Create that script for the dog.
    c. When the dog receives the stop broadcast, he should become hidden. Create that script for the dog.

d. When the dog receives the setup broadcast, he should perform almost all of the same actions as the cat does. Except for two things:
    1. NOTE: Have the dog "go to position x:120 y:0".
    2. NOTE: Have the dog "set size" to 35%.
e. Create the following "go" script for the dog:




    1.
    2. What does this script do?

5. Select the Text object, by clicking on the Text icon in the Sprite window.
    a. Just like the cat and the dog, the Text object is going to need 4 scripts.
        1. NOTE: The text object should be displayed at position x:0 y:0.
        2. NOTE: The text object should be displayed at 100% size.
    b. Create the "green flag" script for the text object.
        1. Note: It should show the title costume.
    c. Create the "stop" script for the text object.
        1. Note: It should show the "continue" screen.
    d. Create the "setup" script for the text object:
        1. Note: For each of the 4 levels, it should show the appropriate costume.
    e. Create the "go" script for the text object:
        1. Note: If it is level 0 or 99 it should be shown, otherwise it should be hidden.

# IV. Run the game

1. Does it work?
2. Is it fun?

# V. Challenge Exercises

1. Add sounds.
2. Add a score variable and a method by which to keep score.
3. Add a difficulty setting.
4. Create a button object. Have the game change level based on whether or not the button is pushed.