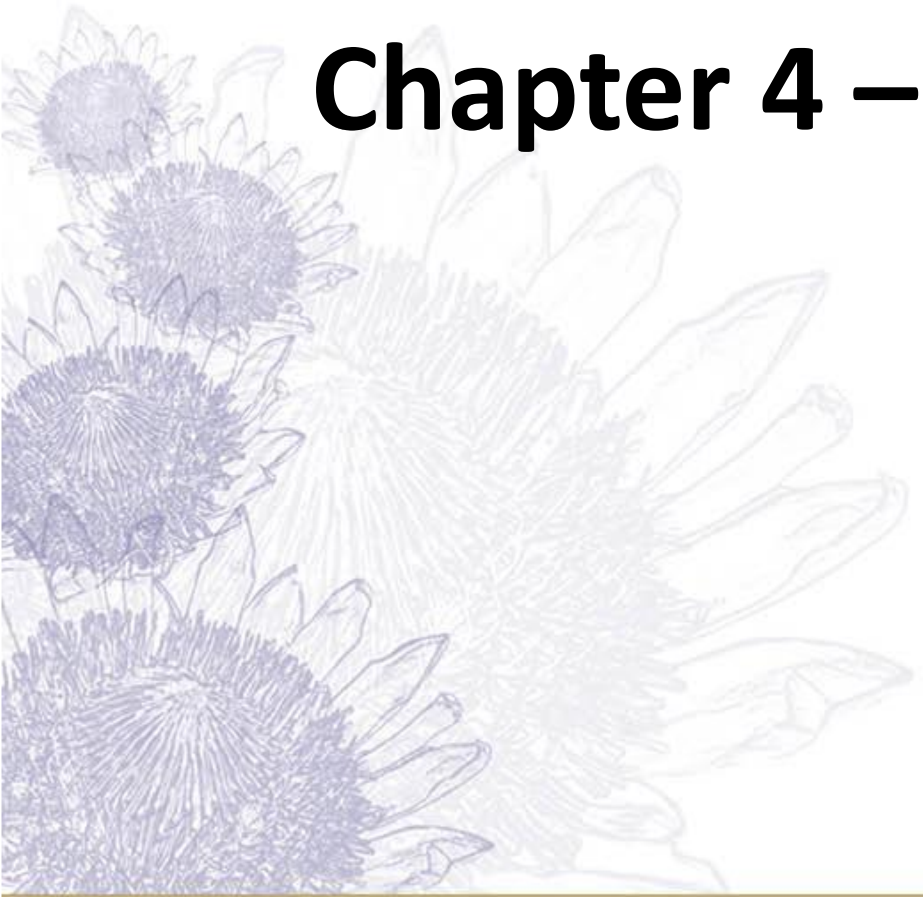

Chapter 4 – Timer and Random

VRA 705

By Kessel Okinga and Mfundo Andrew

Outline

- Timer
- Random

Timer overview

A timer is a tool that can be used to measure the duration between two events such as the start and end time of a game level. In Unity, timer is frequently used to track the amount of time that has passed during game play or other actions.

Timer can be classified into three types:

- ❖ **Countdowns**: timer can be used to create countdowns. For example: New Year Countdown https://codepen.io/nabinbhatt/pen/vYaOpwZ
- ❖ **Time-based scoring**: timer can be used to score a player based on how quickly they complete a task. For example quiz with timer https://codepen.io/harunpehlivan/pen/bGeOPye
- ❖ **Time-based animations**: timer can be used to trigger animations or events after a certain amount of time has passed. For example: Animate objects relative to time instead of frame-based increments

  https://codepen.io/greypants/pen/krmKVq

Countdown

Declare a variable to hold the starting time of the timer

```
public float starter= 60f; // represents a time value
```

Update() method can be used to update the timer every frame.

```
void Update() {
 starter -= Time.deltaTime; //  subtract the time since the last frame (Time.deltaTime) from the start time
    if (starter <= 0.0f) {
          // Do something when the timer runs out
    }
}
```

NB: f refers to a float data type.

Countdown with UI

Declare a Text variable to display the countdown timer on UI

```
 public Text count_down_time_text;
```

Use the Start() method to initialize the value of time and display a formatted countdown timer

```
 void Start()
    {
        count_down_time_text.text = "Remaining time is : " + Mathf.Floor(starter / 60).ToString("00") + ":" +
Mathf.Floor(starter % 60).ToString("00");
    }
```

Countdown with UI (Cont’d)

Explanation:

Mathf.Floor is a function round a given number down to the nearest integer value. For instance, if the original number is a decimal such as 5.12, the result would be rounded down to 5.

Mathf.Floor(starter / 60).ToString("00") calculates the number of minutes, rounds it down to the nearest integer, and formats it as a two-digit string using for example "00".

Mathf.Floor(starter % 60).ToString("00") calculates the number of seconds in, rounds it down to the nearest integer, and formats it as a two-digit string using for example "00".

Create countdown with UI (Cont’d)

Define a boolean variable time_ended and assign a false value.

```
private bool time_ended = false; // to determine whether the countdown time has finished or not.
```

Update() method can be used to update the timer every frame.

```
if (!time_ended)
      {
            starter -= Time.deltaTime; // set the countdown time in seconds
            count_down_time_text.text = "Countdown: " + Mathf.Floor(starter / 60).ToString("00") + ":" + Mathf.Floor(starter % 60).ToString("00");
            if (starter  <= 0)
            {
              time_ended= true;
              // Do something when the countdown is over
            }
      }
```

Random

Random is a process or phenomenon that produces output with an equal likelihood of occurring, without any predictable order.

Using the Random class, it is easy to generate random data (such as random positions for game objects).

Random Value

The statement *public static float value;* Returns a random float within (0.0..1.0) (range is inclusive). The Random.value field in Unity returns a random float between 0.0 (inclusive) and 1.0 (exclusive).

The following code uses the Random.value to generate random float values to create a random color:

```
    void Start()
    {
        Color randomColor = RandomColor();
    }

    Color RandomColor()     // Generate a random color value.
    {
       return new Color(Random.value, Random.value, Random.value); // A different random value is used for each
color component
    }
```

Random Range

The statement *public static float Range(float minInclusive, float maxInclusive);*
Returns a random float within [minInclusive..maxInclusive] (range is inclusive).

The following code uses the Random.Range to generate random float values to display a random question from the question list:

```
public GameObject[] game_object_question;  // declare an array of gameobjects
private int current_index = 0; // declare integer and assign

void Start()
{
    int new_index = Random.Range(0, game_object_question.Length);
    game_object_question[current_index ].SetActive(false); // deactivate old question
    current_index = new_index;
    game_object_question[current_index ].SetActive(true);// activate new question
}
```

Key points

- Timer can be used to measure the duration between two events such as the start and end time of a game level.
- Timer can be broken down into three types, such as countdowns, time-based scoring, and time-based animations.
- The statement *public static float* value returns a random float within the range of (0.0..1.0).
- the statement *public static float Range(float minInclusive, float maxInclusive)* returns a random float within the range of [minInclusive..maxInclusive].

Additional resources

Timer

1. https://www.youtube.com/watch?v=7HudyblojJI&ab_channel=Unity3DSchool
2. https://www.google.com/search?q=timer+using+unity&rlz=1C1GCEU_enZA888ZA888&oq=timer+using+unity&aqs=chrome..69i57j0i22i30l5j69i60l2.296j0j4&sourceid=chrome&ie=UTF-8#kpvalbx=_Y3Y4ZNzgDtDqgQbFxoTQAg_30

Random

1. https://www.youtube.com/watch?v=j8GsxIyJUVw&ab_channel=PitiIT

References

1. Unity Documentation, Scripting API *Timer*, accessed 20th March 2023, <https://docs.unity3d.com/ScriptReference/index.html>
2. Unity Documentation, Scripting API *Random*, accessed 21st March 2023, <https://docs.unity3d.com/ScriptReference/index.html>