public class ConnDB

DB의 연결정보가 담겨있는 클래스이다.

EscMenuManager

Esc키를 눌렀을 때 게임을 일시정지하고 팝업창을 띄우는 클래스이다.

전역변수

private static EscMenuManager \_instance; // ESC메뉴의 인스턴스

public GameObject escMenuBackGround; // 설정 패널 오브젝트

private bool \_settingActive; // 설정 화면 활성화 여부

public static EscMenuManager Instance

싱글톤 패턴을 사용하여 ESC메뉴 인스턴트를 가져온다.

private void Start()

씬이 로드되면 escMenuBackGround를 비활성화 한다.

private void Update()

Esc키가 눌렸을 때 \_settingActive가 false이면 PauseGame(), ActivateEscMenu() 함수를 호출하고,   
\_settingActive가 true이면 ResumeGame(), DeactivateEscMenu() 함수를 호출한다.

public void OnClickConfig()

설정창을 활성화 하는 함수

public void OnClickLoad()

로드창을 활성화 하는 함수

public void OnClickExitGame()

게임을 종료하는 함수

private void PauseGame()

게임을 일시정지 하는 함수

private void ResumeGame()

게임을 재개하는 함수

private void ActivateEscMenu()

EscMenuBackGround 오브젝트를 활성화 하는 함수

private void DeactivateEscMenu()

EscMenuBackGround 오브젝트를 비활성화 하는 함수

public class MainMenuManager

public void OnClickNewStart()

InitUserScene 씬을 로드한다

public void OnClickLoadStart()

MainLevelScene 씬을 로드한다

public void OnClickExitGame()

게임을 종료한다

public class InitUserManager

전역변수

private GameObject \_myGameObject; // StartLevelDao를 담는 빈 오브젝트

private StartLevelDao \_sld; // StartLevelDao를 사용하기 위한 변수

public InputField inputUserNameField; // 사용자 이름 입력 필드

public Dropdown inputUserSexDropDown; // 사용자 성별 입력 드롭다운

public void Awake()

\_myGameObject에 StartLevelDao 컴포넌트를 추가하여 \_sld에 대입한다

public void OnClickEnterUserBtn()

StartLevelDao.SetUserInfo(userName, userSex)를 호출해서 DB에 아빠의 이름과 성별을 저장한다

\_sld.GetUserInfo()를 호출하여 반환값을 userInfo에 대입한다

userInfo를 Json파일에 저장한다

FirstSetDStat 씬을 로드한다

public class StartLevelDao

StartLevel의 DB통신 관련된 기능을 하는 함수가 저장된 클래스

전역변수

private ConnDB \_connDB; // ConnDB 클래스를 참조하는 변수

private void Awake()

DB 정보를 받아온다

public static void SetUserInfo(string name, string gender)

name과 gender를 DB에 저장하는 함수

public static Dictionary<string, string> GetUserInfo()

DB에서 가장 마지막에 저장 된 name과 gender를 불러오는 함수

public class LifeTimeManager

LifeTimeScene 씬의 동작을 제어하는 클래스

전역변수

private LifeTimeGo \_ltgo; // LifeTime의 GameObject들을 한번에 관리하기 위한 클래스

private LifeTimeVo \_ltvo; // LifeTime의 변수를 한번에 관리하기 위한 클래스

private GameObject \_myGameObject; // StartTurnDao를 가져오기 위한 GameObject

private StartTurnDao \_std; // StartTurnDao 클래스 참조

public void Start()

ltgo, ltvo, std의 생성자를 호출한다

InitTodoList()를 호출하여 TodoList에 값을 넣어준다

public void OnClickDateBtn(GameObject button)

달력의 날짜를 클릭했을 때의 동작을 제어하는 함수이다

원래 선택되어 있던 버튼의 테두리를 비활성화 하고 클릭한 버튼의 테두리를 활성화한다

public void OnClickTodoBtn(GameObject button)

TodoList목록 버튼을 클릭했을 때의 동작을 제어하는 함수이다

\_ltvo.IsSelectable이 false이면 실행되지 않는다

매개변수로 넘어 온 버튼(클릭 한 버튼)의 TodoNameComponentVo 컴포넌트를 참조하여 todoName, reward, loseReward, statReward를 dateBtn의 TodoNameComponentVo 컴포넌트에 저장한다

이 후 선택되어 있던 날짜의 버튼의 색을 FindColor(todoNameComponent.index) 함수를 호출하고, ChangeImageColor(\_ltvo.IsSelectDate, color) 함수를 호출하여 컴포넌트에 저장 된 index값에 따라 색을 지정한다

ltvo.IsSelectDate의 4번째 자리 값(정수)를 1 증가하여 다음 날짜로 넘어가는 기능 구현 및 20일 일 때는 ltvo.IsSelectable을 false로 설정하여 비활성화한다

public void OnClickComplete()

CalenderLayout의 요소들을 순회(day1 ~ day20)하여 각 요소의 값들을 ltvo.PlanList에 딕셔너리 리스트형으로 저장한다

OnClickDateBtn(GameObject.Find("Day1")) 함수를 호출하여 선택된 날짜를 day1로 초기화한다

\_ltgo.StartTurn.SetActive(true) 함수를 호출하여 ltgo.StartTurn 오브젝트 이미지를 활성화한다

StartTurn() 함수를 호출하여 턴을 시작한다

public void OnClickNextPhase()

버튼이 클릭되면 현재 시간을 증가시킨다

저녁이면 아침으로 넘어가고, 일을 1 증가 시킨다

20일이면 턴을 종료한다

public void OnClickDelete()

선택된 날짜의 일정을 삭제하는 함수

ltvo.IsSelectable이 false이면 실행되지 않는다

선택한 날짜의 색, 텍스트를 제거한다

DeleteDateComponent(dateBtn) 함수를 호출하여 선택된 날짜의 TodoNameComponentVo 컴포넌트 값을 초기화한다

public void OnClickReturn()

StartTurnScene씬을 로드하는 함수

뒤로 가기 기능

private void InitTodoList()

TodoList 목록을 불러오는 함수

ltgo.StartTurn 이미지를 비활성화하고, ltgo.TodoList 이미지를 활성화한다(부모 오브젝트)

\_std.GetTodoNo(\_ltvo.NowYear, \_ltvo.NowMonth) 함수를 호출하여 반환값을 noList에 대입한다

\_std.GetTodoList(noList) 함수를 호출하여 \_ltvo.TodoList 에 대입한다

현재 날짜를 nowDateComponent 에 대입하여 현재 날짜를 표기한다

\_ltvo.TodoList 딕셔너리 리스트를 순회하며 해당 년월에 있는 일정을 todoList 목록에 버튼을 동적으로 생성한다

\_ltgo.TodoList.SetActive(false) 함수를 호출하여 부모 오브젝트를 비활성화 하고 첫 날의 외곽선을 활성화하여 Day1을 선택한다(초기화)

private void StartTurn()

턴 시작을 구현한 함수

각 텍스트 컴포넌트를 참조하여 현재 시각을 표기한다

timeTxtComponent의 경우 0이면 아침, 1이면 점심, 2면 저녁이다

아침, 저녁이면 대화 버튼을 활성화하고, 점심이면 비활성화한다

\_ltvo.PlanList의 “TODONAME”이 비어있으면 쉬는날로 일정 이름을 정하고, 보상, 소모 재화, 얻는 스탯을 비워둔다

private void EndTurn()

턴이 끝나는 기능을 구현 한 함수

\_ltgo.StartTurn.SetActive(false) 함수를 호출하여 StartTurn 이미지를 비활성화하고, 현재 12월이 아니면 월 증가, 일 초기화하고, 12월이면 년도 증가, 월, 일을 1로 초기화한다

\_ltvo.PlanList.Clear() 함수를 호출하여 PlanList를 비운다

RemoveTodoList(), RemoveCalendar(), InitTodoList() 함수를 호출하여 TodoList, Calender를 비우고, TodoList를 바뀐 년월에 맞게 재배치한다

private void RemoveTodoList()

\_ltgo.TodoListLayout 을 순회하며 자식 오브젝트를 Destroy한다

private void RemoveCalendar()

\_ltgo.CalenderLayout 을 순회하며 자식 오브젝트의 색을 흰색으로 칠하고, 텍스트 컴포넌트를 “”로 초기화 후 DeleteDateComponent(date.gameObject) 함수를 호출하여 TodoNameComponentVo 컴포넌트를 초기화한다

private static void DeleteDateComponent(GameObject obj)

TodoNameComponentVo 컴포넌트의 값을 초기화 하는 함수

private static void ChangeImageColor(string objectName, Color color)

object의 이름과 색을 매개변수로 받아 해당 이미지 오브젝트의 색을 바꾸는 함수이다

private static Color FindColor(int index)

TodoList의 인덱스값을 받아 해당하는 인덱스에 맞는 색을 반환하는 함수이다

public class StartTurnDao

private void Awake()

클래스가 호출되면 DB의 연결 정보를 받아온다

public List<int> GetTodoNo(int year, int month)

현재 날짜의 연, 월을 입력받아 해당하는 TodoNO를 반환하여 리스트에 저장하는 함수이다

public List<Dictionary<string, object>> GetTodoList(List<int> noList)

TodoNO를 이용하여 TodoList를 가져와 리스트에 저장하는 함수이다

public class LifeTimeGo

LifeTime의 GameObject들을 한번에 관리하기 위한 클래스이다

public readonly GameObject YearTxt = GameObject.Find("YearTxt"); // 년도 텍스트 참조

public readonly GameObject MonthTxt = GameObject.Find("MonthTxt"); // 월 텍스트 참조

public readonly GameObject DateTxt = GameObject.Find("DateTxt"); // 일 텍스트 참조

public readonly GameObject TimeTxt = GameObject.Find("TimeTxt"); // 시간 텍스트 참조

public readonly GameObject TodoNameTxt = GameObject.Find("TodoNameTxt"); // TODO 이름 텍스트 참조

public readonly GameObject NowDate = GameObject.Find("NowDate"); // 현재 날짜 참조

public readonly GameObject TodoListPrefab = GameObject.Find("TODOList"); // TODOList 이미지 프리팹 참조

public readonly GameObject TodoList = GameObject.Find("TODOList"); // TODOList 이미지 참조

public readonly GameObject TodoListLayout = GameObject.Find("TODOListLayout"); // TODOList들이 들어갈 레이아웃 참조

public readonly GameObject CalenderLayout = GameObject.Find("CalenderLayout"); // 달력 레이아웃 참조

public readonly GameObject StartTurn = GameObject.Find("StartTurnBackGround"); // 시작 턴 이미지 참조

public readonly GameObject LifeTimeMain = GameObject.Find("LifeTimeMain"); // 라이프 타임 이미지 참조

public readonly GameObject ConvBtn = GameObject.Find("ConvBtn"); // 대화 버튼 참조

public GameObject TodoListInstance; // TODOList의 인스턴스

public class LifeTimeVo

LifeTime의 변수를 한번에 관리하기 위한 클래스이다

public int NowYear = 2024; // 년도 초기화

public int NowMonth = 4;

public int NowDate = 1; // 일 초기화

public int NowTime = 0; // 0: 아침, 1: 점심, 2: 저녁

public bool IsSelectable = true; // TODO 버튼 클릭 가능 여부

public string IsSelectDate = "Day1"; // 선택된 날짜

public readonly List<Dictionary<string, object>> PlanList = new(); // 달력에 적힌 일정을 담는 딕셔너리 리스트

public List<Dictionary<string, object>> TodoList = new(); // TODO리스트를 담는 딕셔너리 리스트

public class TodoNameComponentVo

TodoList의 정보가 담기는 클래스이다

public string todoName; // 일정 이름

public int reward; // 보상

public int loseReward; // 소모 재화

public string statReward; // 스탯 보상

public int index; // 인덱스