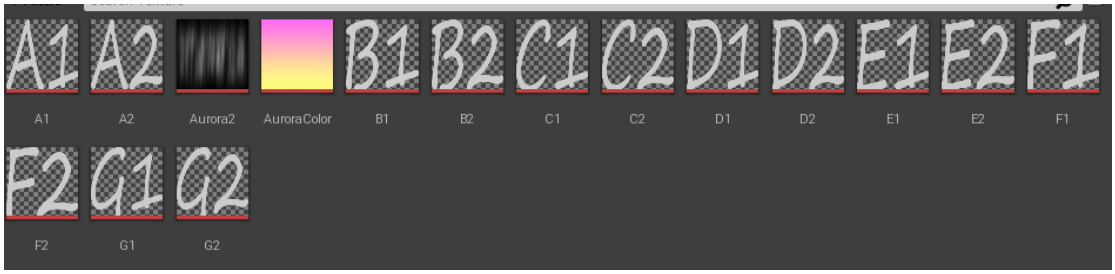
This week I tried to polish the game by adding some visual feedbacks when the player interacts with the game since they are pretty lacking at this stage. I made a system that will spawn an UI overlay on stars depends on the input of the player so that it helps the player to remember what answer they have given for each star. The visual feedback is made with particle effects and I programmed a system through c++ to achieve the functionalities, such as display player's entry and clean up particles when the player removed the entry.

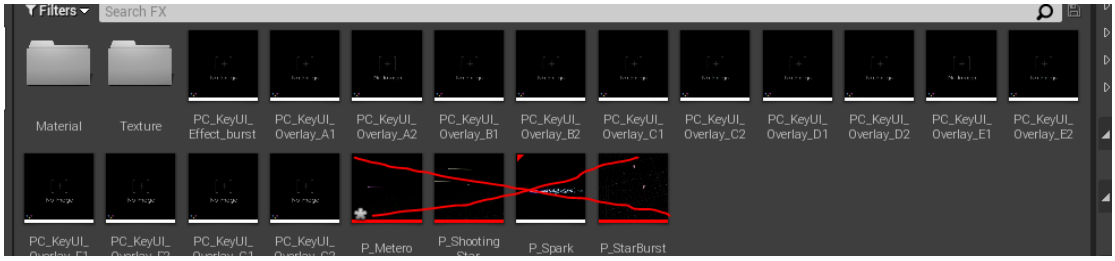
UI overlay and test video:

https://youtu.be/Ca2qe84yrKY

Overlay assets created:
Textures:



Overlay particles



C++ functions:
Determine which star to spawn overlay

```cpp
void APuzzleTemplate::DeterminCurrentStar()
{
    for (int i = 0; i < arrayStarDefault.Num(); ++i)
    {
        if (arrayStarDefault[i]->getOrderNo() == (arrayPlayerEntry.Num() + 1))
        {
            arrayStarDefault[i]->SetIsCurrentStar();
        }
        if (arrayStarDefault[i]->getOrderNo() != (arrayPlayerEntry.Num() + 1))
        {
            arrayStarDefault[i]->SetIsNotCurrentStar();
        }

    }
}
```

Overlay particle clean up function:

```cpp
    //Remove Key UI Overlay
    if (arraySpawnedKeyUIOverlayParticles.Num() > 0)
    {
        arraySpawnedKeyUIOverlayParticles[arraySpawnedKeyUIOverlayParticles.Num() - 1]->DestroyComponent();
        arraySpawnedKeyUIOverlayParticles.RemoveAt(arraySpawnedKeyUIOverlayParticles.Num() - 1);
    }
    //Debug
    //UE_LOG(LogTemp, Warning, TEXT("The integer value is: %d"), arraySpawnedKeyUIOverlayParticles.Num());
}
```

The function to spawn overlay

```cpp
//For StarDefault
void APuzzleTemplate::SpawnKeyUIOverlayParticles(EnumAnswer key)
{
    for (int i = 0; i < arrayAnswer.Num(); ++i)
    {
        FVector spawnLocation = arrayStarDefault[i]->GetActorLocation() + keyUISpawnLocationOffset;
        FVector spawnEffectLocation = arrayStarDefault[i]->GetActorLocation() + keyUIEffectSpawnLocationOffset;

        //Key UI Overlay
        switch (key)
        {
        case EnumAnswer::N:
            break;
        case EnumAnswer::C1:
        {
            if (KeyUIOverlayParticleC1 != NULL)
            {
                KeyUIOverlayParticle = KeyUIOverlayParticleC1;
            }
            if (arrayStarDefault[i]->GetIsCurrentStar())
            {
                UE_LOG(LogTemp, Warning, TEXT("UI spawn command executed."))
                    KeyUIOverlayComp = UGameplayStatics::SpawnEmitterAtLocation(this, KeyUIOverlayParticle,
                        spawnLocation, FRotator::ZeroRotator,
                        true,
                        EPSCPoolMethod::None,
                        true);

                //For despawning the particle UI when redo is pressed, see RemoveLastPlayerEntry()
                arraySpawnedKeyUIOverlayParticles.Add(KeyUIOverlayComp);

                //Key UI Spawn Effect
                if (KeyUISpawnEffect != NULL)
                {
                    KeyUIOverlayComp = UGameplayStatics::SpawnEmitterAtLocation(this, KeyUISpawnEffect,
                        spawnEffectLocation, FRotator::ZeroRotator,
                        true,
                        EPSCPoolMethod::None,
                        true);
                }
            }
        }
        break;
        case EnumAnswer::D1:
```

```
break;
case EnumAnswer::D1:
{
    if (KeyUIOverlayParticleD1 != NULL)
    {
        KeyUIOverlayParticle = KeyUIOverlayParticleD1;
    }
    if (arrayStarDefault[i]->GetIsCurrentStar())
    {
        UE_LOG(LogTemp, Warning, TEXT("UI spawn command executed."))
            KeyUIOverlayComp = UGameplayStatics::SpawnEmitterAtLocation(this, KeyUIOverlayParticle,
                spawnLocation, FRotator::ZeroRotator,
                true,
                EPSCPoolMethod::None,
                true);
        arraySpawnedKeyUIOverlayParticles.Add(KeyUIOverlayComp);

        //Key UI Spawn Effect
        if (KeyUISpawnEffect != NULL)
        {
            KeyUIOverlayComp = UGameplayStatics::SpawnEmitterAtLocation(this, KeyUISpawnEffect,
                spawnEffectLocation, FRotator::ZeroRotator,
                true,
                EPSCPoolMethod::None,
                true);
        }
    }

}
break;
case EnumAnswer::E1:
{
    if (KeyUIOverlayParticleE1 != NULL)
    {
        KeyUIOverlayParticle = KeyUIOverlayParticleE1;
    }
    if (arrayStarDefault[i]->GetIsCurrentStar())
    {
        UE_LOG(LogTemp, Warning, TEXT("UI spawn command executed."))
            KeyUIOverlayComp = UGameplayStatics::SpawnEmitterAtLocation(this, KeyUIOverlayParticle,
                spawnLocation, FRotator::ZeroRotator,
                true,
                EPSCPoolMethod::None,
                true);
        arraySpawnedKeyUIOverlayParticles.Add(KeyUIOverlayComp);
```