Gameplay Ability System: Ability Management and Workaround

One of the biggest issue I have faced is using the FGameplayAbilitySpecHandle to remove specific abillity from the AbilitySystemComponent. However, there is a work-around by having a saved set of ability and clearing all abilities in the AbilitySystemComponent.

For example: I will define four ability slots as our save slots in our CharacterBase.h

```
UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "CharacterBase" )
    TSubclassOf<UGameplayAbility> SkillOneSlot;

UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "CharacterBase" )
    TSubclassOf<UGameplayAbility> SkillTwoSlot;

UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "CharacterBase" )
    TSubclassOf<UGameplayAbility> SkillThreeSlot;

UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "CharacterBase" )
    TSubclassOf<UGameplayAbility> SkillFourSlot;
```

We then will create each ability a function that will equip abilities onto the save slots.

```
void EquipSkillSlotOne(TSubclassOf<UGameplayAbility> Ability)
{
    SkillOneSlot = Ability;
    RefreshAbility()
}

void EquipSkillSlotTwo(TSubclassOf<UGameplayAbility> Ability)
{
    SkillTwoSlot = Ability;
    RefreshAbility()
```

https://md2pdf.netlify.app

The RefreshAbility() function is something we will create as well, which will clear all abilities in the AbilitySystemComponent and re=add all SkillOneSlot - SkillFourSlot back into it. Any new modification will replace the old slot.

```
void RefreshAbility()
{
    AbilitySystemComponent->ClearAllAbilities();
    if(SkillOneSlot)
    {
        AcquireAbility(SkillOneSlot);
    }
    ...
}
```

https://md2pdf.netlify.app