Title

화면 아무곳이나 클릭하면 다음으로 넘어갑니다. 특별한 기능은 없습니다.

Lobby

화면 버튼 중 "새로시작하기", "이어하기", "게임종료"가 구현되었습니다.

"이어하기"를 클릭하면 스크롤팝업이 활성화됩니다.

Content를 보시면 "저장시간", "남은기간", "소지금액"으로 총 3개의 저장 정보와 하단의 이어하기 버튼을 확인하실 수 있습니다.

저장시간 :

플레이어가 게임을 저장한 시각을 System.DateTime구조체를 활용해서 저장했습니다.

GameManager의 자식객체인 PlayerPrefsManager의 스크립트 254줄을 확인하시면 다음과 같은 문장을 확인하실 수 있습니다.

```
string SavedDate = DateTime.Now.ToString("yyyy년 MM월 dd일:HH시 mm분 ss초");
```

이 문장을 현재의 플레이어 키에 PlayerPrefs를 사용하여 저장했습니다.

Lobby에서 "이어하기"를 선택하면 해당키에서 정보를 불러와서 화면에 플레이어가 해당 정보를 어느 시점에 저장했는지 알 수 있습니다.

해당 내용은 SavedPlayerDataPanel 스크립트의 줄18~29에서 확인하실 수 있습니다.

남은기간 :

게임 내의 종료시점까지 남은기간을 표현합니다. 해당내용은 "GameManager"스크립트의 줄 56에 있는 변수 "_currentRemainingPeriod"에 의해 관리되며 마찬가지로 PlayerPrefs로 저장하고 불러왔습니다.

소지금액 :

게임 내에서 통용되는 화폐로 "코인" 단위를 사용합니다. 해당내용은 "PlayManager"스크립트의 줄 30에 있는 변수 "currentPlayerStatus"에 의해 관리되며 마찬가지로 PlayerPrefs로 저장하고 불러왔습니다.

이어하기 :

해당 버튼을 누르면 "GameManager" 줄 52에 있는 변수 "_currentPlayerSaveKey"를 초기화 시키고 "InGame"씬으로 씬전환을 합니다. 그러면 "GameManager" 줄 322에 있는 함수" OnSceneLoaded"가 씬로드 후에 실행되어 현재 플레이어 키를 확인하고 "새로시작하기"인지 "이어하기"인지를 확인하여 "이어하기"를 선택한 경우 필요한 모든 정보를 플레이어키를 활용하여 불러옵니다.

함수" OnSceneLoaded"는 "GameManager" 줄 163을 보시면 "Enable"에서 씬로드시 호출함수에 추가되었고 "Disable"에서 삭제되어 안전하게 사용되었음을 알 수 있습니다.

InGame

텍스트 팝업:

해당 팝업은 특정 조건에 만족되면 활성화되어 코루틴으로 글자를 한글자씩 화면에 보여줍니다. 해당 내용은 "TextWindowView"스크립트에서 확인이 가능합니다.

팝업창이 올라오면 팝업의 아무곳이나 클릭하여 타이핑을 멈추고 전체 문장을 보여주거나 다음 대화로 넘어갈 수 있습니다.

대화도중 선택지가 나올 수 있는데 어떤 선택지를 선택하느냐에 따라서 "CallbackManager"스크립트에서 호출될 함수가 선택됩니다. 해당내용은 "CallbackManager"의 줄 88에 있는 함수 "CallBackList_DefaultText"와 "TextWindowView"의 줄 138에서 확인 하실 수 있습니다.

마찬가지로 대화팝업중 특정 문장이 종료되면 실행되는 콜백함수도 존재합니다. 해당 내용은 "TextWindowView"의 줄 251에서 확인 하실 수 있습니다.

텍스트팝업의 오른쪽 상단의 스킵버튼을 누르면 선택지가 나오는 대화까지 내용을 스킵하거나 선택지가 없는경우 즉시 팝업이 종료됩니다. 그 과정에서 문장 종료시 실행되는 콜백함수는 정상적으로 작동되며 해당 콜백함수가 다른 주제를 가지고 텍스트 팝업을 새로 띄우는 경우도 존재합니다.

게임 내 상호작용 가능한 객체:

게임에서 캐릭터가 이동하다 보면 특정 개체의 앞에 섰을 때 오른쪽 하단의 "!" 문자가 들어간 인터페이스가 활성화 되는 것을 확인하실 수 있습니다. 그럴 경우 해당 인터페이스를 클릭하면 대상과의 상호작용 과정을 텍스트팝업으로 플레이어에게 보여줍니다.

오른쪽 상단의 아이콘뷰:

오른쪽 상단의 화살표를 클릭하면 플레이어가 사용 가능한 여러 팝업들을 열 수 있는 아이콘들이 화면으로 들어옵니다. 해당 화살표를 클릭하고 5초가 지나면 코루틴을 사용하여 자동으로 닫히게 만들었습니다.

플레이어가 특정 조건을 만족하면 아이콘의 잠금이 해방됩니다. 각 아이콘을 클릭하면 해당하는 팝업이 활성화됩니다. 해당 내용은

팝업:

"선택팝업", "확인팝업"을 제외한 모든 팝업은 "PopUpBase"를 상속받고 해당 Class는 "MemoryPool_Queue"를 상속받고 있습니다. Content에 들어갈 객체들은 Prefeb으로 만들어져 있으며 개별 팝업스크립트에서 연결하여 팝업이 활성화 될 때마다 필요한 객체의 수를 확인 후 객체를 초기화 시켜서 사용하고 있습니다. 예시로서 "InventoryPopUp"스크립트를 보시기 추천드립니다.

퀘스트 팝업:

퀘스트 팝업은 "퀘스트 목록"과 "퀘스트 내용"으로 총 2가지가 있습니다.

아이콘에서 "퀘스트"를 클릭하면 "퀘스트 목록"팝업이 활성화되며 퀘스트 목록에서 특정 퀘스트를 클릭하면 "퀘스트 내용" 팝업이 활성화되어 상세한 내용을 알 수 있습니다. 상세한 구조는 아직 만드는 중입니다.

아이템 사용:

아이템은 "사용가능한 경우"와 사용할 수 없고 "확인만 가능한 경우"로 나뉩니다. 사용가능한 경우 아이템 클릭시 선택지 팝업이 활성화되며 팝업의 활성화시점에서 아이템에 따라 사용하기 버튼 클릭시 실행할 콜백함수가 변경됩니다.

"확인만 가능한 경우"는 확인 팝업이 활성화되며 특별한 기능 없이 아이템 상세 내용과 팝업 종료만 가능합니다.

자세한 내용은 "InventoryPopUp"스크립트의 함수 "RefreshPopUp"에서 확인 하실 수 있습니다.

게임 저장하기:

자동저장기능은 일부러 넣지 않았습니다. 플레이어가 Stage2를 시작한 이후부터 집으로 돌아가서 시계와 상호작용을 하면 저장이 가능하도록 만들었습니다.

다음날로 날짜 변경하기:

Stage2 이후부터 플레이어가 집으로 돌아와 침대와 상호작용하면 플레이어가 잠을 자서 다음 날로 넘어갈 수 있습니다.

아직 게임 내에 구현되지 않았지만 카지노에서 모든 게임은 하루에 1번만 입장이 가능하도록 만들 생각입니다. 게임을 포기하거나 다른 사람들의 지갑을 다 털어서 카지노에서 나온 경우 침대로 돌아와 상호작용을 하여 다음 날짜로 이동하고 동시에 게임 내에 종료시점까지 남은 시간을 카운팅 할 생각입니다.

-카지노에서-

카지노에 가면 게임을 선택할 수 있습니다.(아직 1개만 존재합니다.)

게임에 입장하면 아이콘뷰에서 게임어시스턴트가 활성화됩니다. 해당 팝업은 게임 플레이를 돕기 위한 여러가지 기능을 제공합니다.

OnlyOneLives 게임을 선택하면 왼쪽 하단에 아이콘이 2개 보이실 겁니다.

왼쪽의 물음표는 게임의 룰을 설명해줍니다. 스크롤 관련해서는 현재 수정 중에 있습니다.

오른쪽의 돋보기를 활성화시 게임 화면을 확대하여 보여줍니다. 게임 진행 중 상대방의 카드를 자세하게 확인하기 위한 용도입니다.

게임을 진행하면 카지노 딜러와 시스템이 필요한 인터페이스 조작법과 간략한 룰을 설명해줍니다.

주사위 :

서브카메라와 raw Image를 사용하여 주사위 뷰를 만들었습니다. 주사위의 6면에 각각 숫자를 이름으로하는 빈 객체를 붙여놨고 값을 알 수 있도록 "diceChecker"라는 이름의 객체로 해당하는 숫자를 읽어와 게임에 사용됩니다.

오른쪽 하단의 "내 카드 확인하기":

주사위를 굴리고 나서 모든 카드가 분배되면 활성화되는 인터페이스 입니다.

해당 인터페이스로 자신이 갖고있는 카드를 확인하고 선택하여 제시할 수 있습니다.

"선택한 카드 오픈하기" 또는 "선택한 카드 제시하기"는 카드를 다 선택하는 것과 특정 조건이 완료될 때만 활성화 됩니다.

카드 뷰:

"내 카드 확인하기" 버튼을 누르셨다면 서브스크린이 올라올 것입니다. 이 화면도 서브카메라와 raw Image를 활용했습니다. 화면에 보이는 숫자 버튼은 "MemoryPool_Stack "를 상속받은 "CardButtonMemoryPool"스크립트에서 관리되고 있습니다. 카드 개수는 계속해서 달라질 수 있기 때문에 개수에 상관없이 카드를 꺼낼 때 0번 자식부터 꺼내기 위해 메모리풀을 스택으로 설정했습니다. 자세한 내용은 "CardButtonMemoryPool"의 줄70에 있는 함수 "InitCardButton"을 확인하시면 됩니다.