



DES REPORT 2023

AR For Institute Events

Use Of AR For Project Display During Exhibitions

Project by:

Mohammed Hazique Kola 216130011

Akshata Khare 216450009

Guided by:

Prof. Jayesh Patel



DES

Name

Hazique Kola

Akshata Khare

Guide

Prof. Jayesh Pillai



CONTENTS

Declaration
Project Approval
Abstract
Target Audience
Motivation
Incentive/ motivation to use the app
User flow, Wireframe
Explorations
Tibbi character explorations
App Explorations
Tibbi Development
Visual Style
Screens Developement
Technical Implementation snaps
User testing
Future Scope

DECLARATION

I declare that this written submission represents my Ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. Sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated any idea/data/fact/source in my submission.

I understand that any violation of the above will be cause for disciplinary action by the Institute and can also, evoke penal action from the sources which have thus not been appropriately cited or from whom proper permission has not been taken when needed.

Name: **Akshata Khare, Mohammed Hazique Kola**

Roll. No.: **216450009, 216130011**

PROJECT APPROVAL

This Design Exploratory Seminar (DES) project entitled “AR For Institute Events” by Akshata Khare, Roll No. 216450009, Mohammed Hazique Kola, Roll No. 216130011 is approved in partial fulfillment of the requirements for a Master of Design Degree in Communication Design and Industrial design respectively.



Project Guide: **Prof. Jayesh Pillai**

IDC School of Design,
Indian Institute of Technology, Bombay.

ABSTRACT

By employing Augmented Reality, information such as advertisements, posters, and announcements regarding upcoming events, launches, festivals, and more can be communicated without the need for printing posters, pamphlets, and other physical materials that result in waste generation. Our objective through this project is to propose an alternate solution that employs AR to disseminate information, not only reducing waste but also paving the way for further advancements in the digital space.

TARGET AUDIENCE

We started this project with the theme “AR in public spaces”. We further narrowed down to IITB events. Having IITB residents (mainly students, and events are mainly targeted to students) will give us the opportunity to analyse how ready we are to use AR for advertisements.

Assumption:

User group utilize mobile devices to locate and engage with information.

Challenge:

To determine whether users are prepared and at ease utilizing AR as a method of interacting with advertisements.

Can AR generate interest among the target audience?

MOTIVATION

REDUCE WAISTAGE

Numerous posters, banners and pamphlets are printed during college events to spread awareness about the event. There are mainly two types of posters:

- Posters informing about the upcoming event
- Posters used during events for advertising. E.g. sponsors posters in Mood Indigo festival.

Materials like paper, flex banners, sunboard etc., are used for printing and thrown away after use.



USAGE OF BANNERS/ POSTERS/ PAMPLETS

HOSTEL

Places such as lifts, toilet doors, mirrors, mess tables, room doors, bulletin boards. For certain events pamphlets are even slid under the doors.

DEPARTMENT

Majorly, posters are used for event promotion. Sunboard, and paper prints are frequently used for project displays. Banners are hardly used in the college.

IITB CAMPUS

Campus events: Big banners are printed for culturals, tech and sport events withing campus. Events organised by the departments, institute bodies like gender cell, wellness center etc.

Sponsor advertisements: During fests and college events, there are many banners advertising the sponsors.



EVENT POSTERS/ BANNERS WASTAGE @IITB

We did a survey on the wastage generated generated for a IITB institute event E-Submit and waste generated through posters in IDC for events.

Paper decorations
(Printed both sides)

Banners
(Printed both sides)



Image taken during E-Submitt 2023. This location is near cafe 92.



Lot of huge banners we used for decoration purposes.



We noticed old banners being taken down during the event.



Banners had dates written and thus cannot be reused.



We can see waste all banners on the left. Most of these banners promote a event with dates, which cannot be reused later, and thus the banner becomes a waste.

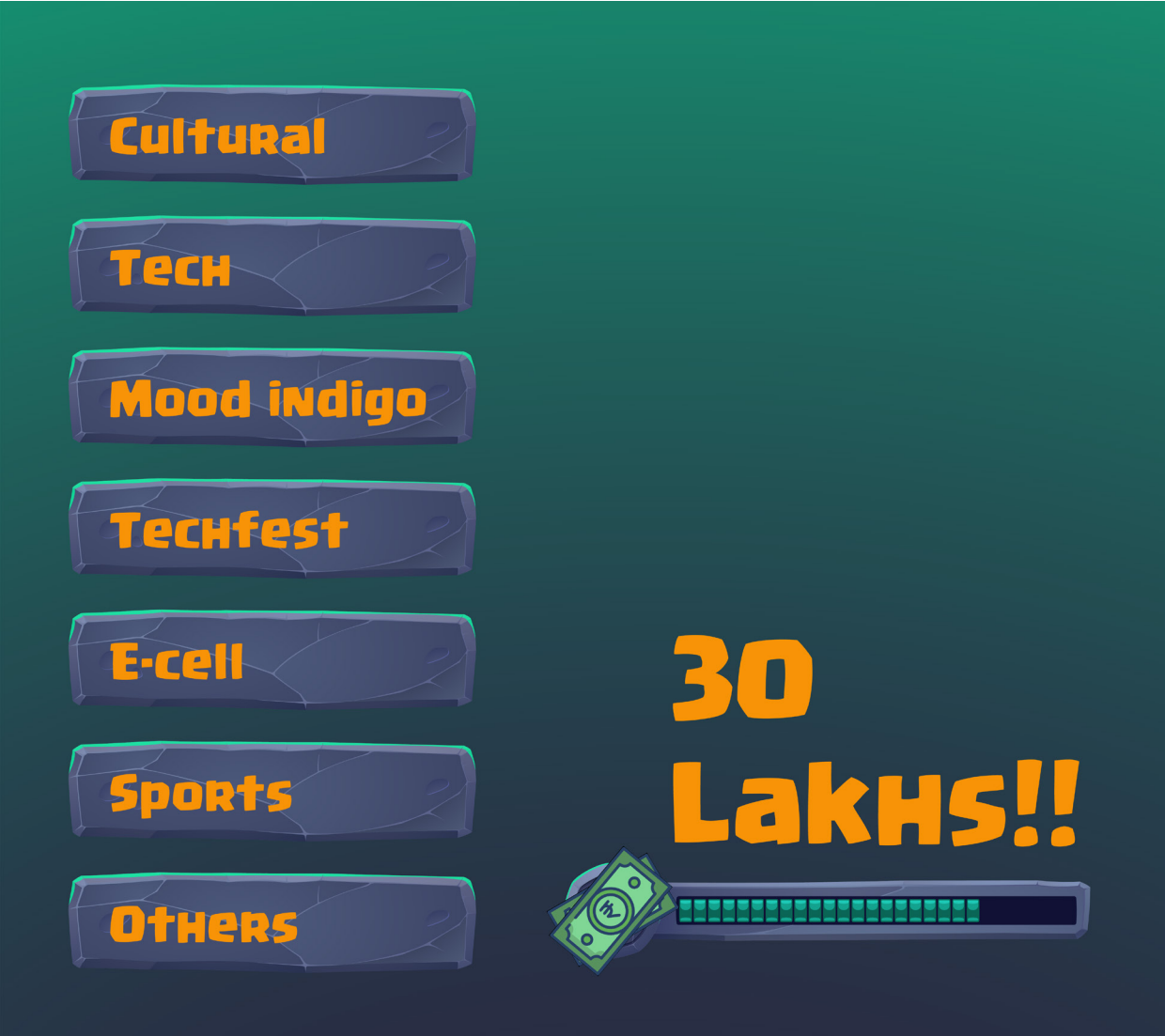


Image of pamphlets taken at H15 hostel. During every event these pamphlets are put up.



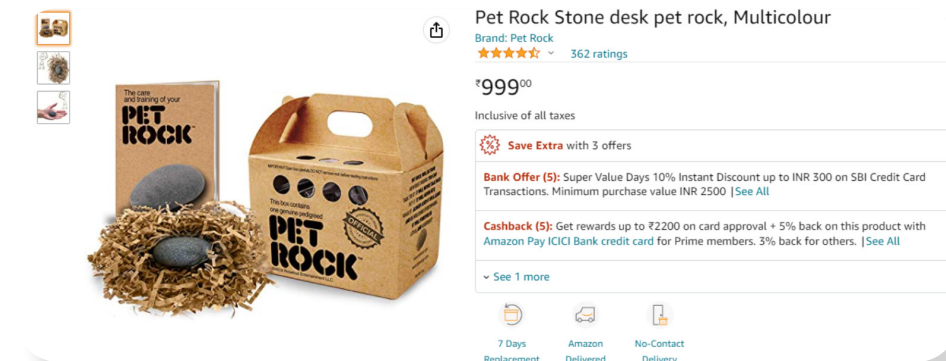
Banner at H15 turn. This is where most of the banners are put up.





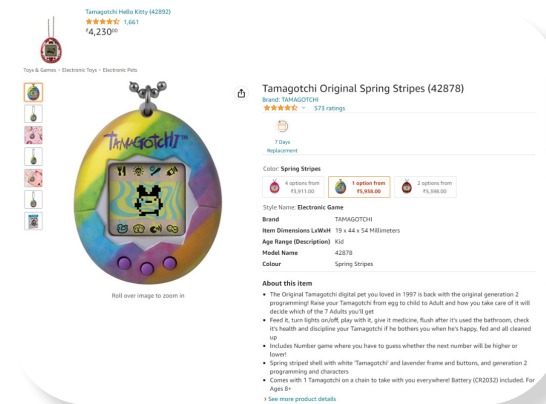
INCENTIVE/ MOTIVATION TO USE THE APP

PEOPLE LIKE HAVING A PET/ BUDDY



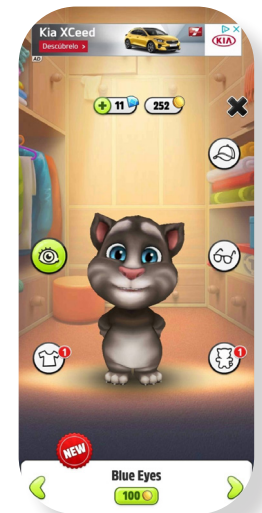
PET ROCK

even a simple thing like rock, when given a name and a manual (to do things) gave a non living thing a personality and people were willing to spend money for it, even though rocks are very common and easily available.



TAMAGOTCHI

people like to have someone to play with and something to call their own.



TALKING TOM

People like to have someone to play/ interact and respond to our actions.

PEOPLE MAKE CONNECTIONS EVEN WITH MACHINES IF THEY RESPOND WITH EMPATHY AND SYMPATHY.



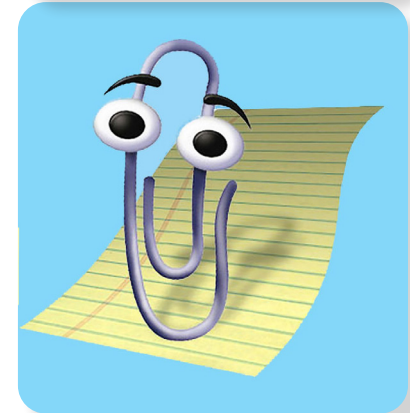
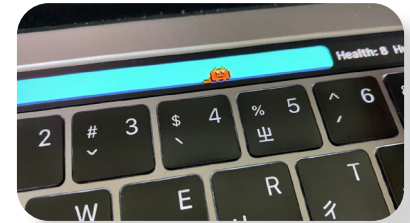
ANIMATED FILM- BEST FRIEND

People feel lonely and things that give them comfort and affection, they tend to get connected with them (behavioral and reflective design).



FILM- HER

They like to keep coming back and keep interacting with the product.
They take help of the OS to perform task and things become easier and faster for them.



DIGITAL COMPANION

Having a character makes the product looks friendly.

**PEOPLE LIKE A STUDY/TASK PARTNER.
(EXTRINSIC MOTIVATIONS)**



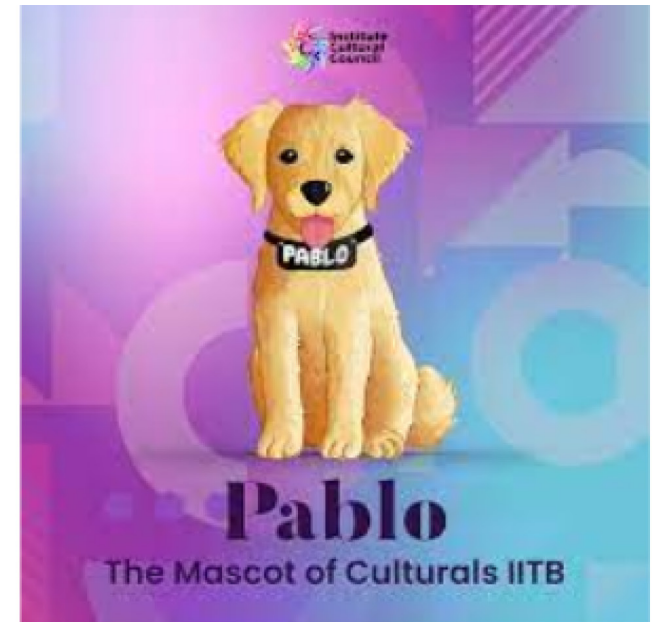
APP- STUDY BUNNY

There are online platforms as well, where people like to study together. Having a companion while working gives extrinsic motivation and a company.

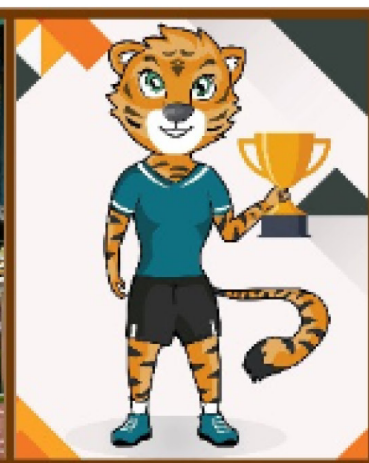
PAST IITB MASCOTS

Before creating our own character, we searched mascots used currently and in the past.

ICC Culturals uses “Pablo” (a dog at IITB campus) as their mascot. Similarly, robots, squirrell, leapord, crocodile can be seen being used as mascots for college events.

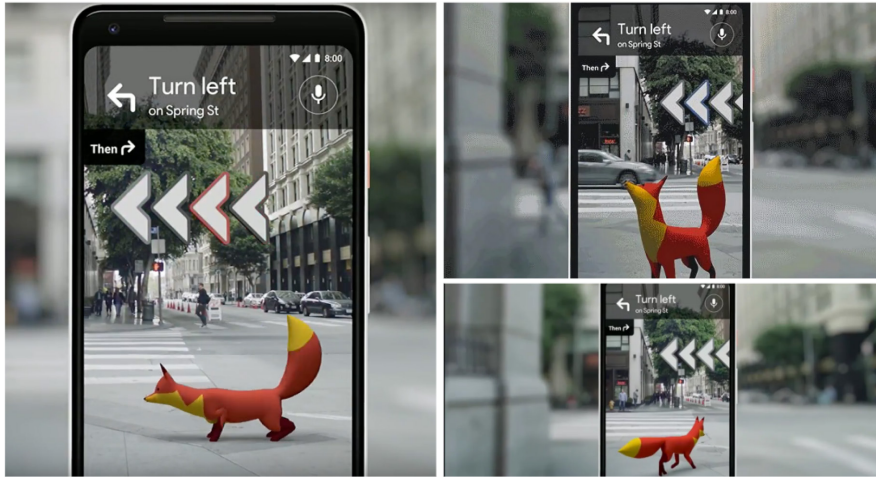


The above images are downloaded from the Internet.



The above images are downloaded from the internet.

APP BUDDY



Inspiration: Google Maps AR, fox concept

When you're unsure about your next move, you can hold your camera up to the street and your surroundings, and Google Maps will be able to provide you with a range of helpful specifics, like how many feet there are to your next destination, or whether to walk forward or back, or turn left or right.

Source: <https://mashable.com/article/google-maps-ar-fox>

BUDDY NAME: TIBBI

When said IITB faster, Tibbi comes out, and hence the name of the buddy. We also thought of names like Alpha, Gogo, Zero and finalised on Tibbi.

CHARACTER

When asked the students of IITB the character they would like, **robotic** character was a popular choice.

Buddy

A companion
in the AR app

Users would need
a motivation to
open the app and
access the
information

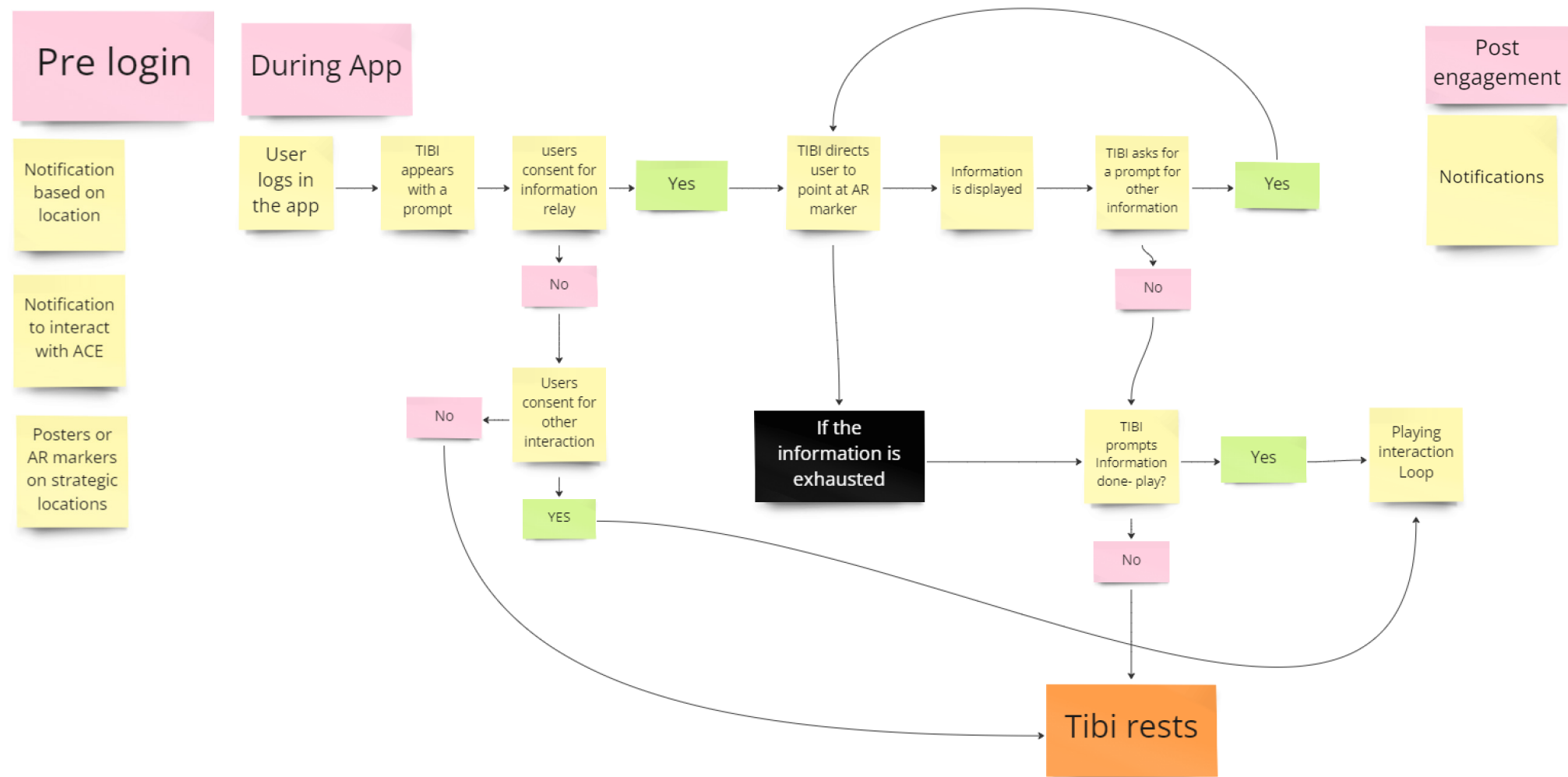
shows the way to
advertisements.
E.g. pablo (dog)
showing the way.

can play
with the
character

feed
it.

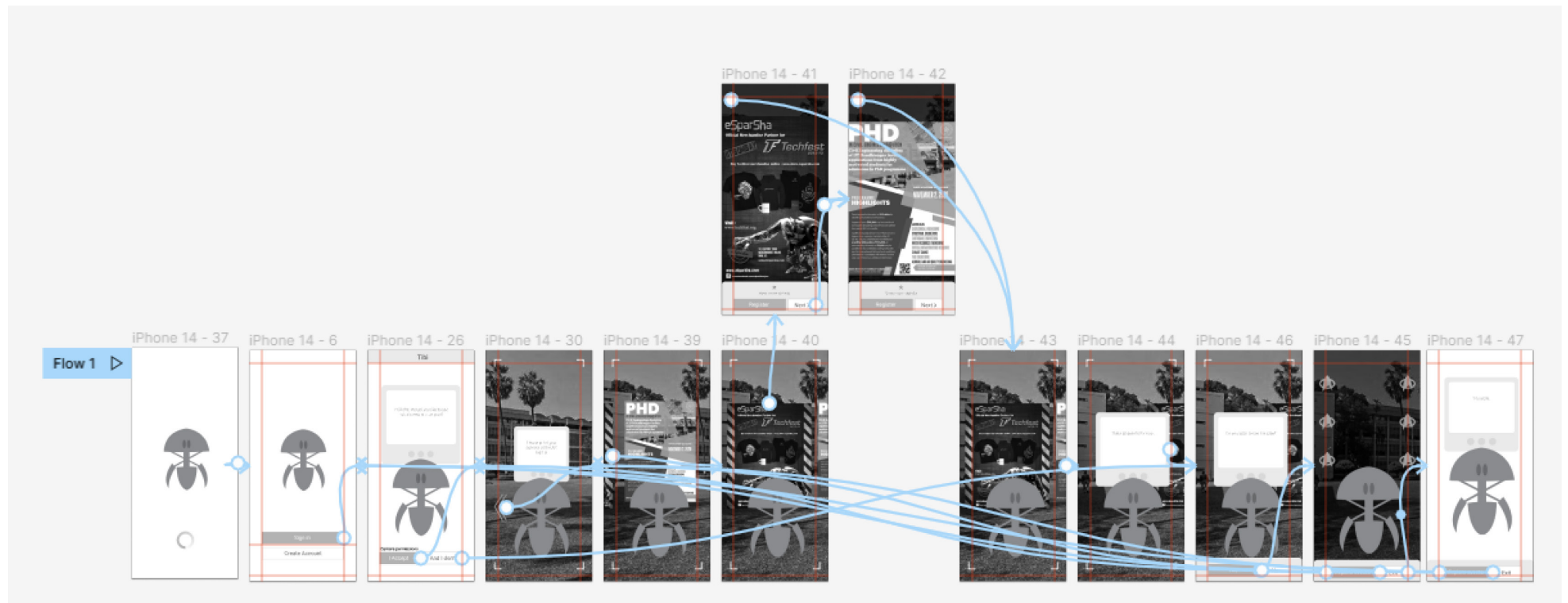
touch with
create a
response.

USER FLOW

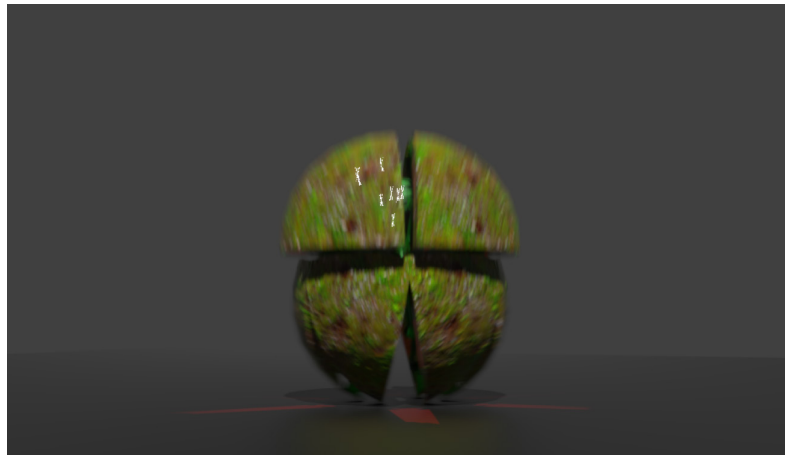


WIREFRAME

<https://www.figma.com/proto/sckGsy4XSB8BrDQfZOzOlB/Quillbot-Task?page-id=0%3A1&node-id=14%3A107&viewport=646%2C132%2C0.28&scaling=contain&starting-point-node-id=22%3A711>



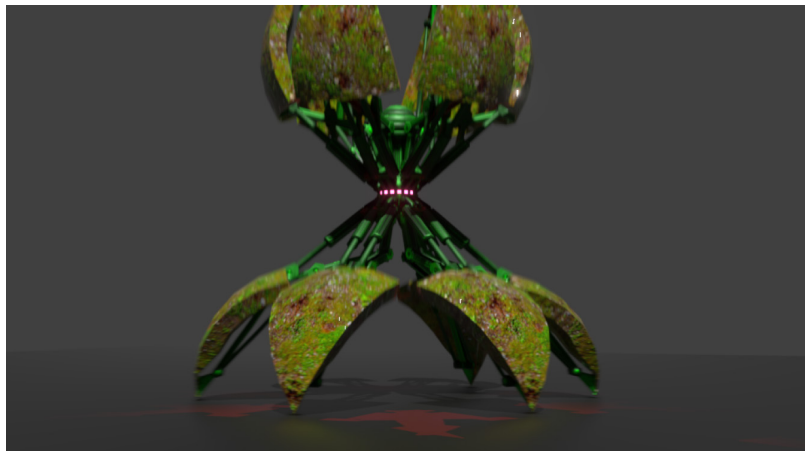
TIBBI CHARACTER EXPLORATIONS



Base Character credits

<https://free3d.com/3d-model/sphere-bot-basic-50922.html>

Texture credit



APP EXPLORATIONS

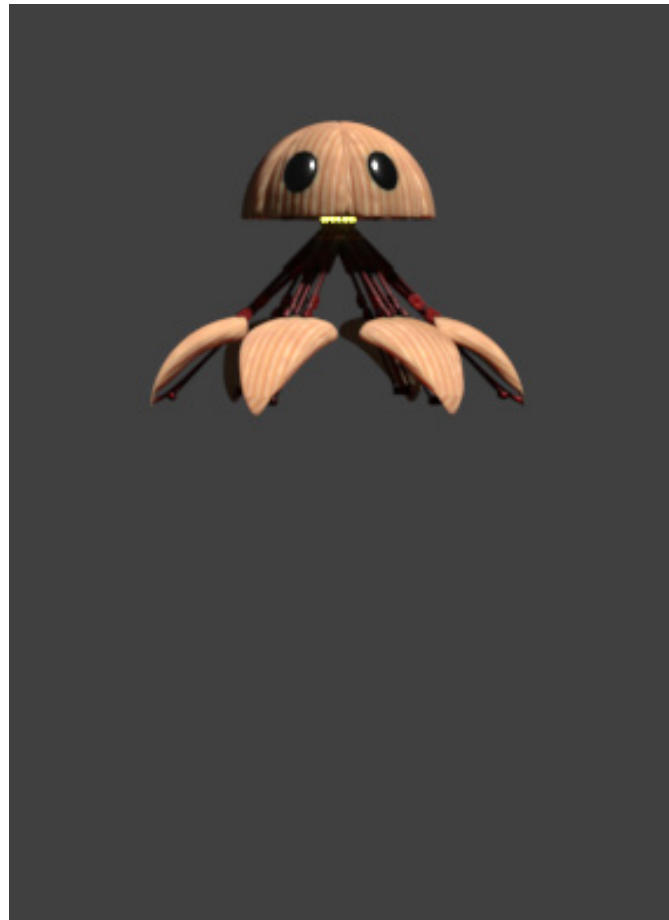
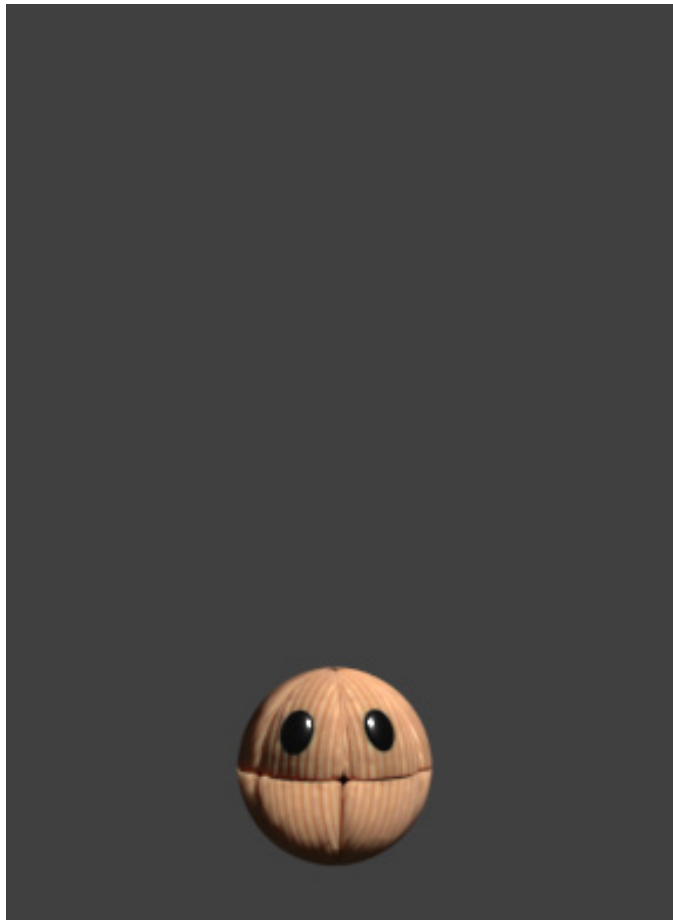
STAGE 1



TIBBI DEVELOPMENT

VERSION 2

We tried to make Tibbi look cuter and more approachable. We made the edges softer, added eyes, changed the texture and kept the color orange.



VISUAL STYLE

ELEMENTS

We had some keywords like: close to nature, save environment, playful, friendly, informative etc in mind. We looked at few references and existing apps Online. We liked the UI of Pokemon Go mobile game and Supercell game series. The following are some of the explorations.

Fonts used: Supercell-Magic

LOGO



Final Logo

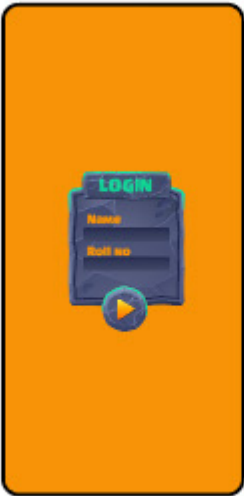
In the earlier logo, it was giving a feeling of eatery app due to its colour and icon. We then tried to make the icon character more friendly and went with a green gradient for the background.



VERSION 1



Loading



Login



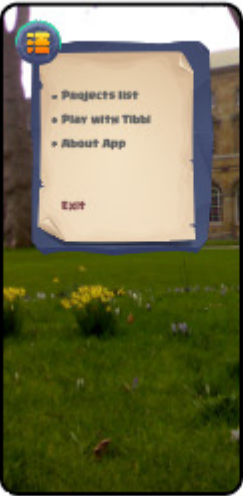
Intoduction
about App



Intoduction
prompt to scan



Camera opens
to view the
posters.



Dropdown
menu for other
options like:
project list, play
with tibbi, about
app and exit.



on clicking a
poster we can
see it full screen
and navigate.
There is also
an option to
explore further
links on the
poster like
student films.



Play with tibbi.
There will be
different options
of actions of
what we would
like Tibbi to do.

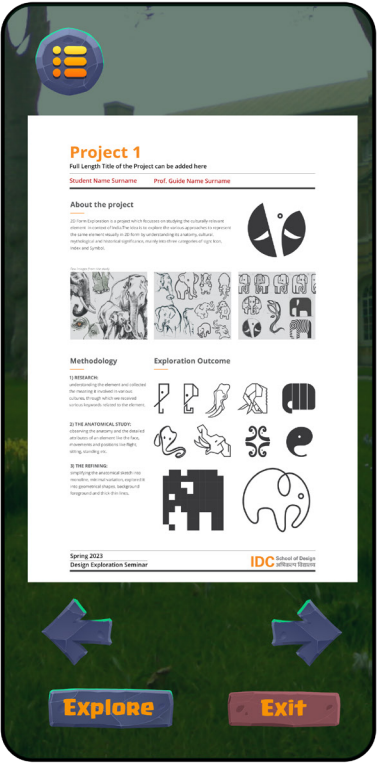
In the first version, we tried to use orange to give the App a friendly feel. Since this project aims to reduce the waste generated during events and promotions, we tried to have a theme close to nature and paper.

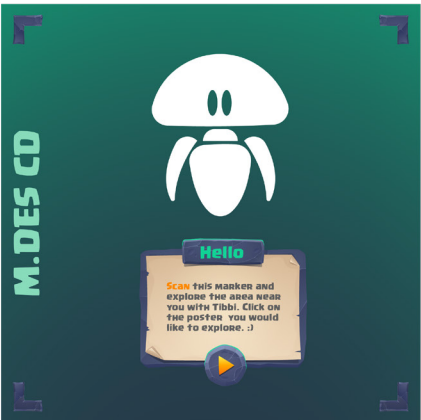
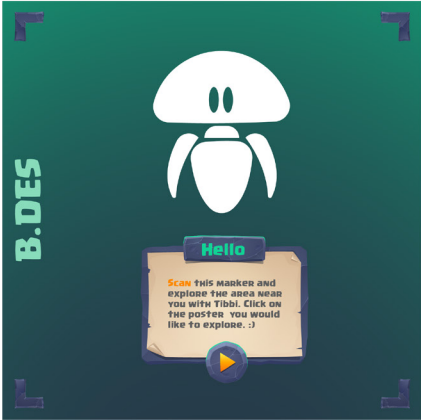
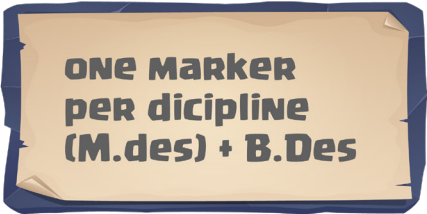


Version 2









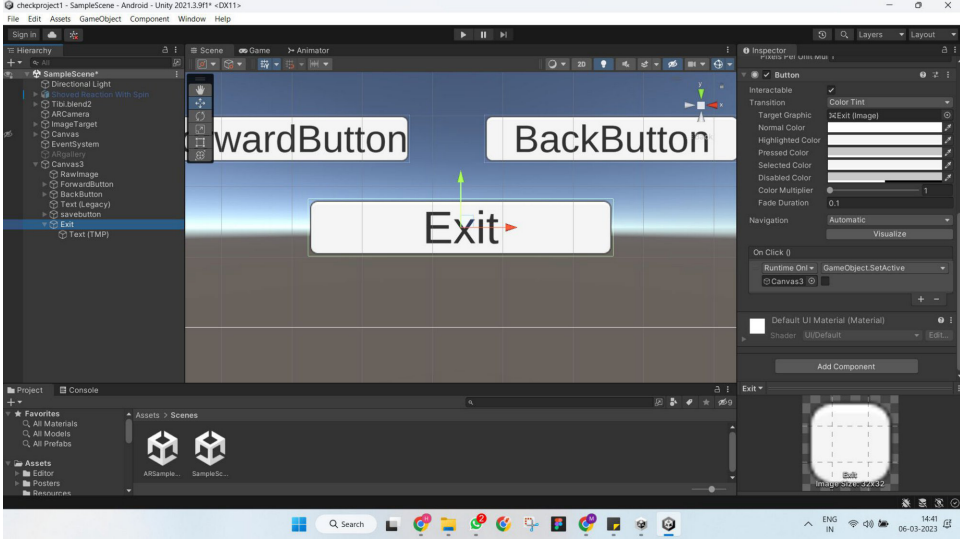
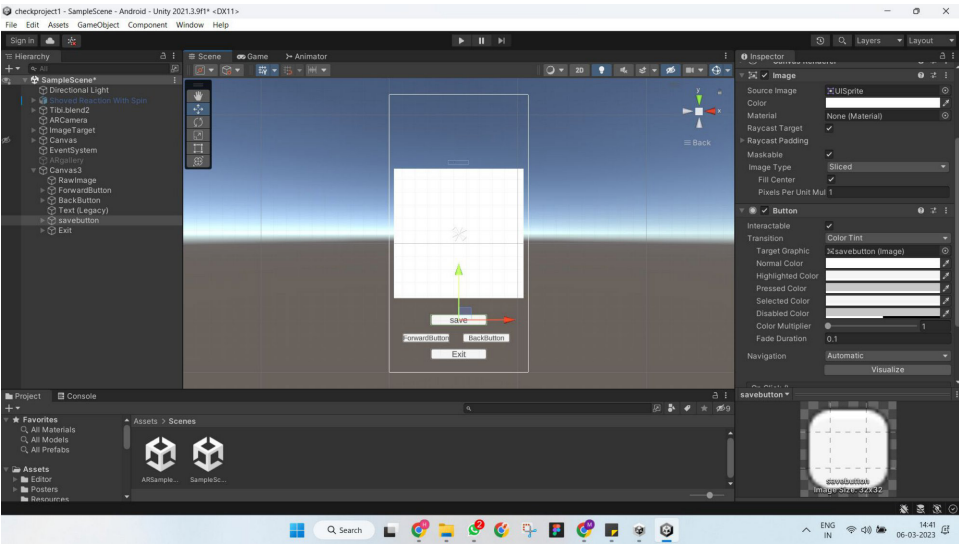
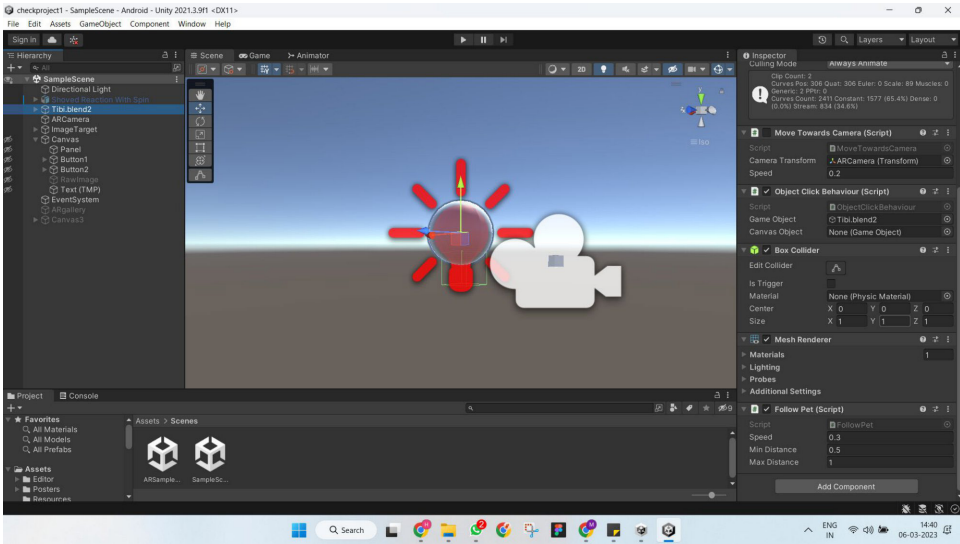
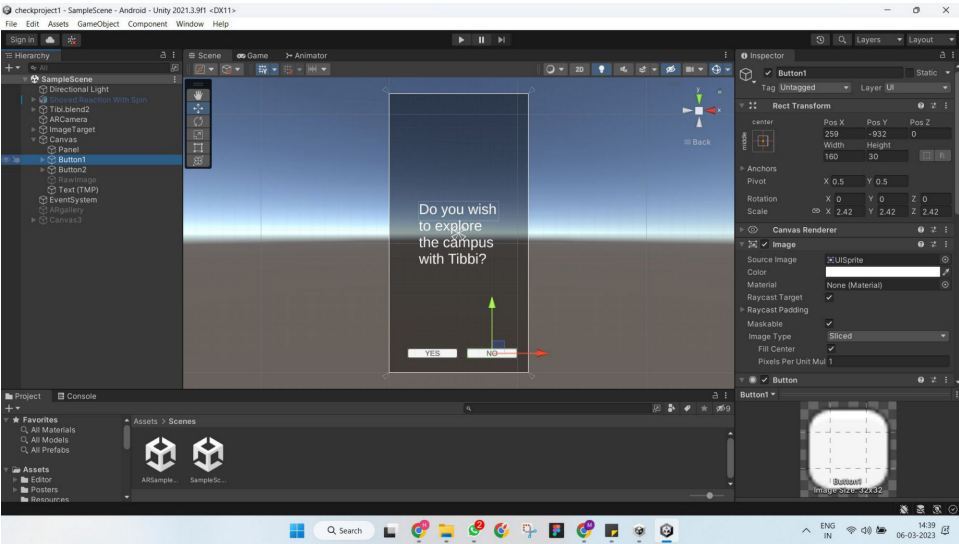


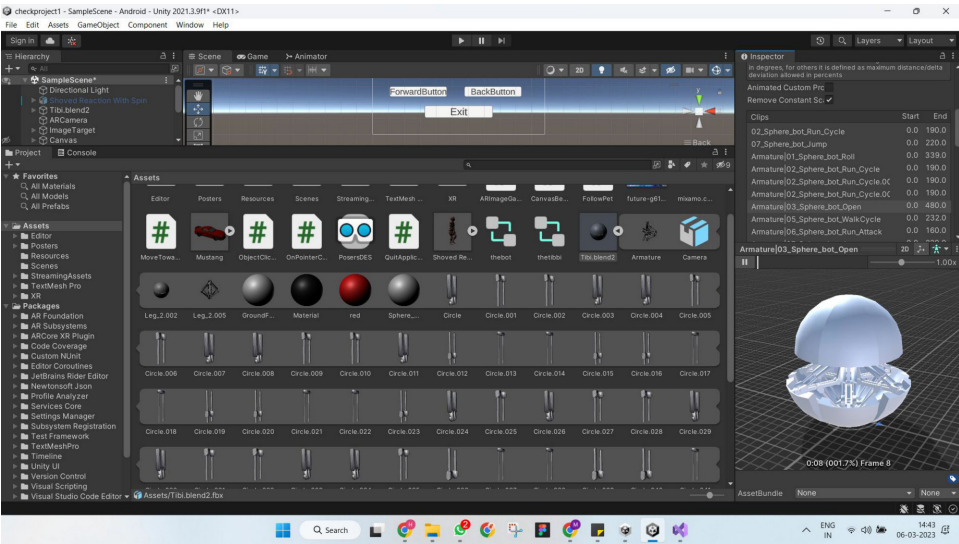
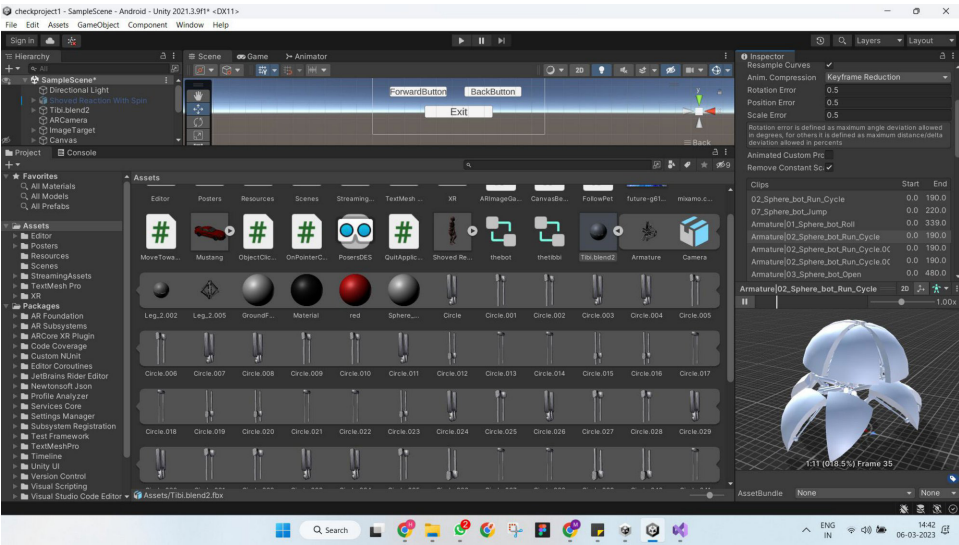






TECHNICAL IMPLEMENTATION SNAPS





ChatGPT

H write a C# script for unity to perform the following function. track the object if it is in the view of AR camera. If the object is not in the field of view, add a 2D marker image on the screen. Add the marker in the direction of the object. the marker appears only when the object is in the view and disappears when not in the view. The marker should be in the corner of the screen in the direction of the object. The script should take canvas as the input

Note: This script assumes that the tracked object is always visible in the scene and that the AR camera is the main camera. If the tracked object can move out of the scene or if there are multiple cameras in the scene, additional checks and modifications may be needed.

H rewrite the script considering the object is not visible in the screen and the marker appears only when the tracked object is in screen and disappears when it is out of the scene

G Sure, here's the updated C# script that tracks the object and shows a marker only when the object is not visible on the screen. The marker will appear in the direction of the object and disappear when it is back in view.

```
csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.UI;

public class ObjectTracker : MonoBehaviour
```

G Here's an updated version of the C# script that will add a marker image in the corner of the screen in the direction of the object when it is not in the field of view of the AR camera. The script takes a canvas object as input and adds the marker as a child of the canvas.

```
csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.UI;

public class ObjectTracker : MonoBehaviour
{
    [SerializeField] private ARCameraManager arCameraManager;
    [SerializeField] private GameObject trackedObject;
    [SerializeField] private Image markerPrefab;
    [SerializeField] private Canvas canvas;

    private Image markerImage;

    void Start()
    {
        // Create the marker image as a child of the canvas
        markerImage = Instantiate(markerPrefab);
```

H 'ARCameraManager' does not contain a definition for 'WorldToScreenPoint' and no accessible extension method 'WorldToScreenPoint' accepting a first argument of type 'ARCameraManager' could be found (are you missing a using directive or an assembly reference?)
This is the error message, update the script

G My apologies for the confusion. The 'ARCameraManager' does not have a 'WorldToScreenPoint' method. Instead, we can use the 'Camera' class to convert the world position to a screen position. Here's the updated script with the corrected method:

```
csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.UI;

public class ObjectTracker : MonoBehaviour
{
    [SerializeField] private ARCameraManager arCameraManager;
    [SerializeField] private GameObject trackedObject;
    [SerializeField] private Image markerPrefab;
```

H what do i add in ar camera manager

In the `arCameraManager` field of the script, you should assign a reference to the `ARCameraManager` component that is responsible for tracking the AR camera in the scene. This `ARCameraManager` component is typically added automatically when you add the AR Session component to the scene.

To add the `AR Session` and `AR Session Origin` components to your scene, you can go to the GameObject menu, select "XR", and then select "AR Session". This will create an `AR Session` GameObject in the scene, along with a child `AR Session Origin` GameObject that has the AR camera and other tracking components attached.

Once you have added the `AR Session` and `AR Session Origin` components to your scene, you can assign the reference to the `ARCameraManager` component in the `arCameraManager` field of the script. To do this, you can drag the `AR Session Origin` GameObject from the Hierarchy window to the `arCameraManager` field in the Inspector window of the `ObjectTracker` script.

H MoveMarkerTowardsPosition IS NOT IN THE ABOVE SCRIPT, PLEASE TELL ME HOW DO I CHANGE MARKER SPEED IN THE LATEST CODE YOU HAVE SENT

I apologize for the confusion again. To adjust the speed of the marker in the updated code I provided, you can modify the `markerSpeed` parameter in the `Update` method.

Here's the updated code with the `markerSpeed` parameter:

```
csharp
Copy code

public class ARObjectTracker : MonoBehaviour
{
    public ARTrackedObject trackedObject;
    public GameObject markerPrefab;
    public Canvas canvas;
```

H markerSpeed is not found in the above script

I apologize for the confusion. The `markerSpeed` variable is not present in the script I provided earlier. To adjust the speed of the marker, you can modify the `lerpSpeed` parameter in the `UpdateMarkerPosition` method.

H lerpSpeed is also not available in the above script

I apologize again for the confusion. You are correct that the `lerpSpeed` parameter is not present in the script I provided.

To adjust the speed of the marker image, you can modify the `MoveMarkerTowardsPosition` method. Specifically, you can adjust the value of `lerpTime`, which controls how long the marker image takes to move from its current position to the target position.

H ARCameraManager' does not contain a definition for 'WorldToScreenPoint' and no accessible extension method 'WorldToScreenPoint' accepting a first argument of type 'ARCameraManager' could be found
PLEASE FIX TIS ERROR AND UPDATE THE SCRIPT

I apologize for the confusion. It looks like I made a mistake in the previous code snippet. The `WorldToScreenPoint` method is actually a member of the `Camera` class, not the `ARCameraManager` class.

Here's the updated code with the corrected method call:

I apologize for the confusion. The `Vector2` class does not have a `z` component, only `x` and `y`. I mistakenly used the `z` component in the previous code snippet. Here's the updated code with the correction:

```
csharp
Copy code

public class ARObjectTracker : MonoBehaviour
```



How do I add texture to an object in unity

To add texture to an object in Unity, you can follow these steps:

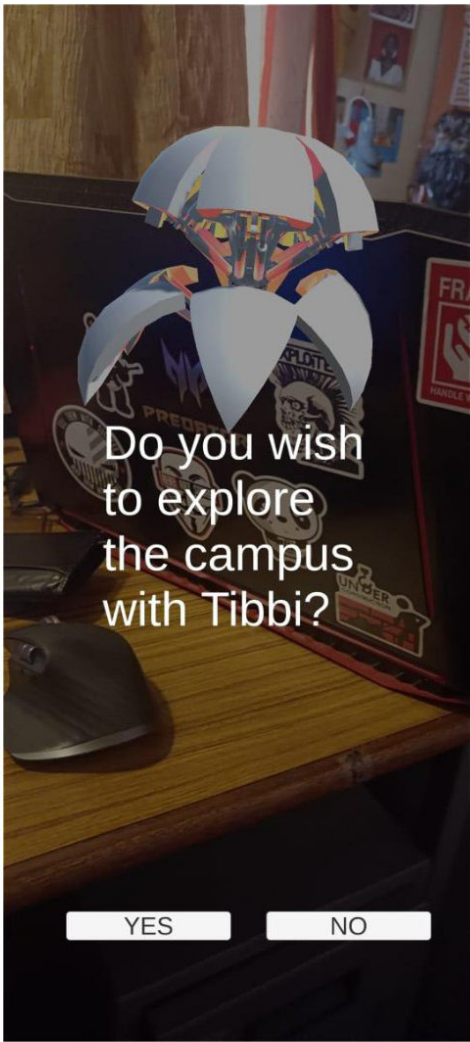
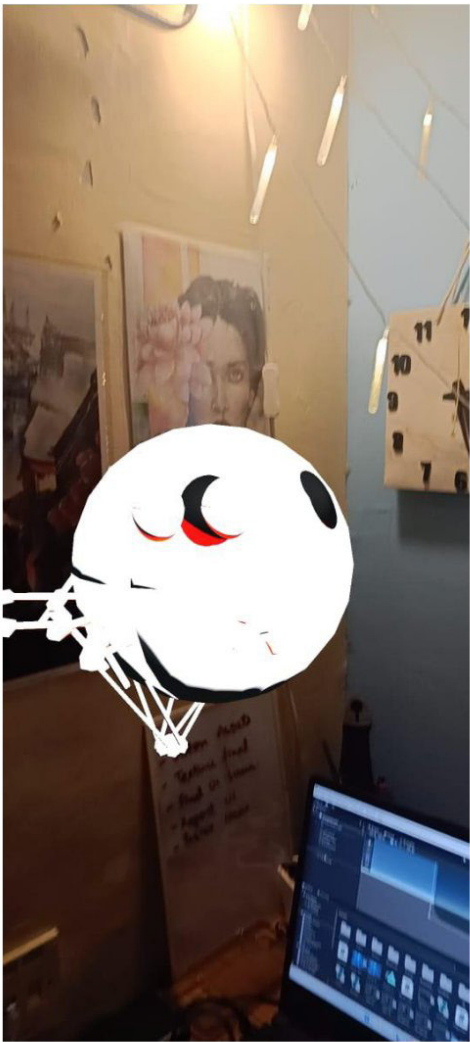
1. Import the texture into Unity by dragging and dropping it into the project window.
2. Select the object you want to apply the texture to in the Scene view or Hierarchy window.
3. In the Inspector window, click on the Add Component button and select Renderer from the menu.
4. Click on the small circle next to the Albedo field under the Materials section in the Renderer component.
5. Select the texture you imported from the project window.
6. You can adjust the tiling and offset of the texture in the Material Inspector to control how the texture is applied to the object.
7. You can also adjust other settings in the Material Inspector, such as the transparency and smoothness.
8. Preview the texture by clicking on the Play button in the Unity editor, or by pressing the Preview button in the Scene view.

That's it! With these steps, you should be able to apply textures to any object in Unity.

```
File Edit View Git Project Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Solution1
FollowPet.cs x QuitApplications OnPointerClickQuits ObjectClickBehaviours MoveTowardsCameras
Miscellaneous Files FollowPet
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class FollowPet : MonoBehaviour
6 {
7     public float speed; // Speed at which the pet will move towards the camera
8     public float minDistance; // Minimum distance the pet should be from the camera
9     public float maxDistance; // Maximum distance the pet should be from the camera
10     private Transform cameraTransform; // Reference to the camera's transform
11
12     void Start()
13     {
14         cameraTransform = Camera.main.transform; // Get reference to main camera's transform
15     }
16
17     void Update()
18     {
19         // Calculate direction vector towards camera
20         Vector3 direction = cameraTransform.position - transform.position;
21         direction.y = 0f; // Keep the pet at the same height as the camera
22
23         // Rotate the pet towards the camera
24         transform.rotation = Quaternion.Slerp(transform.rotation, Quaternion.LookRotation(direction), 0.1f);
25
26         // Calculate the current distance between the pet and the camera
27         float distance = Vector3.Distance(transform.position, cameraTransform.position);
28
29         // Move the pet towards the camera with a constant speed
30         if (distance < minDistance)
31         {
32             transform.position -= transform.forward * speed * Time.deltaTime;
33         }
34         else if (distance > maxDistance)
35         {
36             transform.position += transform.forward * speed * Time.deltaTime;
37         }
38     }
39 }
```

```
File Edit View Git Project Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Solution1
QuitApplications OnPointerClickQuits ObjectClickBehaviours MoveTowardsCameras
Miscellaneous Files QuitApplication
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class QuitApplication : MonoBehaviour
6 {
7     //public Button quitButton;
8
9     void Start()
10     {
11         // Attach a listener to the button's click event
12         // quitButton.onClick.AddListener(Quit);
13         // Get reference to the button component on this gameObject
14         // Button quitButton = GetComponent<Button>();
15
16         // Add a listener for the button click event
17         //quitButton.onClick.AddListener(QuitGame);
18     }
19
20     void QuitGame()
21     {
22         // Quit the application
23         Application.Quit();
24     }
25
26 }
```

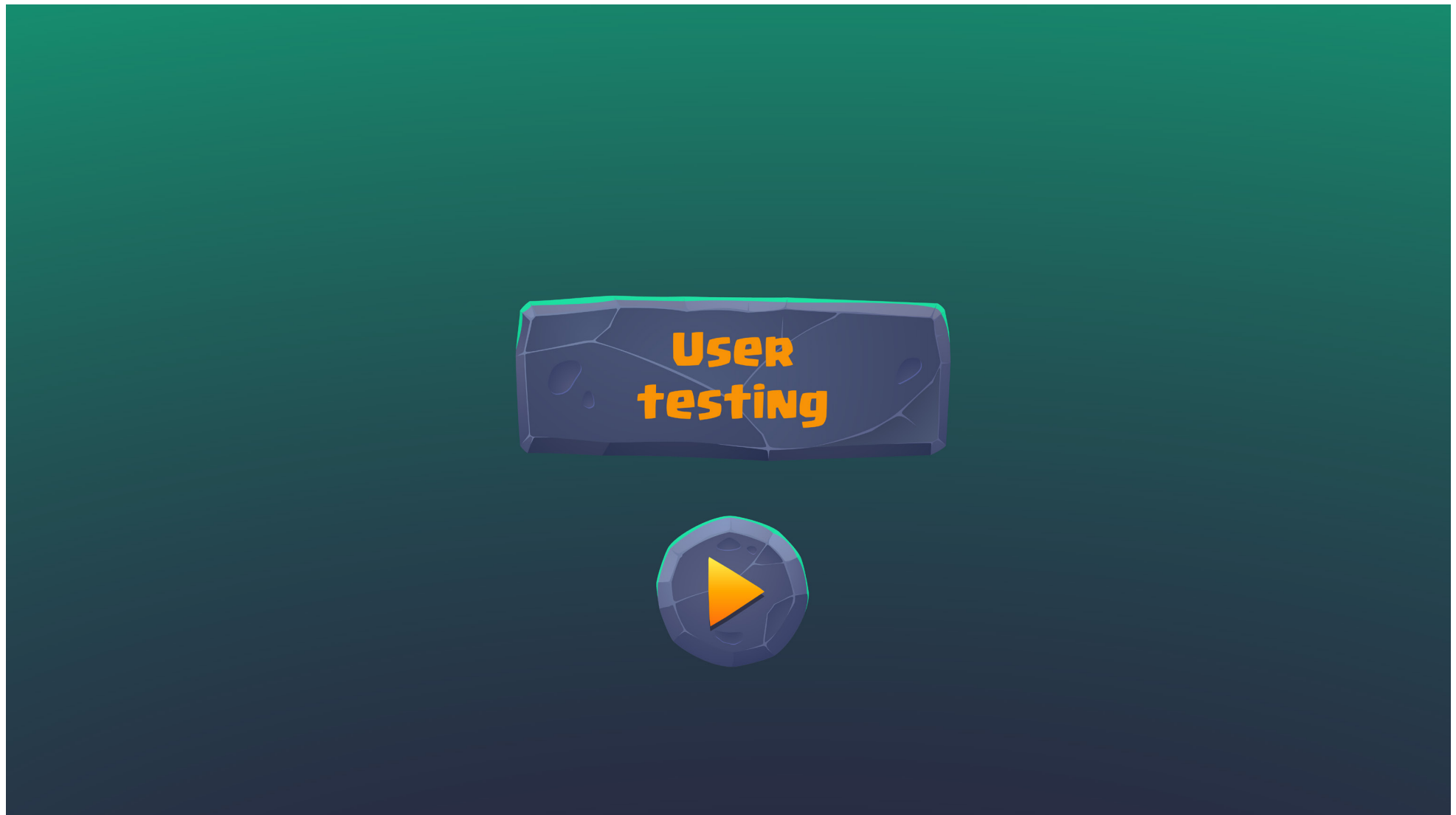
```
File Edit View Git Project Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Solution1
ObjectClickBehaviours MoveTowardsCameras
Miscellaneous Files ObjectClickBehaviour
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class ObjectClickBehaviour : MonoBehaviour
6 {
7     //public AudioSource source;
8     public GameObject gameObject;
9     public GameObject canvasObject;
10     string objectName;
11
12     // Start is called before the first frame update
13     void Start()
14     {
15         //source = GetComponent<AudioSource>();
16     }
17
18     // Update is called once per frame
19     void Update()
20     {
21         if (Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began)
22         {
23             Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position);
24             RaycastHit hit;
25             if (Physics.Raycast(ray, out hit))
26             {
27                 objectName = hit.transform.name;
28                 switch (objectName)
29                 {
30                     case "Cube":
31                         canvasObject.SetActive(true);
32                         break;
33                     case "T1bl.blend2":
34                         // gameObject.SetActive(false);
35                         break;
36                 }
37             }
38         }
39     }
40 }
```

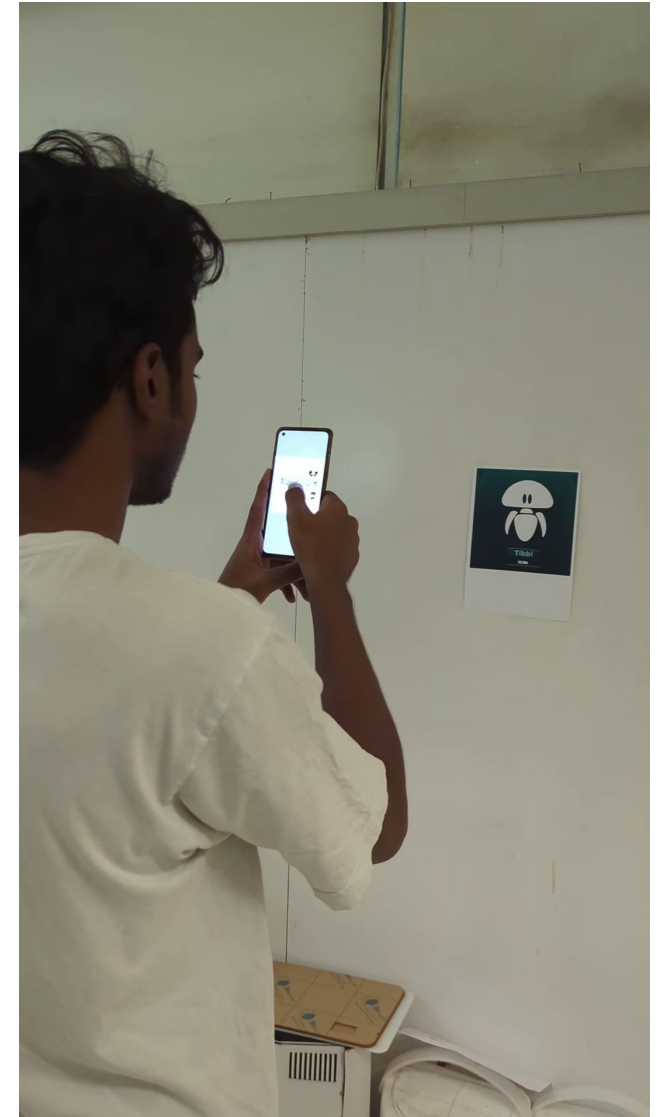
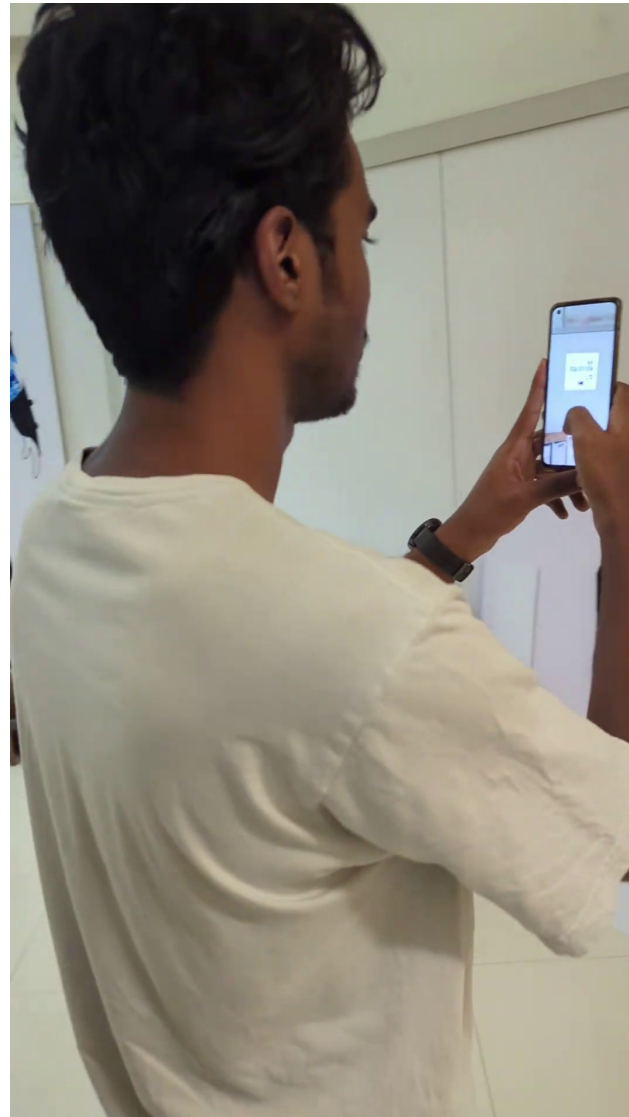
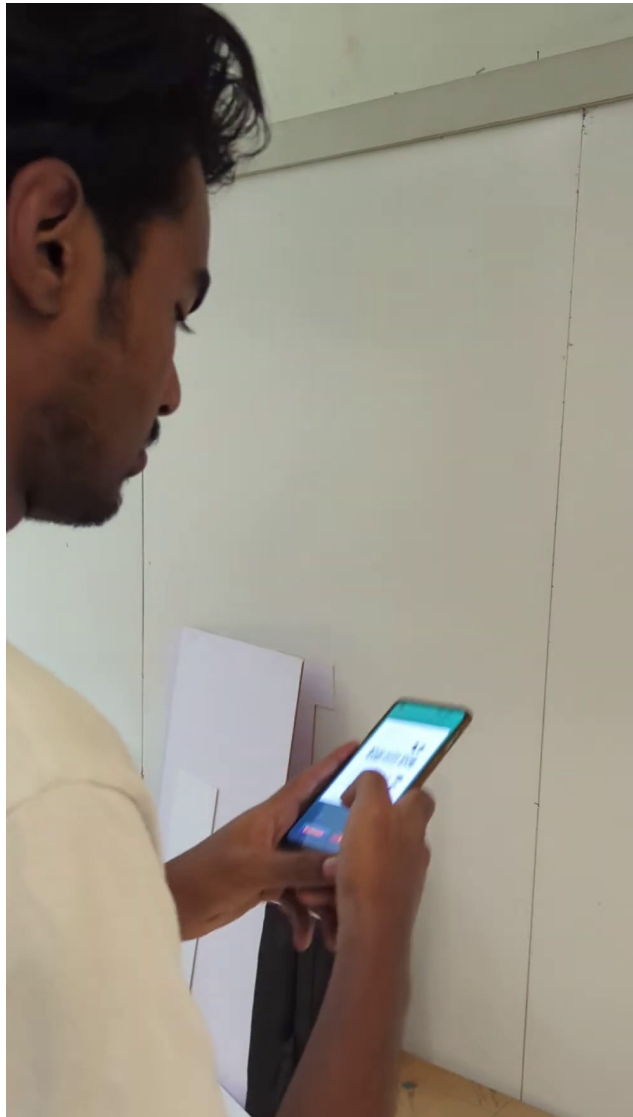
Do you wish
to explore
the campus
with Tibbi?

YES

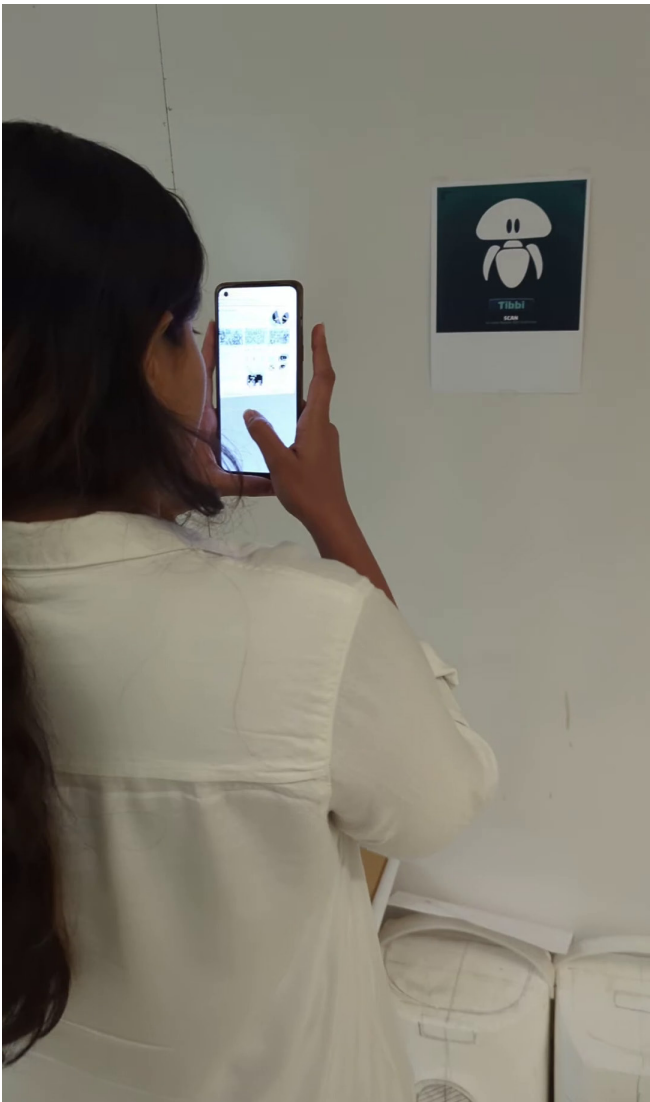
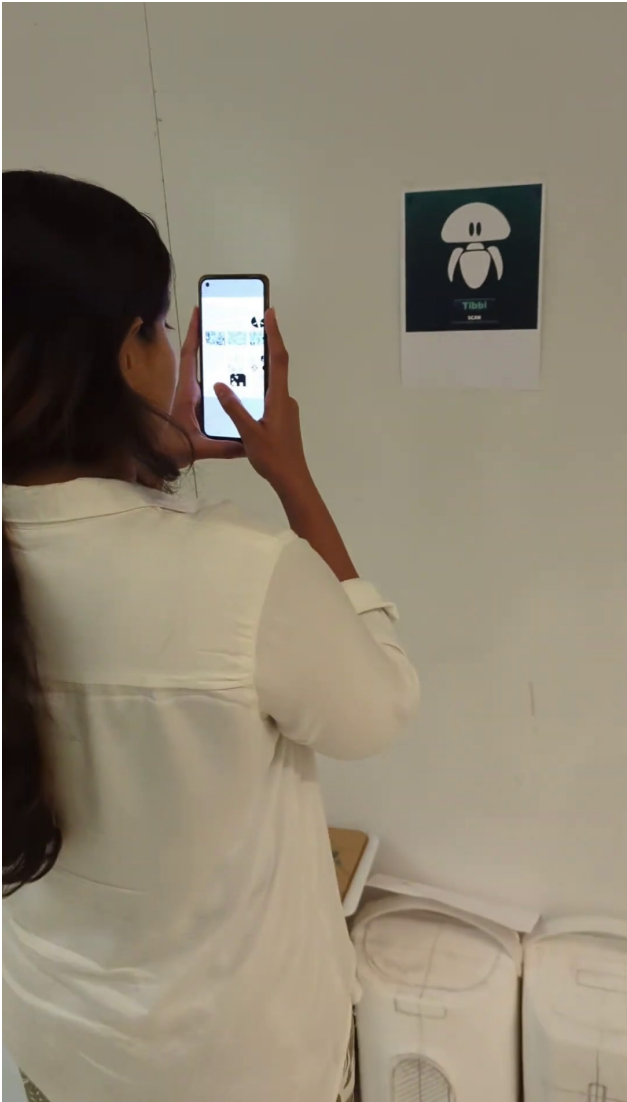
NO

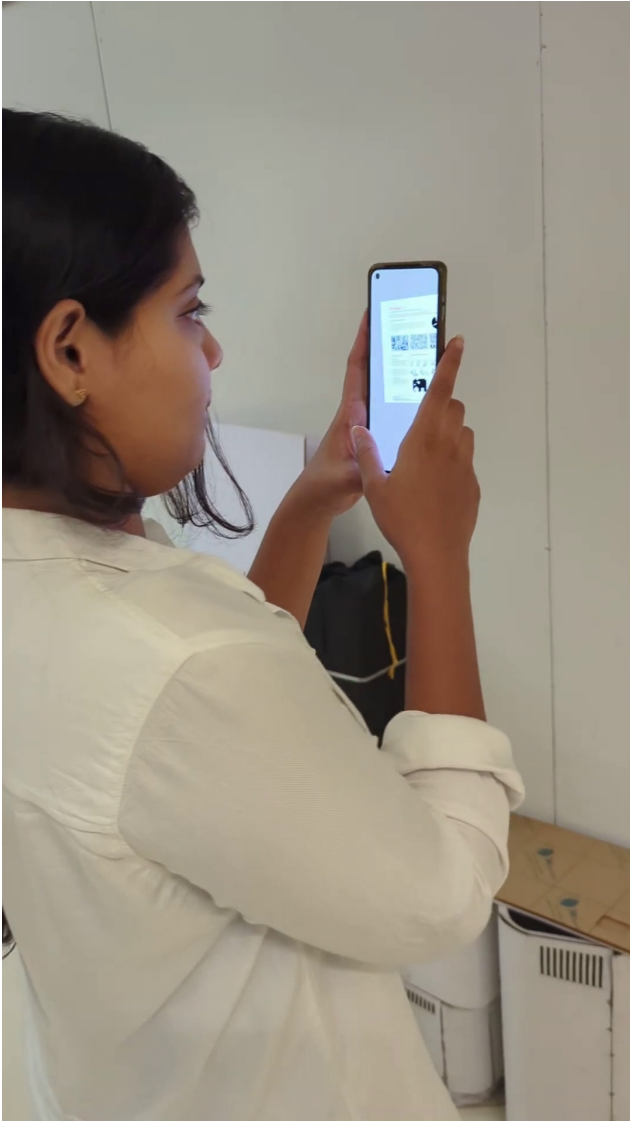
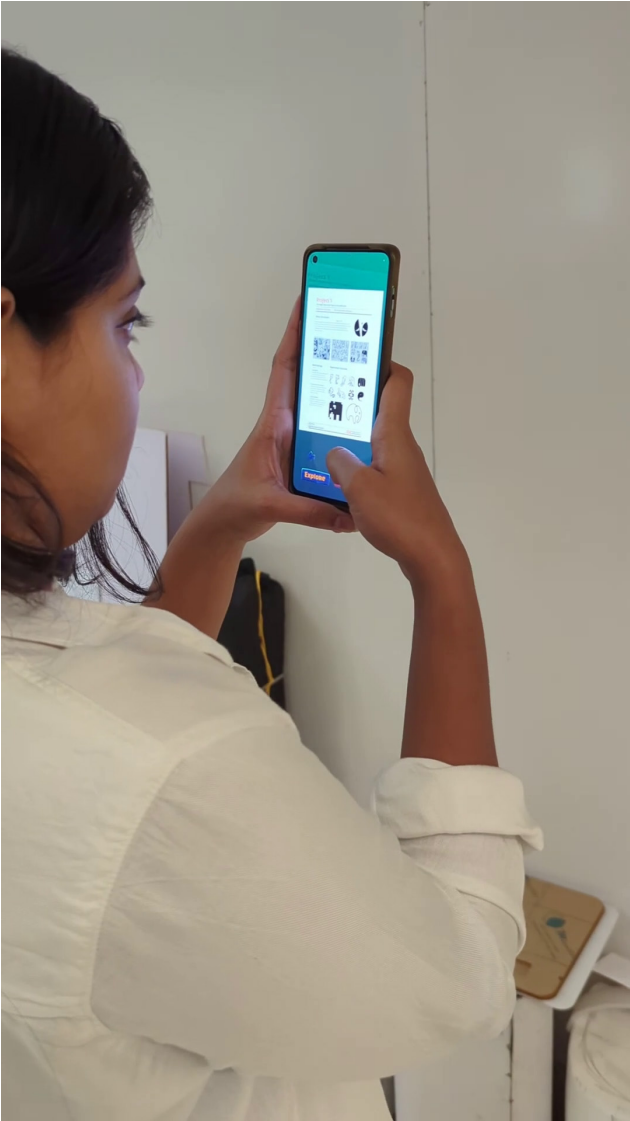


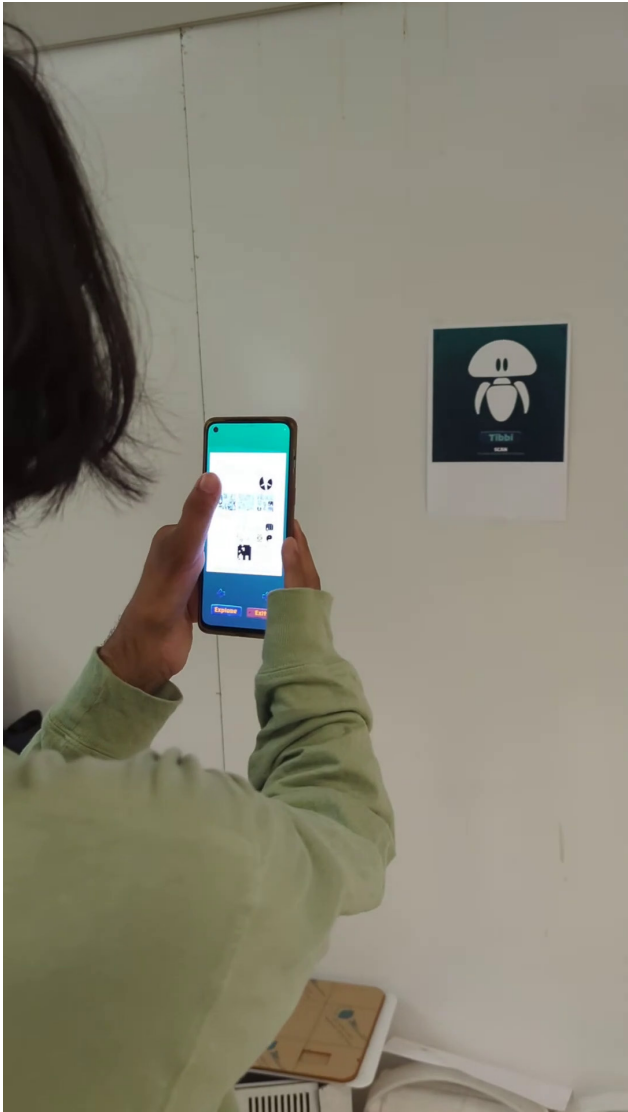
USER TESTING



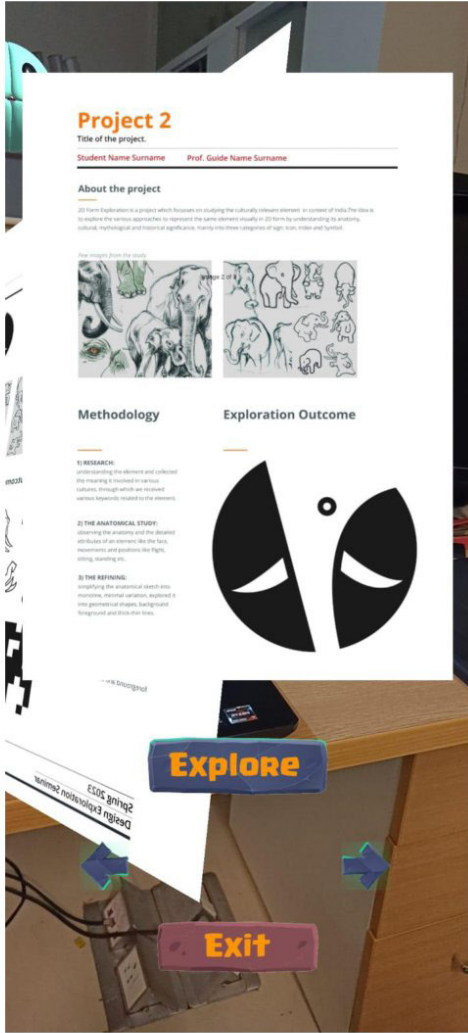
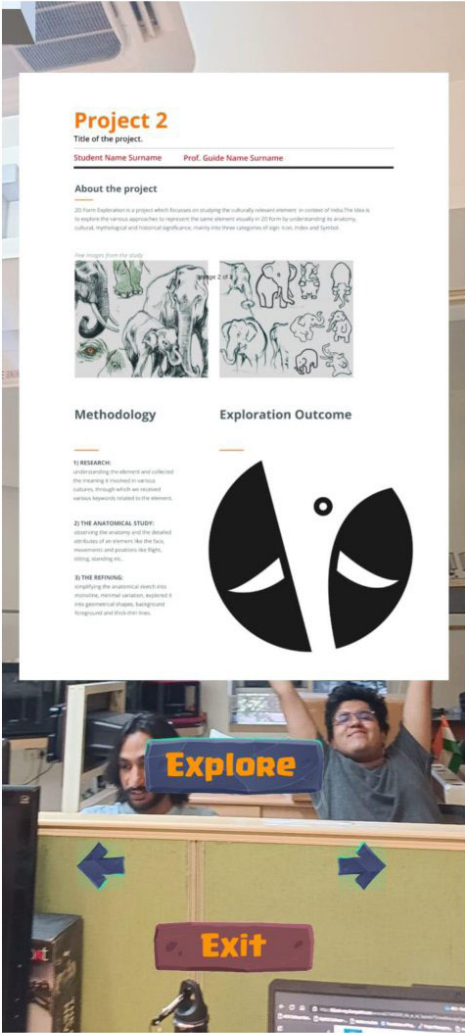












FUTURE SCOPE

