

```
1 // -----
2 // Simulering av Dykkerpost
3 // Bachelor Oppgave våren 2022
4 //
5 // Aahed Diyab, Olav Pete
6 // -----
7
8 using Core;
9 using System;
10 using System.Collections;
11 using System.Collections.Generic;
12 using UnityEngine;
13
14 namespace InputController
15 {
16     public class InputController : MonoBehaviour
17     {
18         #region Fields
19
20
21         #endregion
22
23         #region Properties
24         [field: SerializeField]
25         private float keyPressDuration;
26
27         [field: SerializeField]
28         private KeyCode[] unPressedKeys;
29         private KeyCode[] pressedKeys;
30         private float[] timer;
31
32
33         [field: SerializeField]
34         public InputEvent KeySpaceInputEvent { get; set; }
35
36
37
38         #endregion
39
40         #region Public Methods
41
42         #endregion
43
44         #region Private Methods
45
46         private void ToRaise(KeyCode keyCode, int i)
47         {
48             //Lock the pressed key
49             unPressedKeys[i] = 0;
50             pressedKeys[i] = keyCode;
51             timer[i] = Time.time;
52             Debug.Log(keyCode);
53         }
54     }
55 }
```

```
53     }
54
55     private void Timing(KeyCode keyCode, int i, float time)
56     {
57         //Determin if its hold or not
58         if (time - timer[i] < keyPressDuration)
59         {
60             KeySpaceInputEvent.Raise(new InputData { KeyCode = KeyCode.Space, ↗
61                 Holding = false });
62         }
63         else
64         {
65             KeySpaceInputEvent.Raise(new InputData { KeyCode = KeyCode.Space, ↗
66                 Holding = false });
67         }
68
69         //unlock the pressed key
70         //unPressedKeys[i] = keyCode;
71
72         //Raise the event
73
74     }
75
76
77     //private bool Measure(KeyCode kc) { }
78     #endregion
79
80     #region Unity Methods
81
82
83     private void Start()
84     {
85         //Initialize the array
86         pressedKeys = new KeyCode[unPressedKeys.Length];
87         timer = new float[unPressedKeys.Length];
88
89         //for (int i = 0; i < keys.Length; i) InputData a = new InputData();
90
91     }
92
93     private void Update()
94     {
95         for (int i = 0; i < unPressedKeys.Length; i++)
96         {
97             if (Input.GetKeyDown(unPressedKeys[i]))
98             {
99                 ToRaise(unPressedKeys[i], i);
100             }
101         }
102     }
```

```
103         for (int i = 0; i < timer.Length; i++)
104         {
105             if (timer[i] != 0)
106             {
107                 //check if held
108
109                 if (timer[i] == -1)
110                 {
111                     Debug.Log("Held");
112                     if (Input.GetKeyUp(pressedKeys[i]))
113                     {
114                         timer[i] = 0;
115                         unPressedKeys[i] = pressedKeys[i];
116                         pressedKeys[i] = 0;
117                     }
118                 }
119
120
121                 if (Time.time - timer[i] > keyPressDuration)
122                 {
123
124
125                     //Press
126                     if (Input.GetKeyUp(pressedKeys[i]))
127                     {
128                         Debug.Log("press");
129                         timer[i] = 0;
130                         unPressedKeys[i] = pressedKeys[i];
131                         pressedKeys[i] = 0;
132                     }
133                     else
134                     //Hold
135                     {
136                         Debug.Log("hold");
137                         timer[i] = -1;
138                     }
139                 }
140             }
141
142             //    KeySpaceInputEvent.Raise(new InputData {KeyCode
143             //    =KeyCode.Space, Holding = false });
144
145         }
146         #endregion
147     }
148 }
149 }
150 }
```