# PlayerHud.cs Script

Declare a dictionary for UI elements. Its key should be the name of that element. The value should be the position we want the arrow to appear at.

private Dictionary<string, Vector2> \_arrowPositions;

## public void PointArrowTo(string element)

Take in a string of the UI element we’re interested in. Use \_arrowPositions to move the arrow to that position.

## public void SetTutorialText(string message)

Take in a string message. Set the textbox to store this message now.

## *Test*

*Check it works so far by calling these functions from the Start() method. Try once using something like:*

*PointArrowTo(“timeline”);*

*SetTutorialText(“This is the timeline”);*

*Run it, make sure the arrow and message have been changed. Then change those function calls to something different and repeat, e.g.*

*PointArrowTo(“team name”);*

*SetTutorialText(“This is your team”);*

# Tutorial.cs Script

## State Class

Create a new class called State. It should store at least the following public variables:

public string Message;

public string ElementToPointTo;

public KeyCode InputTrigger;

It needs a constructor to initialise the variables.

## Tutorial Class

Declare a list of States as a private variable and store the index of the current state:

private List<State> \_states;

private int \_currentState;

### void Start()

Initialise the list, set the current state to 0 and call the CreateStates function.

### private void CreateStates()

Instantiate each State object and append it to the list in order. For example:

State newState = new State(“Press 1 to time travel”, “left button”, KeyCode.Alpha1);

\_states.Add(newState);

Check out the documentation for a list of keycodes:

<https://docs.unity3d.com/ScriptReference/KeyCode.html>

### void Update()

Check for user input with GetKeyDown(). If it matches the InputTrigger of the current State in \_states:

1. Move onto the next state by incrementing \_currentState.
2. Call SetArrowTo() and SetTutorialText() in PlayerHud to update the changes, passing in the values of the current State as parameters.

## *Test*

*Check it works so far by running the game and pressing the correct buttons that should trigger the tutorial to move onto the next state each time. You might get an error when the \_currentState index moves beyond the end of the list and will need to think of a way to fix it.*