

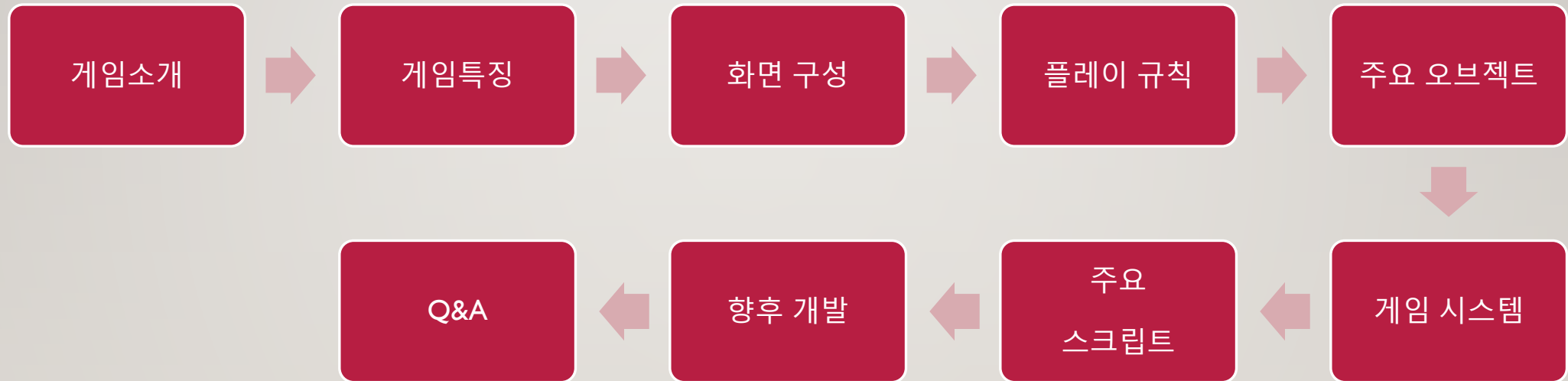
# MATCH&RAID

---

2084029 이상헌

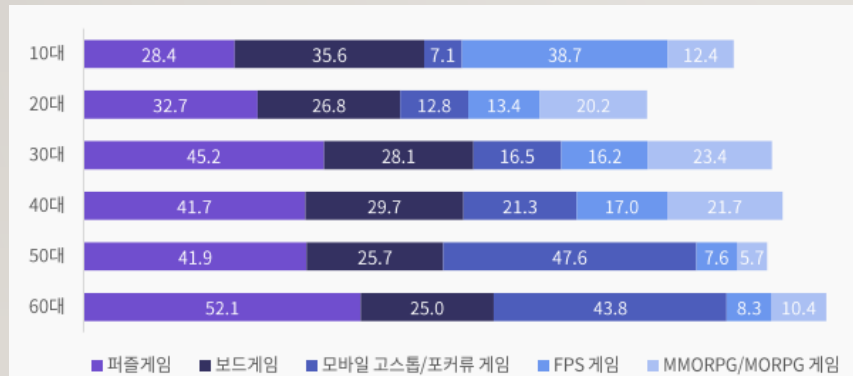
# 목차

---



# 게임 소개

- 게임이름: Match&Raid
- 게임장르: 퍼즐 게임
- 플랫폼: 모바일
- 타겟 유저: 30대 이상의 중년층



# 게임 특징

---

- 각 캐릭터를 상징하는 퍼즐을 맞출 시 해당 캐릭터가 공격한다.
- 각 캐릭터는 공격력이 다르기에 유의해야한다.
- 한번에 클리어가 아닌 여러 번 시도하며 영웅을 강화하는 게임이다.



# 화면 구성



# 화면 구성

플레이어 캐릭터

적

퍼즐

배경



# 플레이 규칙

---

- 기본적인 UI 상호작용은 탭방식을 사용하며 퍼즐을 맞출 때는 손가락을 사용한 슬라이드 방식을 사용한다
- 게임의 제한시간은 60초로 타이머가 0초가 되거나 적을 쓰러트리면 종료된다.
- 타이머가 0초가 되면 클리어 골드를 적의 남은 체력에 따라 지급한다.
- 타이머가 끝나기 전 적을 잡으면 클리어 골드를 2배로 받는다.

# 주요 오브젝트

---





# 게임 시스템

---

- 플레이 한번 할 때마다 하트라는 재화를 소모한다
- 하트는 현실시간 5분에 하나씩 충전되며 최대 5개 보유가 가능하다
- 각 퍼즐의 색상마다 공격하는 캐릭터가 다르다
- 똑같은 색상의 퍼즐을 일정 횟수를 맞추는 경우 비교적 강한 공격이 가능하다
- 영웅 업그레이드 부분에서 각 캐릭터의 공격력과 강한 공격을 가할 때 추가되는 데미지를 올릴 수 있다.

# 주요 스크립트

---

- 모바일 게임의 핵심 기능 중 하나인 시간에 따른 재화 충전 기능 스크립트

```
public void Init()
{
    if (PlayerPrefs.HasKey("HeartAmount"))
    {
        Debug.Log("PlayerPrefs has key : HeartAmount");
        m_HeartAmount = PlayerPrefs.GetInt("HeartAmount");
        if (m_HeartAmount < 0)
        {
            m_HeartAmount = 0;
        }
    }
    else
    {
        m_HeartAmount = 5;
    }
    m_RechargeRemainTime = 0;
    m_AppQuitTime = new DateTime(1970, 1, 1).ToLocalTime();
    //heartRechargeTimer.text = string.Format("Timer : {0} s", m_RechargeRemainTime);
}
```

```
public bool SaveAppQuitTime()
{
    Debug.Log("SaveAppQuitTime");
    bool result = false;
    try
    {
        var appQuitTime = DateTime.Now.ToLocalTime().ToBinary().ToString();
        PlayerPrefs.SetString("AppQuitTime", appQuitTime);
        PlayerPrefs.Save();
        Debug.Log("Saved AppQuitTime : " + DateTime.Now.ToLocalTime().ToString());
        result = true;
    }
    catch (System.Exception e)
    {
        Debug.LogError("SaveAppQuitTime Failed (" + e.Message + ")");
    }
    return result;
}
```

```
public bool LoadAppQuitTime()
{
    Debug.Log("LoadAppQuitTime");
    bool result = false;
    try
    {
        if (PlayerPrefs.HasKey("AppQuitTime"))
        {
            Debug.Log("PlayerPrefs has key : AppQuitTime");
            var appQuitTime = string.Empty;
            appQuitTime = PlayerPrefs.GetString("AppQuitTime");
            m_AppQuitTime = DateTime.FromBinary(Convert.ToInt64(appQuitTime));
        }
        Debug.Log(string.Format("Loaded AppQuitTime : {0}", m_AppQuitTime.ToString()));
        //appQuitTimeLabel.text = string.Format("AppQuitTime : {0}", m_AppQuitTime.ToString());
        result = true;
    }
    catch (System.Exception e)
    {
        Debug.LogError("LoadAppQuitTime Failed (" + e.Message + ")");
    }
    return result;
}
```

```
public void SetRechargeScheduler(Action onFinish = null)
{
    if (m_RechargeTimerCoroutine != null)
    {
        StopCoroutine(m_RechargeTimerCoroutine);
    }
    var timeDifferenceInSec = (int)((DateTime.Now.ToLocalTime() - m_AppQuitTime).TotalSeconds);
    var heartToAdd = timeDifferenceInSec / HeartRechargeInterval;
    var remainTime = timeDifferenceInSec % HeartRechargeInterval;
    m_HeartAmount += heartToAdd;
    if (m_HeartAmount >= MAX_HEART)
    {
        m_HeartAmount = MAX_HEART;
    }
    else
    {
        m_RechargeTimerCoroutine = StartCoroutine(DoRechargeTimer(remainTime, onFinish));
    }
}
```



# 향후 개발

---

- 아트 리소스를 충당하여 그래픽의 통일성을 부여
- 퍼즐 기믹의 보완
- 상점 요소 등 장기간 플레이에 필요한 콘텐츠 추가
- 전체적인 재 빌드를 통한 정리 및 체제화

# 플레이 영상



# Q & A

---

