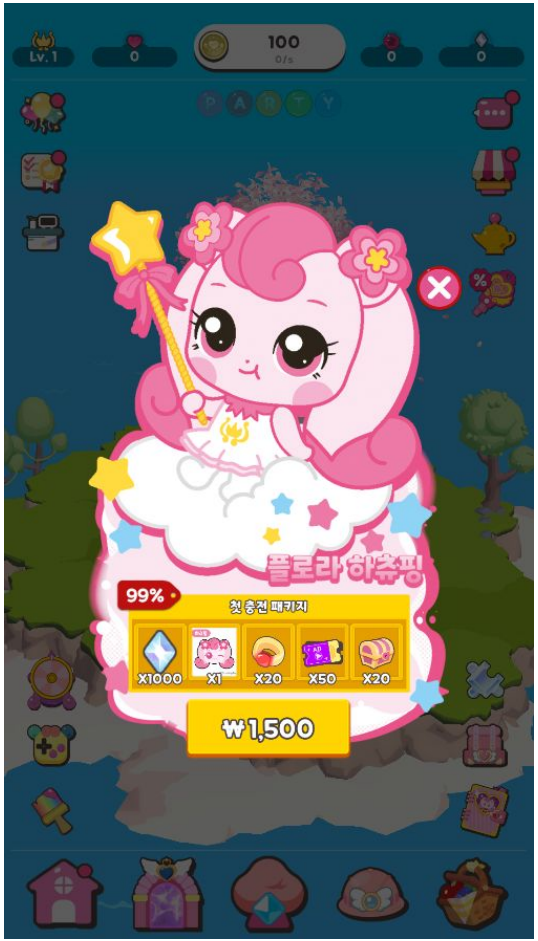
이모션티니핑

플레이스토어 :
https://play.google.com/store/apps/details?id=com.entereal.emotionteenieping
ios :
https://apps.apple.com/kr/app/%EC%9D%B4%EB%AA%A8%EC%85%98%ED%8B%B0%EB%8B%88%ED%95%91/id6450882377

사용한 스킬

- FireBase Auth
- FireBase Messaging
- FireBase Realtime Database(현재는 사용안함)
- FireBase RemoteConfig
- 뒤끝
- GoogleSignin
- GPGS(현재는 사용안함)
- 인앱결제(안드로이드,ios)
- DotTween
- WebCam

인앱 결제 (뒤끝 영수증 검증)




```
public PurchaseProcessingResult ProcessPurchase(PurchaseEventArgs args)
{
    GameManager.Instance.UiManager.ActiveIndicatePage(false);
    Debug.LogError("ProcessPurchase");
UNITY_ANDROID
    BackendReturnObject validation = Backend.Receipt.IsValidateGooglePurchase(args.purchasedProduct.receipt, "receiptDescription", false);
if UNITY_IOS
    BackendReturnObject validation = Backend.Receipt.IsValidateApplePurchase(args.purchasedProduct.receipt, "receiptDescription");
dif
    if (validation.IsSuccess())
    {
        if (String.Equals(args.purchasedProduct.definition.id, PurchaseProductId, StringComparison.Ordinal))
        {
            Debug.LogError(string.Format("ProcessPurchase: PASS. Product: '{0}'", args.purchasedProduct.definition.id));
            GameManager.Instance.DataManager.BuyCompleteByPurchaser(PurchaseProductId);
            GameManager.Instance.DataManager.SendSaveData(TransactionType.Update, "ProcessPurchase");
        }
    }
    else
    {
        Debug.LogError(string.Format("ProcessPurchase: FAIL. Unrecognized product: '{0}'", args.purchasedProduct.definition.id));
    }

    return PurchaseProcessingResult.Complete;
}
```

```
public void TryPurchaseProduct(string id)
{
    GameManager.Instance.UiManager.ActiveIndicatePage(true);
    if (!IsInitialized())
    {
        Initialize();
    }

    PurchaseProductId = id;
    BuyProductID(PurchaseProductId);
}
```


```
.DORotate(targetRotation, spinDuration, RotateMode.FastBeyond360)
.SetEase(Ease.InOutQuart)
.OnUpdate(() =>
{
    float diff = Mathf.Abs(prevAngle - currentAngle);
    if (diff >= halfPieceAngle)
    {
        if (isIndicatorOnTheLine)
        {
            audioSource.PlayOneShot(audioSource.clip);
        }
        prevAngle = currentAngle;
        isIndicatorOnTheLine = !isIndicatorOnTheLine;
    }
    currentAngle = wheelCircle.eulerAngles.z;
})
.OnComplete(() =>
{
    _isSpinning = false;
    if (onSpinEndEvent != null)
        onSpinEndEvent.Invoke(piece);
    audioSource.PlayOneShot(endAudioClip);
    onSpinStartEvent = null;
    onSpinEndEvent = null;
});
```

```
        m_timerCurrent += Time.deltaTime * Speed;
        transform.position = new Vector3(
            CubicBezierCurve(m_points[0].x, m_points[1].x, m_points[2].x, m_points[3].x),
            CubicBezierCurve(m_points[0].y, m_points[1].y, m_points[2].y, m_points[3].y),
            CubicBezierCurve(m_points[0].z, m_points[1].z, m_points[2].z, m_points[3].z)
        );
    }
    /// <summary>
    /// 어떤 아이템인지 체크
    /// </summary>
    /// <param name="startPos">시작위치</param>
    /// <param name="target">목표</param>
    /// <param name="type">아이템타입</param>
    /// <param name="_action">도착할때 콜백함수</param>
    참조 1개
    public void Initialize(Vector3 startPos, Transform target, GoldType type, UnityAction _action = null)...
    /// <summary>
    /// 베지어 값
    /// </summary>
    /// <param name="a"></param>
    /// <param name="b"></param>
    /// <param name="c"></param>
    /// <param name="d"></param>
    /// <returns></returns>
    참조 3개
    public float CubicBezierCurve(float a, float b, float c, float d)...
```

룰렛기능
BezierCurve

WebCam&티니핑 움직이기

```
    WebCamDevice[] devices = WebCamTexture.devices;

    int selectedCameraIndex = -1;
    for (int i = 0; i < devices.Length; i++)
    {
        // 후면 카메라인지 체크
        if (devices[i].isFrontFacing == false)
        {
            // 해당 카메라 선택
            selectedCameraIndex = i;
            break;
        }
    }

    if (selectedCameraIndex >= 0)
    {
        // 선택된 카메라에 대한 새로운 WebCamTexture를 생성
        textureWebCam = new WebCamTexture(devices[selectedCameraIndex].name);

        // 원하는 FPS를 설정
        if (textureWebCam != null)
        {
            textureWebCam.requestedFPS = 60;
        }
    }

    // objectTarget으로 카메라가 표시되도록 설정
    if (textureWebCam != null)
    {
        //image.material.mainTexture = textureWebCam;
        renderer.material.mainTexture = textureWebCam;
        renderer.transform.localScale = initialScale;
    }
}
/// <summary>
/// 삭제됐을때 webcam삭제
/// </summary>
Unity 메시지 | 참조 0개
void OnDestroy()...
/// <summary>
/// 현재 캔버스의 크기에맞춰 해상도조절
/// </summary>
참조 1개
void BGScale()...
/// <summary>
/// WebCamStart
/// </summary>
참조 1개
public void OnPlayButtonClick()...
참조 1개
public void OnStopButtonClick()...
```

```
    if (Input.GetMouseButton(0))
    {
        Touch touch = Input.GetTouch(0);

        if (!EventSystem.current.IsPointerOverGameObject(touch.fingerId))
        {
            Ray ray = Camera.main.ScreenPointToRay(touch.position);
            RaycastHit hit;

            int layerMask = LayerMask.GetMask("Deco");
            if (Physics.Raycast(ray, out hit, Mathf.Infinity, layerMask))
            {
                Vector3 worldPosition = hit.point;
                Vector3 newPosition = new Vector3(worldPosition.x, worldPosition.y, originz);
                CatchTeenieping.transform.position = newPosition;
            }
            else
            {
                Debug.Log("UI에 터치되었습니다.");
            }
        }

    }
}
if (Input.touchCount == 2)
{
    Touch touch1 = Input.GetTouch(0);
    Touch touch2 = Input.GetTouch(1);

    float distance = Vector2.Distance(touch1.position, touch2.position);
    float prevDistance = Vector2.Distance(touch1.position - touch1.deltaPosition, touch2.position - touch2.deltaPosition);
    float deltaDistance = distance - prevDistance;

    float zoomSpeed = 0.005f;

    float scaleFactor = 1.0f + (deltaDistance * zoomSpeed);
    Vector3 currentScale = CatchTeenieping.transform.localScale;

    Vector3 newScale = currentScale * scaleFactor;
    newScale.x = Mathf.Clamp(newScale.x, minScale, maxScale);
    newScale.y = Mathf.Clamp(newScale.y, minScale, maxScale);
    newScale.z = Mathf.Clamp(newScale.z, minScale, maxScale);

    CatchTeenieping.transform.localScale = newScale;
}
yield return null;
```

```
FirebaseApp.CheckAndFixDependenciesAsync().ContinueWith(task =>
{
    if (task.IsCompleted)
    {
        if (task.Result == DependencyStatus.Available)
        {
            auth = FirebaseAuth.DefaultInstance;
            firebaseConnet = true;
            FirebaseRemoteConfig.DefaultInstance.SetDefaultsAsync(serverKeyValue).ContinueWithOnMainThread(task =>
            {
                FetchData();
            });
        }
        else
            Debug.Log("Could not resolve all Firebase dependencies: " + task.Result.ToString());
    }
    else
    {
        Debug.Log("Dependency check was not completed. Error : " + task.Exception.Message);
    }
});
void FetchData()
{
    FirebaseRemoteConfig.DefaultInstance.FetchAsync(TimeSpan.Zero).ContinueWithOnMainThread(fetchTask =>
    {
        if (fetchTask.IsCompleted)
        {
            FirebaseRemoteConfig.DefaultInstance.ActivateAsync().ContinueWithOnMainThread(activateTask =>
            {
                string serverType = FirebaseRemoteConfig.DefaultInstance.GetValue("ServerType").StringValue;
                Debug.Log("ServerType: " + serverType);
                if (serverType == "0" || serverType == "1")
                    DataManager.I.ServerType = (DataManager.Server)int.Parse(serverType);
                isFecth = true;
            });
        }
        else
        {
            Debug.LogError("Failed to fetch data");
        }
    });
}
```

FireBase RemoteConfig

GoogleSignIn & 파이어베이스 & 뒤끝연동

```
#region 로그인 & 로그아웃
/// <summary>
/// 로그인 후 데이터 가져오기
/// </summary>
참조 1개
public void GetData()...
/// <summary>
/// 게스트로그인
/// </summary>
참조 1개
private void SignInWithGuestLogin()...
/// <summary>
/// 구글로그인후 파이어베이스 연동
/// </summary>

참조 2개
private void SignInWithGoogleOnFirebase()...
/// <summary>
/// GoogleSignIn 로그인
/// </summary>
/// <param name="task"></param>
참조 0개
internal void OnAuthenticationFinished(Task<GoogleSignInUser> task)...
/// <summary>
/// 뒤끝 구글 로그인
/// </summary>
/// <param name="isSuccess"></param>
/// <param name="errorMessage"></param>
/// <param name="token"></param>
참조 2개
private void OnGoogleLoginCallback(bool isSuccess, string errorMessage, string token)...
참조 0개
public void BackendLogin()...
/// <summary>
/// 뒤끝 게스트로그인
/// </summary>
참조 1개
private void GhestLogin()...
/// <summary>
/// 로그아웃
/// </summary>
참조 1개
private void Logout()...
/// <summary>
/// 로그인 정보 상태
/// </summary>
/// <param name="_Code"></param>
/// <param name="_message"></param>

참조 3개
public void CodeCheck(string _Code, string _message)...
#endregion
```

```
void GetLiveData()
{
    IsGetData = false;
    var bro = Backend.Chart.GetChartListV2();

    if (!bro.IsSuccess())
    {
        Debug.LogError("에러가 발생했습니다 : " + bro.ToString());
        return;
    }



    JsonData json = bro.FlattenRows();

    for (int i = 0; i < json.Count; i++)
    {
        ChartCardV2 chartCard = new ChartCardV2();

        chartCard.chartName = json[i]["chartName"].ToString();
        chartCard.selectedChartFileId = json[i]["selectedChartFileId"].ToString();
        ChartDatakeyValue[json[i]["chartName"].ToString()] = json[i]["selectedChartFileId"].ToString();
    }
    IsGetData = true;
    SetData();
}
public T ChartData<T>() where T : class
{
    Type type = typeof(T);
    string classname = type.Name;
    string chartId = ChartDatakeyValue[classname];
    var bro = Backend.Chart.GetChartContents(chartId);

    if (bro.IsSuccess() == false)
    {
        Debug.LogError($"{chartId}의 차트를 불러오는 중, 에러가 발생했습니다. : " + bro);
        return null;
    }
    JsonData json = bro.FlattenRows();


    T data = null;

    for (int i = 1; i < json.Count; ++i)
    {
        object myInstance = Activator.CreateInstance(type);

        foreach (var field in type.GetFields())
        {
            field.SetValue(myInstance, Convert.ChangeType(json[i][field.Name].ToString(), field.FieldType));
        }

        data = (T)myInstance;
    }
    return data;
}
```

소개 영상 :

https://www.youtube.com/watch?v=5jtGv-boAaE

https://www.youtube.com/shorts/qLWoEDDnyAk

BJ 영상 :

https://www.youtube.com/watch?v=13VLe-coavs

https://www.youtube.com/watch?v=gFHRfyNiE-s

봐주셔서 감사합니다.

더 자세한코드는 회사 자산이므로
보여드릴수 없는점 양해부탁드립니다.