

<유니티 project 만들기>

- 버전은 2019.4.17.f1 버전으로 사용한다.
 - 개인 PC의 검색창 -> unity hub -> 프로젝트 -> 새로 생성 -> 템플릿(3D), 프로젝트 이름(HelloBot_Unity), 저장 위치(HelloBot 폴더 선택)
1. UI > Canvas와 Scroll View 추가(Canvas에 Scroll View가 속하도록)
 2. Scroll View > viewpoint > Content의 Anchor를 left-bottom으로 설정
 3. Content에서, Add Component > Content Size Fitter, Vertical Layout Group추가.
 4. 좌측 하단에 Assets 폴더에 Prefabs 폴더 생성. 이후 방금까지의 'Text' gameobject를 prefabs 폴더에 드래그하여 prefab으로 저장.
 5. Canvas에서 우클릭하여 UI > InputField 추가. 다시 우클릭하여 UI > Button 추가. Button의 text는 'Send'로 설정.
 6. <https://syn.co.in/download/Syn.Bot.Demo.UnityPackage> 다음 주소에서 유니티와 Syn Bot Demo를 연결 시켜줄 패키지 다운.
 7. 좌측 상단에 Assets > Import Package > Custom Package를 통해 다운받은 패키지를 import한다.
 8. Hierarchy에서 ctrl + shift + N 눌러서 GameObject 생성 후 이름을 ChatManager로 변경.
 9. Assets 폴더에 Scripts 폴더 생성 후 거기서 [ChatManager] C# Script를 생성
 10. 스크립트를 아래와 같이 채운다.

```
1  using Syn.Bot.Oscova;
2  using Syn.Bot.Oscova.Attributes;
3  using System;
4  using System.Collections.Generic;
5  using UnityEngine;
6  using UnityEngine.UI;
7
8  public class Message
9  {
10     public string Text;
11     public Text TextObject;
12     public MessageType MessageType;
13 }
14
15 public enum MessageType
16 {
17     User, Bot
18 }
19
```

```
20 public class ChatManager : MonoBehaviour
21 {
22     OscovaBot MainBot;
23
24     List<Message> Messages = new List<Message>();
25
26     public GameObject chatPanel, textObject;
27     public InputField chatBox;
28     public Color UserColor, BotColor;
29
30     // Start is called before the first frame update
31     void Start()
32     {
33     }
34
35     // Update is called once per frame
36     void Update()
37     {
38     }
39
40 }
41
42
```

```
42
43
44 public void AddMessage(string messageText, MessageType messageType)
45 {
46     if (Messages.Count >= 25)
47     {
48         //Remove when too much.
49         Destroy(Messages[0].TextObject.gameObject);
50         Messages.Remove(Messages[0]);
51     }
52
53     var newMessage = new Message { Text = messageText };
54     var newText = Instantiate(textObject, chatPanel.transform);
55
56     newMessage.TextObject = newText.GetComponent<Text>();
57     newMessage.TextObject.text = messageText;
58     newMessage.TextObject.color = messageType == MessageType.User ? UserColor : BotColor;
59     Messages.Add(newMessage);
60 }
```

```

62 public void SendMessageToBot()
63 {
64     var userMessage = chatBox.text;
65
66     if (!string.IsNullOrEmpty(userMessage))
67     {
68         Debug.Log($"OscovaBot:[USER] {userMessage}");
69         AddMessage($"User: {userMessage}", MessageType.User);
70         var request = MainBot.MainUser.CreateRequest(userMessage);
71         var evaluationResult = MainBot.Evaluate(request);
72         evaluationResult.Invoke();
73
74         chatBox.Select();
75         chatBox.text = "";
76     }
77 }

```

11. ChatManager의 Start()를 아래와 같이 코드 변경

```

// Start is called before the first frame update
void Start()
{
    try
    {
        MainBot = new OscovaBot();
        OscovaBot.Logger.LogReceived += (s, o) =>
        {
            Debug.Log($"OscovaBot: {o.Log}");
        };

        MainBot.Dialogs.Add(new BotDialog());
        MainBot.Trainer.StartTraining();

        MainBot.MainUser.ResponseReceived += (sender, evt) =>
        {
            AddMessage($"Bot: {evt.Response.Text}", MessageType.Bot);
        };
    }
    catch (Exception ex)
    {
        Debug.LogError(ex);
    }
}

```

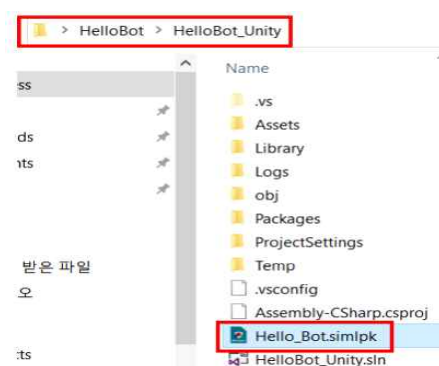
12. BotDialog 클래스 생성

```

public class BotDialog : Dialog
{
    [Expression("Hello Bot")]
    public void Hello(Context context, Result result)
    {
        result.SendResponse("Hello User!");
    }
}

```

13. Simlpk 파일을 유니티에 이식(+ ChatManager 스크립트 수정) 후 실행하기



```

23 //public class BotDialog : Dialog
24 //
25 // [Expression("Hello Bot")]
26 // public void Hello(Context context, Result result)
27 // {
28 //     result.SendResponse("Hello User!");
29 // }
30 //

```

```

32 public class ChatManager : MonoBehaviour
33 {
34     //OscovaBot MainBot:
35     SimiBot simiBot;
36
37     List<Message> Messages = new List<Message>();
38
39     public GameObject chatPanel, textObject;
40     public InputField chatBox;
41     public Color UserColor, BotColor;
42
43     // Start is called before the first frame update
44     // Unity 메시지 참조 0개
45     void Start()
46     {
47         try
48         {
49             //MainBot = new OscovaBot();
50             //OscovaBot.Logger.LogReceived += (s, o) =>
51             //{
52                 // Debug.Log($"OscovaBot: {o.Log}");
53             //};

```

```

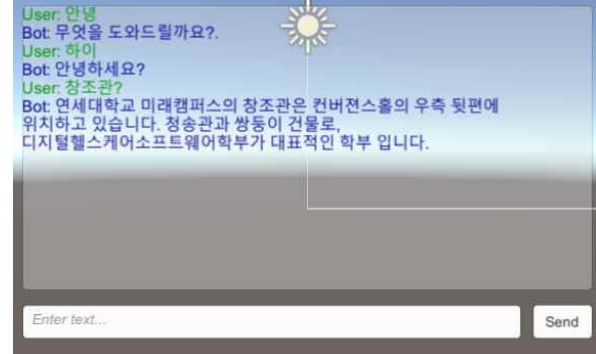
54     }
55
56     //MainBot Dialogs.Add(new BotDialog());
57     //string path = "HelloBot";
58     //TextAsset txtAsset = Resources.Load<TextAsset>(path);
59     //Debug.Log(txtAsset);
60     //var txtFile = txtAsset.text;
61     //MainBot.ImportWorkspace("txtFile");
62     //MainBot.Trainer.StartTraining();
63
64     //MainBot.MainUser.ResponseReceived += (sender, evt) =>
65     //{
66         // AddMessage($"Bot: {evt.Response.Text}", MessageType.Bot);
67     //};
68
69     simiBot = new SimiBot();
70     simiBot.PackageManager.LoadFromStrings(File.ReadAllText("Hello_Bot.simph"));
71     SimiBot.Logger.LogReceived += (s, o) =>
72     {
73         Debug.Log($"SimiBot: {o.Log}");
74     };
75
76     catch (Exception ex)
77     {
78         Debug.LogError(ex);
79     }

```

```

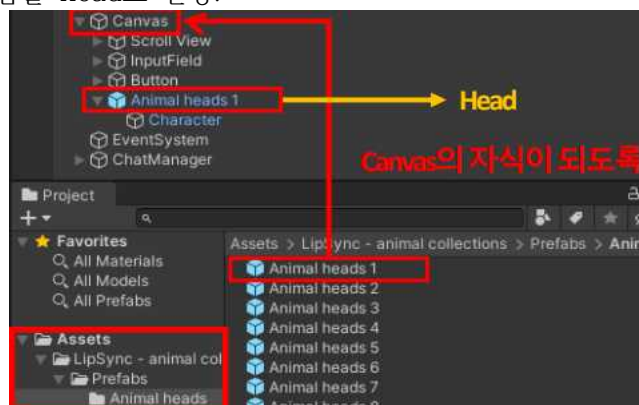
109 public void SendMessageToBot()
110 {
111     var userMessage = chatBox.text;
112
113     if (!string.IsNullOrEmpty(userMessage))
114     {
115         Debug.Log($"SIMBot: [USER] {userMessage}");
116         AddMessage($"User: {userMessage}", MessageType.User);
117         //var request = MainBot.MainUser.CreateRequest(userMessage);
118         //var evaluationResult = MainBot.Evaluate(request);
119         //evaluationResult.Invoke();
120
121         var result = simiBot.Chat(userMessage);
122         AddMessage($"Bot: {result.BotMessage}", MessageType.Bot);
123
124         chatBox.Select();
125         chatBox.text = "";
126     }
127 }
128
129

```



<아바타 적용시키기>

1. 아바타 적용전 지금껏 만든 Unity를 백업(Export)한다. Assets > Export Package
2. HelloBot_Animation 이름의 새로운 Unity Project를 생성.
3. Assets > Import Package > Custom Package > All > Import 백업한 패키지 불러오기
4. 좌측 상단 Window > Asset Store > LipSync-animal collections 검색, 구매 및 Import
5. animal collections 폴더 > Prefabs > Animal heads > Animal heads1, Animal heads1의 이름을 head로 변경.



6. Pronunciation_Kor_En.cs를 Assets 폴더 내로 위치하기.(업로드 된 자료에 있다.)
7. Pronunciation_Kor_En.cs를 ChatManager에 드래그하여 Component를 추가한다.

8. ChatManager.cs 수정

```

21 public class ChatManager : MonoBehaviour
22 {
23     SimlBot Chatbot;
24     TextToSpeech TTS;
25     LipSyncManager LSM;
26     Pronunciation_Kor_En PKE;
27
28     List<Message> Messages = new List<Message>();
29
30     public GameObject chatPanel, textObject;
31     public InputField chatBox;
32     public Color UserColor, BotColor;
33
34     void Start()
35     {
36         try
37         {
38             //MainBot = new OscovaBot();
39             // https://developer.syn.co.in/tutorial/bot/siml/tutorial.html
40             Chatbot = new SimlBot();
41             // SIML Chatbot MyUnitySIMLChatbotProject/Knowledge.simlek
42             Chatbot.PackageManager.LoadFromStrings(File.ReadAllText("Hello_Bot.simlek"));
43
44             // Ensure that logs from the bot is redirected to Unity debugger..
45             //OscovaBot.Logger.LogReceived += (s, o) =>
46             //{
47                 Debug.Log($"OscovaBot: {o.Log}");
48             //};
49             SimlBot.Logger.LogReceived += (s, o) =>
50             {
51                 Debug.Log($"SimlBot: {o.Log}");
52             };
53
54             // https://gamedev.stackexchange.com/questions/178053/unity-initializing-classes
55             TTS = GetComponent<TextToSpeech>();
56             LSM = this.transform.Find("Head").gameObject.GetComponent<LipSyncManager>();
57             PKE = GetComponent<Pronunciation_Kor_En>();
58         }
59     }
60
61     public void SendMessageToBot()
62     {
63         // Get the Input Field text value
64         var userMessage = chatBox.text;
65
66         if (!string.IsNullOrEmpty(userMessage))
67         {
68             Debug.Log($"SimlBot: [USER] {userMessage}");
69             AddMessage($"User: {userMessage}", MessageType.User);
70             //var request = MainBot.MainUser.CreateRequest(userMessage);
71             //var evaluationResult = MainBot.Evaluate(request);
72             //evaluationResult.Invoke();
73             var result = Chatbot.Chat(userMessage);
74             Debug.Log(result.BotMessage);
75             string res_str = PKE.InputString(result.BotMessage);
76
77             LSM.Input = res_str;
78             LSM.PutInputIntoQueue = true;
79             TTS.Init(result.BotMessage);
80
81             //LSM.SetInputAndPlay(result.BotMessage);
82             AddMessage($"Bot: {result.BotMessage}", MessageType.Bot);
83
84             chatBox.Select();
85             chatBox.text = "";
86             // if (string.IsNullOrEmpty())
87         }
88     }

```

9. TextToSpeech.cs 수정

```

50 public class TextToSpeech : MonoBehaviour
51 {
52     private string apiUrl = "https://texttospeech.
53     SetTextToSpeech tts = new SetTextToSpeech();
54     public AudioSource audioSource_sound = null;
55     //public AudioSource audioSource_sync = null;
56
57     // Start is called before the first frame update
58     // Unity 메시지 참조 0개
59     void Start()
60     {
61         //Init();
62         //CreateAudio();
63         //audioSource_sound = FindObjectOfType<Aud
64         // https://stackoverflow.com/questions/337
65         //audioSource_sync = GetComponentInChildre
66         //audioSource_sync = this.transform.Find("
67     }
68
69     private void CreateAudio()
70     {
71         // str에 base64 형태로 인코딩된 파일 저장.
72         var str = TextToSpeechPost(tts);
73         GetContent info = JsonUtility.FromJson<GetContent>(str);
74
75         // bytes로 디코딩
76         var bytes = Convert.FromBase64String(info.audioContent);
77
78         // byte array to float array
79         var f = Convert.ToSingle(bytes);
80
81         AudioClip audioClip = AudioClip.Create("audioContent", f.Length, 1, 44100, false);
82         audioClip.SetData(f, 0);
83
84         // https://docs.unity3d.com/ScriptReference/Component.GetComponentInChildren.html
85         //if ((audioSource_sound != null) && (audioSource_sync != null))
86         if (audioSource_sound != null)
87         {
88             audioSource_sound.PlayOneShot(audioClip);
89             //audioSource_sync.PlayOneShot(audioClip);
90         }
91     }

```

10. LipSyncManager.cs 수정

```

44 private void Start()
45 {
46     Input = "";
47     imageComponent = transform.Find("Character").GetComponent<Image>();
48     WaitTimeBetweenCharacters = 0.052f;
49     WaitTimeBetweenWords = 0.08f;
50     WaitTimeBetweenSentences = 0.2f;
51
52     CheckForAllLetters();
53     CheckForDuplicates();
54     AddConversionElements();
55
56     // TODO: These are test inputs. Remove these lines!
57     //_inputQueue.Enqueue("안녕하세요.");
58     //_inputQueue.Enqueue("Have a nice day.");
59     //_inputQueue.Enqueue("Cheers.");
60     //StartCoroutine("ReadInput");
61
62 }

```


11. 최종 결과

