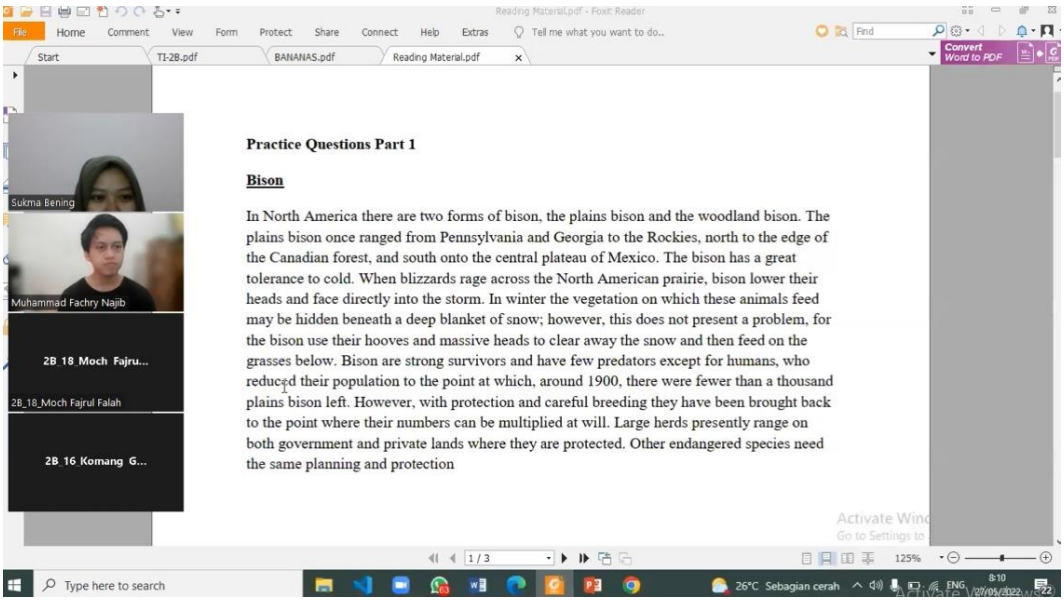
LAMPIRAN

Eksperimen dilakukan dengan didampingi oleh dosen pembimbing, dengan partisipan mahasiswa dari program studi Teknik Informatika Jurusan Teknologi Informasi Politeknik Negeri Malang.

Lampiran 1 Dokumentasi eksperimen



Pada Penelitian awal di gunakan 3 bahan bacaan pada Initial Test berikut bahan bacaan yang digunakan:

Lampiran 2 Bahan Bacaan

The text below is for numbers 1-3

Bison

In North America there are two forms of bison, the plains bison and the woodland bison. The plains bison once ranged from Pennsylvania and Georgia to the Rockies, north to the edge of the Canadian forest, and south onto the central plateau of Mexico. The bison has a great tolerance to cold. When blizzards rage across the North American prairie, bison lower their heads and face directly into the storm. In winter the vegetation on which these animals feed may be hidden beneath a deep blanket of snow; however, this does not present a problem, for the bison use their hooves and massive heads to clear away the snow and then feed on the grasses

below. Bison are strong survivors and have few predators except for humans, who reduced their population to the point at which, around 1900, there were fewer than a thousand plains bison left. However, with protection and careful breeding they have been brought back to the point where their numbers can be multiplied at will. Large herds presently range on both government and private lands where they are protected. Other endangered species need the same planning and protection

1. What is the topic of the passage?

a. The diversity of climates in america
b. National parks of north america
c. Cold-blooded animals of the southwestern desert
d. The endangered grizzly of north America

2. Where would Bison be found during severe winter storm?
   a. Seeking shelter behind boulders
   b. In the open
   c. in caves
   d. Behind trees
   3. It can be concluded from the passage that...
   a. Bison will eventually be extinct
   b. Bison are more fragile than they appear
   c. The Bison population can be controlled
   d. Bison were native to a limited territory

The text below is for numbers 4-6

Ultra Marathon Aimed at Promoting Lake Toba, Samosir

For a second time, North Sumatra is set to host the Samosir Lake Toba Ultra Marathon on Sept. 17. The race will begin at Sitio-tio Hotel and end at the Menara Pandang observatory, said to be the best spot for viewing Lake Toba.

Up to 1,050 runners from 16 countries will join the competition, which is divided into four categories: 50 kilometers, 25 km, 10 km and 5 km, with a winning prize of 125 million rupiah (US$9,468). Several government officials will reportedly join the race, including Coordinating Maritime Affairs Minister Luhut Binsar Pandjaitan, Tourism Minister Arief Yahya, Investment Coordinating Board (BKPM) chairman Thomas Lembong and Financial Services Authority (OJK) chairman Muliaman Hadad.

H3 is mostly run by people working in the financial industry. Around 40 percent of the members are of North Sumatra's Batak ethnic group. In addition to the ultra marathon, H3 has held other sport events this year to promote tourism in the region, such as cycling event Toba GranFondo, a fun walk and golf competition. (kes)

4. Where is the best spot for viewing Lake Toba?
   a. Sitio-tio Hotel
   b. Hilton Hotel
   c. Menara Pandang observatory
   d. Samosir Island

5. What does the text mainly discuss?

a. There will be a running competition in North Sumatra.

b. There will be a cultural exhibition in Toba Lake.

c. Ultra marathon will be held to promote Toba Lake.

d. Government officials will join ultra marathon around Toba Lake.

6. Which one of the following government officials is not mentioned in the text?

a. Coordinating Maritime Affairs Minister

b. Education and Culture Minister

c. Investment Coordinating Board (BKPM) chairman

d. Financial Services Authority (OJK) chairman

7. What were the other sport events held to promote tourism

a. Toba GranFondo, fun walk, and golf competition.

b. Tennis competition, fun run, and cycling event.

c. Cycling event, fun run, and golf competition.

d. Cyvling event, Toba GranFondo and tennis competition.

The text below is for numbers 8-10

The Bermuda Triangle

The Bermuda Triangle occupies a disturbing and almost unbelievable place in the world's catalog of unexplained mysteries. More than a hundred planes and ships have vanished in this area into the air since 1945, and more than a thousand lives have been lost, without a single body or even a piece wreckage from the vanishing planes or ships having been found. Many of the planes concerned have vanished while in normal radio contact with their base until the very moment of their disappearance, while others have radioed the most extraordinary messages, implying that they could not get their instruments to function, that their compasses were spinning, that the sky had turned yellow and hazy on a clear day, and the ocean, which was calm nearby, didn't look right without further clarification of what was wrong.

8. One can infer from the reading that…

a. The wreckages of some ships and planes have been found in the Bermuda Triangle

b. The number of incidents involving lost ships is no larger than that of any other heavily traveled region of the world.

c. The ships. And the planes couldn't contact their base due to the lack of equipment.

d. The first mention of disappearances in the area was made in 1945.

9. It is pointed out in the passage that…

a. Thousands of people lost their lives in the Bermuda Triangle in 1945.

b. The Bermuda Triangle mystery was solved in 1945

c. Most of the missing planes could contact with their base by their own special means until the very moment of disappearing.

d. The ocean floor near Bermuda, highly unexplored, is host to many strange phenomena.

10. The reading mainly deals with…

a. Why so many ships and planes disappear in the Bermuda Triangle.

b. The mysterious disappearances of ships and planes in the Bermuda Triangle.

c. The frequency of the disappearances in the Bermuda Triangle.

d. The unpredictable weather conditions in the Bermuda Triangle.

Berikut teks bacaan yang digunakan dalam minggu kedua untuk Pre-Test dan Pre-Test, dengan keseluruhan 10 soal.

COMPUTERS

Nowadays, computers are used for many different kinds of work, offices, banks, factories, hospitals, universities, and schools. Their use is becoming more widespread as cheaper and smaller computers become available. There are three main kinds of computers in use today: the mainframe, the minicomputer and the microcomputer. What are computers and what can they be used for?

Computers are electronic machines that process information. They can accept information in the form of letters and figures, known as the input data. This can be put in by various means including keyboard, tape or cards. This data is processed according to a set of instructions called a program, and the results of this program, the output data can be printed out or shown on a screen.

All the processing is done by a series of arithmetic and logical operations, such as addition, subtraction and deciding whether one number is greater than another. The computer itself is known as the hardware, in contrast to the programs which are the software.

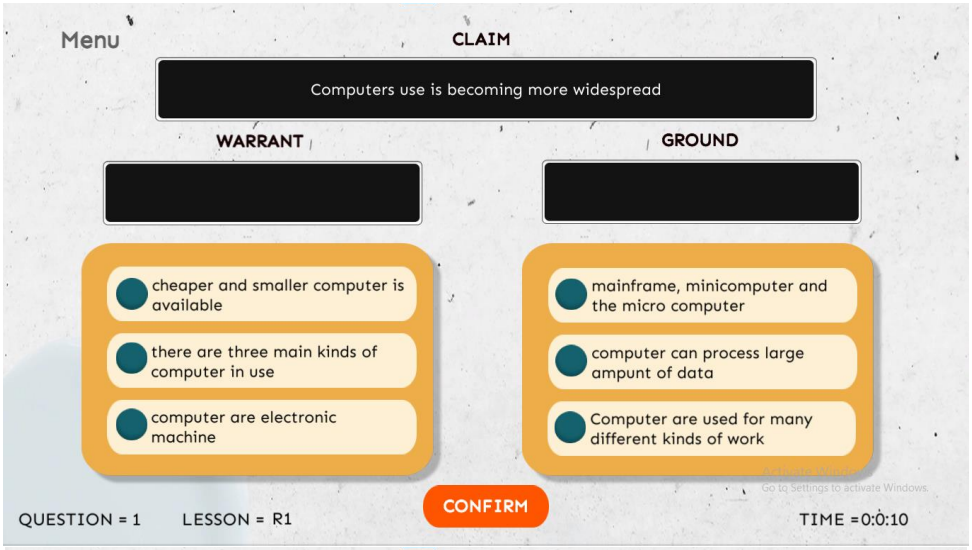
Computers can process large amounts of data very quickly, and this is why they are so useful. They can process different types of data, too. A scientist or engineer for example may use a computer to do numerical calculations. A businessman may want to analyze a list of customers or keep a record of how much stock he has. An engineer can produce diagrams and plans on a computer.

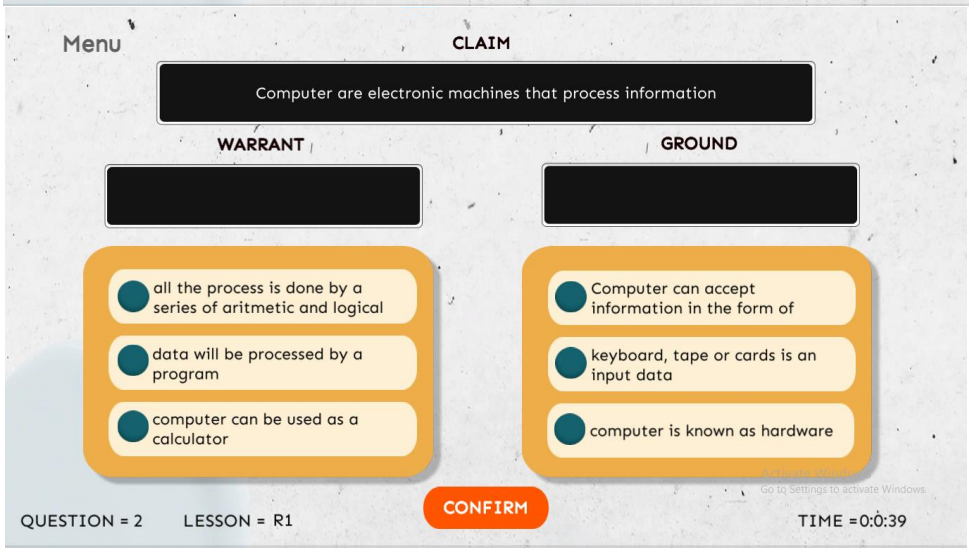
Computers are changing our world and civilization in the same way as the invention of the printing press did in the fifteenth century.

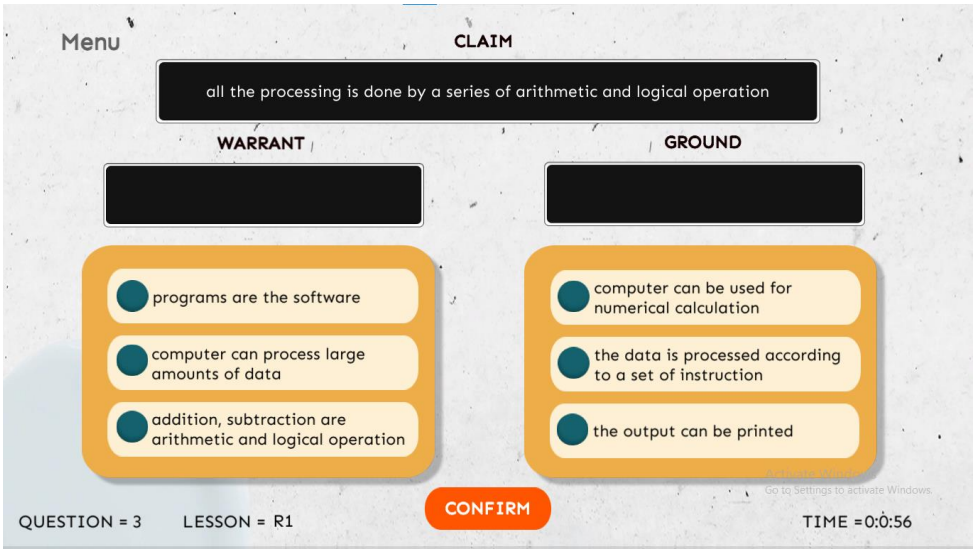
1. Based on the text above, why does computer becoming more widespread?
    1. **It can be applied to numerous jobs**
    2. It becomes flexible and applicable
    3. It functions a tool for everyone to do the job
    4. It is divided into mainframe, minicomputer and microcomputer
2. What does it mean by *"computer is a machine to process information"?*
    1. Computer can be put in by various means including keyboard, tape or cards
    2. **Computer can accept information in the form of letters and figures**
    3. Computer can process according to a set of instructions
    4. Computer can provide an output data to be printed or shown on a screen

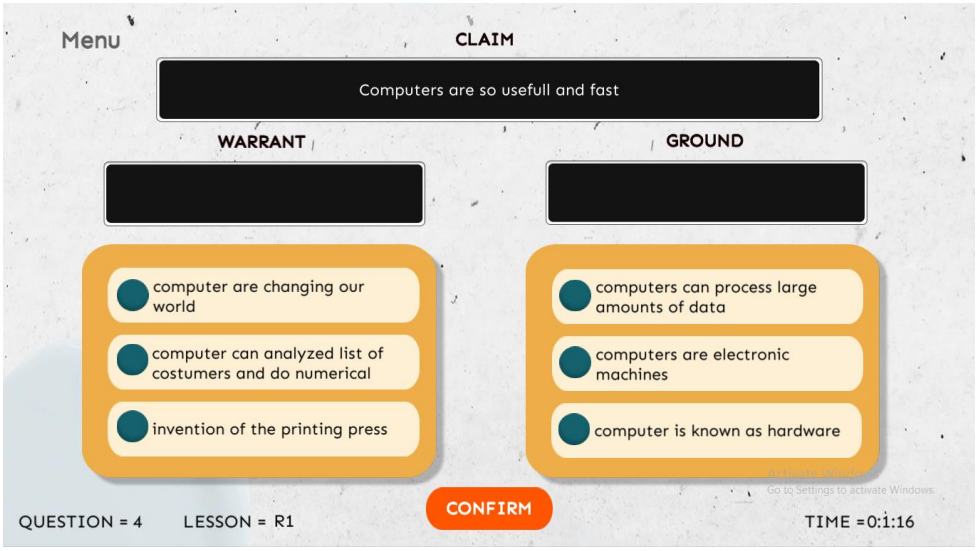
3. What is a computer?
    1. A machine that includes programs
    2. An electronic machine to input data
    3. A series of machine that belongs to electronic devices
    4. **An electronic machine that process information**
4. **<u>This</u>** can be put in by various means including keyboard, tape or cards (line 7). What does the underlined word refer to?
    1. Computer
    2. Information
    3. **Input data**
    4. Output data
5. A series of data of instructions is called ……
    1. **Program**
    2. Hardware
    3. Software
    4. Input data
6. *"Computers can process large amounts of data very quickly"*. What does the following sentence mean?
    1. **It can be used to process different types of data**
    2. It can analyze customer's stock records
    3. It can produce diagrams and plans on a computer
    4. It can be used to do a numerical calculations
7. ….. and deciding whether one number is greater than **<u>another</u>**. What does the underlined word refer to?
    1. Arithmetic
    2. **Logical operations**
    3. Addition
    4. Subtraction
8. The followings are the jobs that are assisted by the computer, except….
    1. **Architect**
    2. Businessman
    3. Scientist
    4. Engineer
9. What is the invention that is similar to computer?
    1. **Printing**
    2. Industrial
    3. Civilization
    4. Telephone
10. What can be inferred from the text above?
    1. Programs can help human's jobs
    2. Computer can assist human 's life
    3. Computer hardware and software can do
    4. **Computer has made a difference in human's life**

Berikut soal yang digunakan oleh Kelompok experiment saat test.


Menu
CLAIM
Computers use is becoming more widespread
WARRANT
GROUND
cheaper and smaller computer is available
there are three main kinds of computer in use
computer are electronic machine
mainframe, minicomputer and the micro computer
computer can process large ampunt of data
Computer are used for many different kinds of work
CONFIRM
QUESTION = 1
LESSON = R1
TIME =0:0:10


Menu
CLAIM
Computer are electronic machines that process information
WARRANT
GROUND
all the process is done by a series of aritmetic and logical
data will be processed by a program
computer can be used as a calculator
Computer can accept information in the form of
keyboard, tape or cards is an input data
computer is known as hardware
CONFIRM
QUESTION = 2
LESSON = R1
TIME =0:0:39


Menu
CLAIM
all the processing is done by a series of arithmetic and logical operation
WARRANT
GROUND
programs are the software
computer can process large amounts of data
addition, subtraction are arithmetic and logical operation
computer can be used for numerical calculation
the data is processed according to a set of instruction
the output can be printed
CONFIRM
QUESTION = 3
LESSON = R1
TIME =0:0:56



Lampiran 3 Kode Program

login.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using LitJson;

public class loginn : MonoBehaviour
{
    public InputField EmailLogin;
    public InputField PassLogin;
    public InputField EmailForget;
    public GameObject loading;
    public Text user;
    public Text nama;
    public Text kelas;
    public Text email;
    public Text induk;
    public Text pass;
    public Text namaSiswa;
    public Text namaGuru;
    public Text warningLog;
    public Text warningForg;
    public GameObject LoginForm;
    public GameObject ForgetForm;
    public GameObject ForgetDetail;
    public GameObject MenuSiswa;
    public GameObject MenuGuru;


    //host url global variable
    public static string host_url = "http://localhost/viatmap1/";
    // Start is called before the first frame update
    void Start()
```

```
    {

    }

    // Update is called once per frame
    void Update()
    {

    }

    IEnumerator login(string _user, string _pass){
        string url = host_url+"login.php";
        var form = new WWWForm();
        form.AddField("email",_user);
        form.AddField("pass", _pass);
        loading.SetActive(true);
        var download = new WWW(url,form);

        yield return download;
        loading.SetActive(false);
        Debug.Log(download.text);
        JsonData bacaData;
        bacaData = JsonMapper.ToObject(download.text);
        Debug.Log(bacaData["status"]);

        if ("" + bacaData["status"] == "berhasil")
        { if ("" + bacaData["kelas"] == "guru")
        {
            Debug.Log("Masuk guru");
            namaGuru.text = ""+bacaData["nama"];
            MenuGuru.SetActive(true);
            LoginForm.SetActive(false);

        }
        else {
            Debug.Log("Masuk siswa");
            namaSiswa.text = ""+bacaData["nama"];
            MenuSiswa.SetActive(true);
            LoginForm.SetActive(false);
        }
        }
        else
        {
            warningLog.text = ("Email dan Password salah");
        }
    }

    IEnumerator forget(string _email){
        string url = host_url + "forget.php";
        var form = new WWWForm();
        form.AddField("email",_email);
        loading.SetActive(true);
        var download = new WWW(url,form);
```

```
        yield return download;
        loading.SetActive(false);
        Debug.Log(download.text);
        JsonData bacaData;
        bacaData = JsonMapper.ToObject(download.text);
        Debug.Log(bacaData["status"]);

        if ("" + bacaData["status"] == "berhasil")
        {
            Debug.Log("bisa");
            user.text = ""+bacaData["id"];
            nama.text = ""+bacaData["nama"];
            kelas.text = "" + bacaData["kelas"];
            induk.text = ""+bacaData["induk"];
            email.text = ""+bacaData["email"];
            pass.text = ""+bacaData["pass"];
            ForgetDetail.SetActive(true);
            ForgetForm.SetActive(false);

        }
        else
        {
            warningForg.text = ("Email Tidak Terdaftar");
        }
    }

    public void btnLogin (){
        if (EmailLogin.text == "" || PassLogin.text == "")
        {
            warningLog.text = "semua field harus di isi";
        }
        else {

            StartCoroutine(login(EmailLogin.text, PassLogin.text));
            warningLog.text = "";

        }
    }

    public void btnForgt (){
        if (EmailForget.text == "")
        {
            warningForg.text = "isikan email anda";
        }
        else {

            StartCoroutine(forget(EmailForget.text));
            warningForg.text = "";

        }
    }
```

```
    public void forget(){
        ForgetForm.SetActive(true);
        LoginForm.SetActive(false);
        ForgetDetail.SetActive(false);
        EmailForget.text = "";
    }

    public void backLog(){
        ForgetDetail.SetActive(false);
        LoginForm.SetActive(true);
        EmailLogin.text="";
        PassLogin.text="";
    }
}
```

dopractice.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.Networking;
using LitJson;
public class dopractice : MonoBehaviour
{

    public Text lesson;
    public Text number;
    public Text time;
    public Text email;
    public Text claim;
    public Text war1;
    public Text war2;
    public Text war3;
    public Text gnd1;
    public Text gnd2;
    public Text gnd3;
    public Text max_num;
    public GameObject popup;
    public GameObject menu;
    public GameObject practice;
    public GameObject benar;
    public GameObject salah;
    public GameObject cancel;
    public GameObject loading;

    private string response;
    private string host_url = loginn.host_url;
    // Start is called before the first frame update

    IEnumerator getlesson(string _email, string _lesson,string
_number)
    {
        WWWForm form = new WWWForm();
```

```
        string url = host_url+"cek_do_practice.php";
        form.AddField("email", _email);
        form.AddField("lesson", _lesson);
        form.AddField("latihan", _number);
        loading.SetActive(true);
        UnityWebRequest req = UnityWebRequest.Post(url, form);

        yield return req.SendWebRequest();
        loading.SetActive(false);
        response = req.downloadHandler.text;
        Debug.Log(response);

        if (response == "kosong ")
        {
            popup.SetActive(true);
        }
        else
        {
            JsonData bacaData;
            bacaData = JsonMapper.ToObject(response);
            lesson.text = "" + bacaData["lesson"];
            claim.text = "" + bacaData["claim"];
            war1.text = "" + bacaData["warrant0"];
            war2.text = "" + bacaData["warrant1"];
            war3.text = "" + bacaData["warrant2"];
            gnd1.text = "" + bacaData["ground0"];
            gnd2.text = "" + bacaData["ground1"];
            gnd3.text = "" + bacaData["ground2"];
            max_num.text = "" + bacaData["max_num"];
            Debug.Log(max_num.text);
            practice.SetActive(true);
            menu.SetActive(false);
            benar.SetActive(false);
            salah.SetActive(false);
            cancel.SetActive(false);

        }

    }
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {

    }
    public void getlessonbtn()
    {
        Debug.Log(email.text);
        int num = 1;
        number.text = num.ToString();
```

```
        StartCoroutine(getlesson(email.text, lesson.text,
number.text));
    }

    public void ok()
    {
        popup.SetActive(false);
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;
using UnityEngine.UI;
using UnityEngine.Networking;
using LitJson;
using System;

public class dragdrop : MonoBehaviour, IPointerDownHandler,
IBeginDragHandler, IEndDragHandler, IDragHandler, IDropHandler,
IPointerUpHandler
{
    [SerializeField] private Canvas canvas;
    private RectTransform rectransform;
    private CanvasGroup canvasGroup;
    private GameObject ans_img;
    public GameObject warrpopup;
    public GameObject imgpopup;
    public GameObject itemslot;
    public GameObject loading;
    public InputField warrant;
    public Image img_desc;
    public Text lesson;
    public Text number;
    public Text ans;
    public Text img_war1;
    public Text img_war2;
    public Text img_war3;
    public Text img_gnd1;
    public Text img_gnd2;
    public Text img_gnd3;
    public Toggle conf_feature;

    private float startX;
    private float startY;
    private bool insert;
    private bool img;
    public bool move;
    Vector2 item_initialLoc;
    Vector2 slot_initialLoc;
    Vector2 drop;
    Vector2 item_loc1;
    Vector2 item_loc2;
```

```
    Vector2 item_loc3;

    private void Awake()
    {
        rectransform = GetComponent<RectTransform>();
        canvasGroup = GetComponent<CanvasGroup>();
    }
    public void OnBeginDrag(PointerEventData eventData)
    {
        Debug.Log("Begin-Drag");
        canvasGroup.blocksRaycasts = false;
        img = false;
    }

    public void OnDrag(PointerEventData eventData)
    {
        drop = rectransform.anchoredPosition += eventData.delta /
canvas.scaleFactor;
    }

    public void OnEndDrag(PointerEventData eventData)
    {
        Debug.Log("End-Drag");
        canvasGroup.blocksRaycasts = true;
        startX = this.transform.localPosition.x -
slot_initialLoc.x;

        startY = this.transform.localPosition.y -
slot_initialLoc.y;
        if (insert) {
            if (Mathf.Abs(startX) <= 100f && Mathf.Abs(startY) <=
30f)
            {
                rectransform.anchoredPosition = slot_initialLoc;
            }
            else
            {
                rectransform.anchoredPosition = item_initialLoc;
            }
        }
        else
        {
            rectransform.anchoredPosition = item_initialLoc;
        }
        item_loc1 =
GameObject.FindGameObjectWithTag("war1").transform.localPosition;
        item_loc2 =
GameObject.FindGameObjectWithTag("war2").transform.localPosition;
        item_loc3 =
GameObject.FindGameObjectWithTag("war3").transform.localPosition;
        if (item_loc1 != slot_initialLoc && item_loc2 !=
slot_initialLoc && item_loc3 != slot_initialLoc)
        {
            warrant.image.color = Color.white;
```

```
            insert = true;
        }
        else
        {
            if (insert)
            {
                if (conf_feature.isOn)
                {
                    warrpopup.SetActive(true);
                }
                else
                {
                    warrant.image.color = Color.green;
                }
                if (item_loc1 == slot_initialLoc)
                {
                    ans.text = "war0";
                    Debug.Log("war0");
                }
                else if (item_loc2 == slot_initialLoc)
                {
                    ans.text = "war1";
                    Debug.Log("war1");
                }
                else if (item_loc3 == slot_initialLoc)
                {
                    ans.text = "war2";
                    Debug.Log("war2");
                }
            }
            insert = false;

        }

    }

    public void OnPointerDown(PointerEventData eventData)
    {
        Debug.Log("Up-klik");
        img = true;
    }
    public void OnDrop(PointerEventData eventData)
    {
        Debug.Log("On-Drop");
    }


    public void OnPointerUp(PointerEventData eventData)
    {
        Debug.Log("Down-Drop");
    }
    // Start is called before the first frame update
    void Start()
    {
```

```
        item_initialLoc = rectransform.anchoredPosition;
        slot_initialLoc = itemslot.transform.localPosition;
        move = true;

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

dragdrop1.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;
using UnityEngine.UI;
using UnityEngine.Networking;
using LitJson;
using System;

public class dragndrop1 : MonoBehaviour, IPointerDownHandler,
IPointerUpHandler, IBeginDragHandler, IEndDragHandler,
IDragHandler, IDropHandler
{
    [SerializeField] private Canvas canvas;
    private RectTransform rectransform;
    private CanvasGroup canvasGroup;
    private GameObject ans_img;
    public GameObject gndpopup;
    public GameObject itemslot;
    public GameObject imgpopup;
    public GameObject loading;
    public InputField ground;
    public Image img_desc;
    public Text lesson;
    public Text number;
    public Text gnd_ans;
    public Toggle conf_feature;

    private float startX;
    private float startY;
    private bool insert;
    private bool img;
    Vector2 item_initialLoc;
    Vector2 slot_initialLoc;
    Vector2 drop;
    Vector2 item_loc1;
    Vector2 item_loc2;
    Vector2 item_loc3;
```

```
    private void Awake()
    {
        rectransform = GetComponent<RectTransform>();
        canvasGroup = GetComponent<CanvasGroup>();
    }
    public void OnBeginDrag(PointerEventData eventData)
    {
        Debug.Log("Begin-Drag");
        canvasGroup.blocksRaycasts = false;
        img = false;
    }

    public void OnDrag(PointerEventData eventData)
    {
        drop = rectransform.anchoredPosition += eventData.delta /
canvas.scaleFactor;
    }

    public void OnEndDrag(PointerEventData eventData)
    {

        Debug.Log("End-Drag");
        canvasGroup.blocksRaycasts = true;
        startX = this.transform.localPosition.x -
slot_initialLoc.x;
        startY = this.transform.localPosition.y -
slot_initialLoc.y;
        if (insert)
        {
            if (Mathf.Abs(startX) <= 100f && Mathf.Abs(startY) <=
30f)
            {
                rectransform.anchoredPosition = slot_initialLoc;
            }
            else
            {
                rectransform.anchoredPosition = item_initialLoc;
            }
        }
        else
        {
            rectransform.anchoredPosition = item_initialLoc;
        }

        item_loc1 =
GameObject.FindGameObjectWithTag("gnd1").transform.localPosition;
        item_loc2 =
GameObject.FindGameObjectWithTag("gnd2").transform.localPosition;
        item_loc3 =
GameObject.FindGameObjectWithTag("gnd3").transform.localPosition;
        if (item_loc1 != slot_initialLoc && item_loc2 !=
slot_initialLoc && item_loc3 != slot_initialLoc)
        {
            ground.image.color = Color.white;
```

```
            insert = true;
        }
        else
        {
            if (insert)
            {
                if (conf_feature.isOn)
                {
                    gndpopup.SetActive(true);
                }
                else
                {
                    ground.image.color = Color.green;
                }
                if (item_loc1 == slot_initialLoc)
                {
                    gnd_ans.text = "gnd0";
                    Debug.Log("gnd0");
                }
                else if (item_loc2 == slot_initialLoc)
                {
                    gnd_ans.text = "gnd1";
                    Debug.Log("gnd1");
                }
                else if (item_loc3 == slot_initialLoc)
                {
                    gnd_ans.text = "gnd2";
                    Debug.Log("gnd2");
                }
            }

            insert = false;
        }

    }

    public void OnPointerDown(PointerEventData eventData)
    {
        Debug.Log("down-klik");
        img = true;
    }
    public void OnPointerUp(PointerEventData eventData)
    {

        Debug.Log("up-klik");

    }
    public void OnDrop(PointerEventData eventData)
    {
        Debug.Log("On-Drop");

    }
    // Start is called before the first frame update
    void Start()
```

```
    {
        item_initialLoc = rectransform.anchoredPosition;
        slot_initialLoc = itemslot.transform.localPosition;
    }

    // Update is called once per frame
    void Update()
    {

    }
}
```