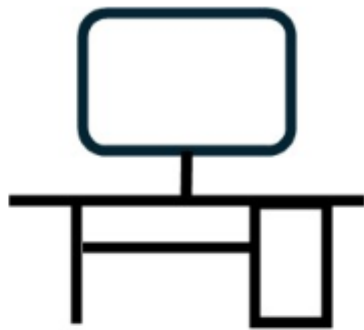


3. Design

Desco (Desk decorate)



DESCO

(Student No, Name, E-mail)

22313525, 박인석, ibsq123@yu.ac.kr

[Revision history]

Revision date	Version #	Description	Author
2025/05/31	1.00	First draft	박인석
2025/03/15	2.00	Class, sequence, state machine 다이어그램 수정	박인석

= Contents =

1. Introduction	
2. Class diagram	
3. Sequence diagram	
4. State machine diagram	
5. Implementation requirements	
6. Glossary	
7. References	

1. Introduction

- Summarize the contents of this document.
- Describe the important points of your design.
- 12pt, 160%.

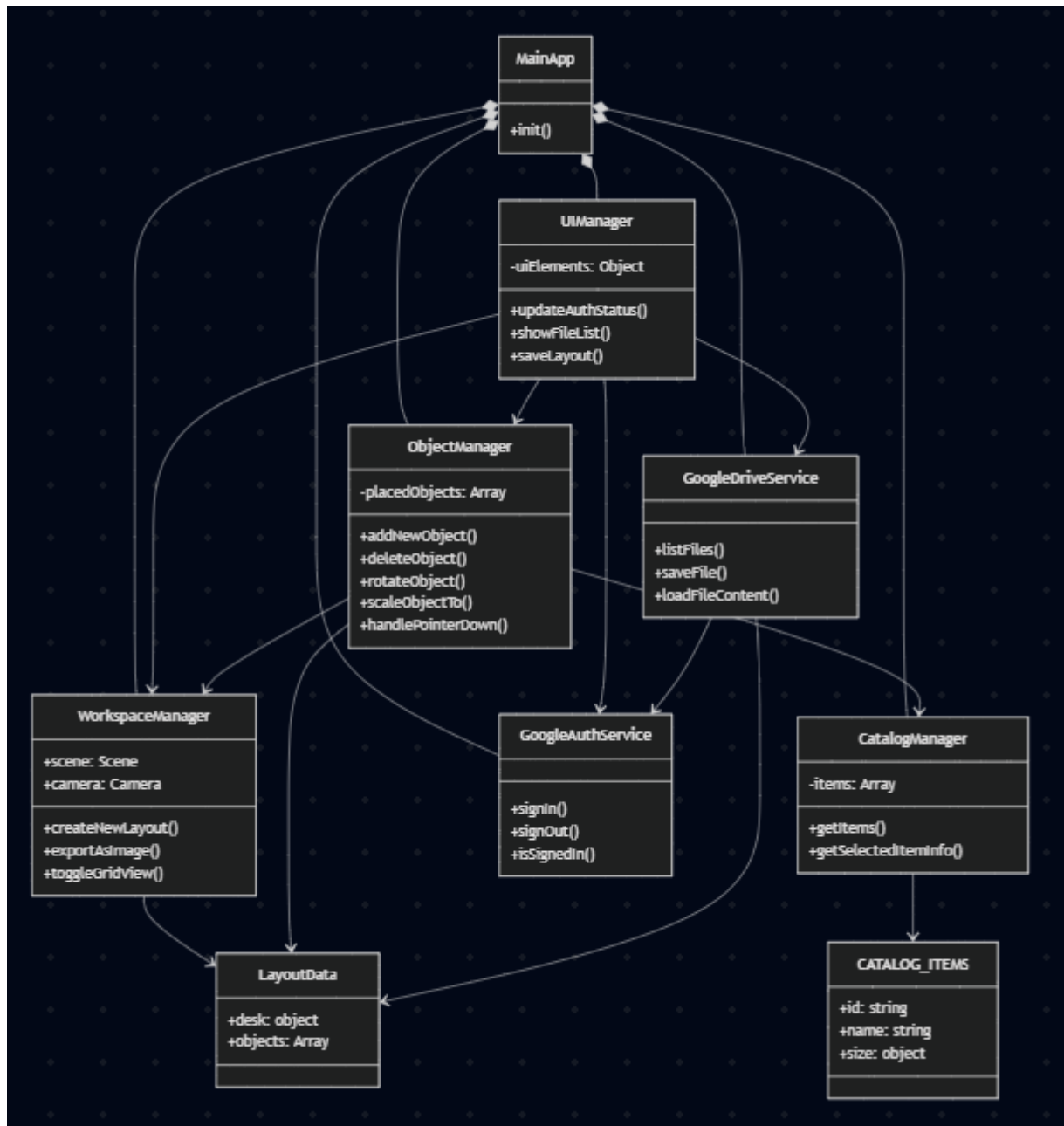
본 문서는 'Desco (Desk decorate)' 애플리케이션의 시스템 디자인을 설명합니다.

클래스 다이어그램을 통해 시스템의 정적 구조와 주요 구성 요소 간의 관계를 정의합니다. 시퀀스 다이어그램은 Google Drive 연동, 파일 저장 및 불러오기, 오브젝트 배치 및 수정, 뷰 관리, 이미지 내보내기 등 주요 기능에 대한 객체 간의 동적 상호작용 흐름을 상세히 보여줍니다. 스테이트 머신 다이어그램은 시스템 및 주요 객체의 생명 주기 동안 발생할 수 있는 다양한 상태 변화를 모델링합니다. 또한, 시스템 구현을 위한 운영 환경 요구사항과 본 문서에서 사용된 용어에 대한 정의를 포함합니다.

핵심 디자인 원칙은 모듈화된 아키텍처를 기반으로 하며, Google Drive API를 활용한 클라우드 기반 데이터 저장 및 사용자 인증 기능을 통합하는 데 중점을 둡니다. 이는 사용자에게 직관적인 데스크 꾸미기 경험을 제공하는 동시에, 데이터의 영속성과 접근성을 확보하기 위한 것입니다.

2. Class diagram

- Draw a class diagram.
- Describe each class in detail (attributes, methods, others) (table type).
- 12pt, 160%.



Class name	explanation
	Attributes: + scene: Scene 모든 3D 객체가 담기는 공간. + camera: Camera 3D 공간을 보는 시점.

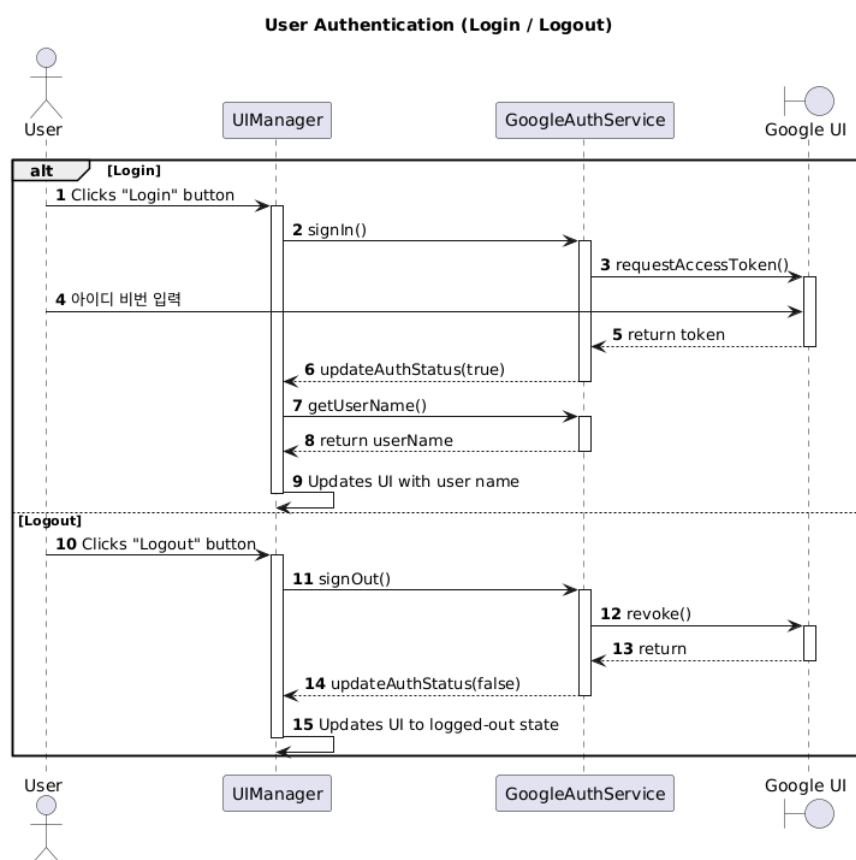
WorkspaceManager	Methods: + createNewLayout(): 새로운 책상 레이아웃을 생성 + exportAsImage(): 현재 뷰를 이미지 파일로 내보냄 + toggleGridView(): 책상 상판의 격자 모드를 토글
GoogleAuthService	Attributes: Methods: + signIn(): 로그인 프로세스를 시작합니다. + signOut(): 로그아웃 프로세스를 시작합니다. + isSignedIn(): 현재 로그인 상태를 확인합니다.
GoogleDriveService	Attributes: Methods: + listFiles(): 파일 목록을 조회합니다. + saveFile(): 파일을 저장합니다. + loadFileContent(): 파일 내용을 불러옵니다.
ObjectManager	Attributes: - placedObjects: Array: 씬에 배치된 모든 오브젝트의 목록. Methods: + addNewObject(): 새로운 오브젝트를 씬에 추가. + deleteObject(): 특정 오브젝트를 삭제 + rotateObject(): 특정 오브젝트를 회전 + scaleObjectTo(): 특정 오브젝트의 크기를 조절 + handlePointerDown(): 마우스 클릭 이벤트를 처리하

	여 오브젝트 선택 등의 동작을 시작
UiManager	Attributes: Methods: + updateAuthStatus(): 로그인 상태에 따라 UI를 업데이트 + showFileList(): 파일 목록을 화면에 표시 + saveLayout(): 현재 레이아웃을 저장하는 과정을 시작
CatalogManager	Attributes: - items: Array: 카탈로그 아이템 데이터의 배열 Methods: + getItems(): 전체 아이템 목록을 반환 + getSelectedItemInfo(): 선택된 아이템의 정보를 반환
CATALOG_ITEMS	Attributes: + id: <string> 고유 id + name: <string> 이름 + size: object 크기 Methods:
LayoutData	Attributes: + desk: object 정보 + objects: Array 있는 오브젝트 배열

	Methods:
--	-----------------

3. Sequence diagram

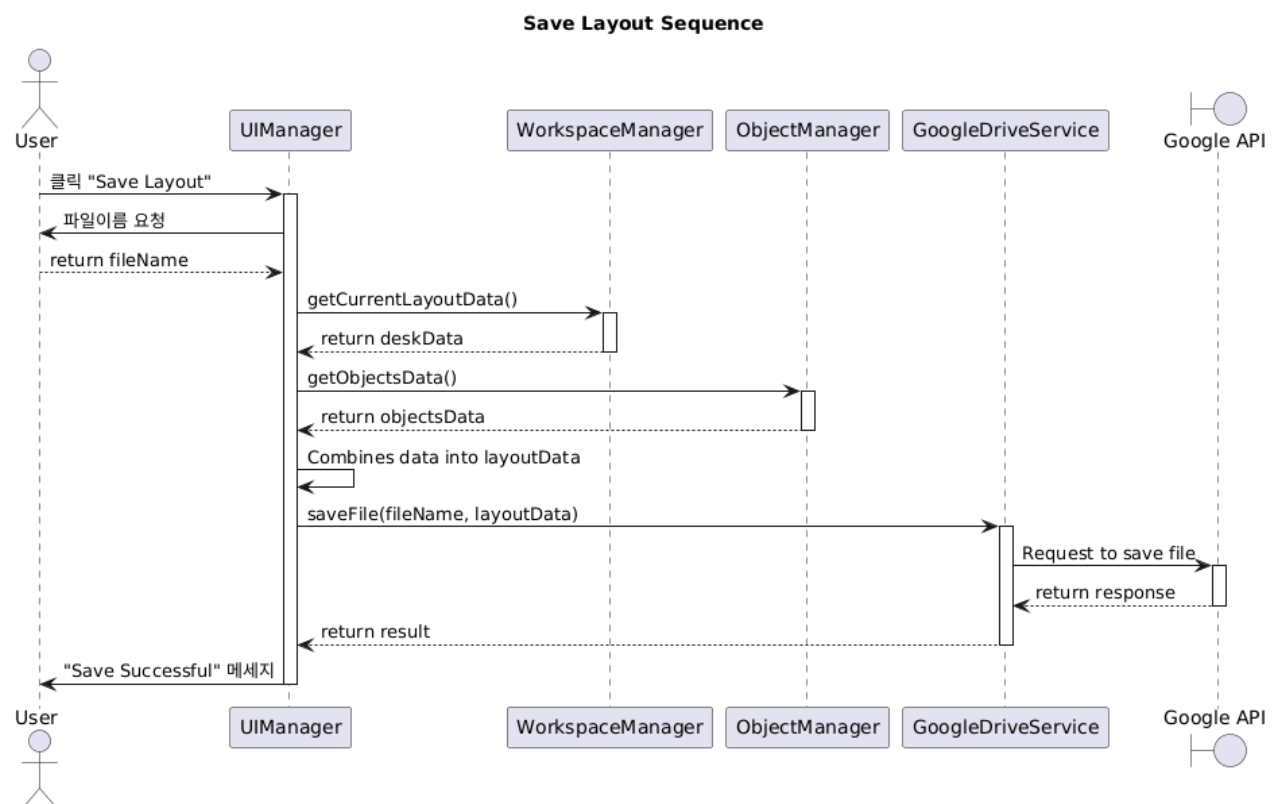
- Draw sequence diagrams for the whole functions of your system (this is related to the use cases you made in the Conceptualization phase).
- Explain each sequence diagram.
- 12pt, 160%.



사용자 인증 (로그인 및 로그아웃)

로그인: 사용자가 "Login" 버튼을 누르면, UIManager는 GoogleAuthService의 `signIn()` 메서드를 호출하여 로그인 프로세스를 시작합니다. GoogleAuthService는 Google의 인증 라이브러리를 통해 사용자에게 로그인 및 권한 동의 팝업창을 표시합니다. 사용자가 인증을 완료하면, Google은 토큰 정보를 GoogleAuthService에게 전달하고, GoogleAuthService는 등록된 콜백 함수를 통해 UIManager에게 로그인 성공 상태를 알립니다. 상태를 받은 UIManager는 다시 GoogleAuthService에게 사용자 이름을 요청하고(`getUserName()`), 반환된 이름으로 UI를 업데이트하여 최종적으로 로그인된 상태를 화면에 표시합니다.

로그아웃: 사용자가 "Logout" 버튼을 누르면, UIManager는 GoogleAuthService의 `signOut()` 메서드를 호출합니다. GoogleAuthService는 Google API 서버에 현재 토큰을 무효화하도록 요청하고, 성공적으로 처리되면 UIManager에게 로그아웃 상태를 알립니다. UIManager는 UI를 '로그아웃됨' 상태로 갱신합니다.



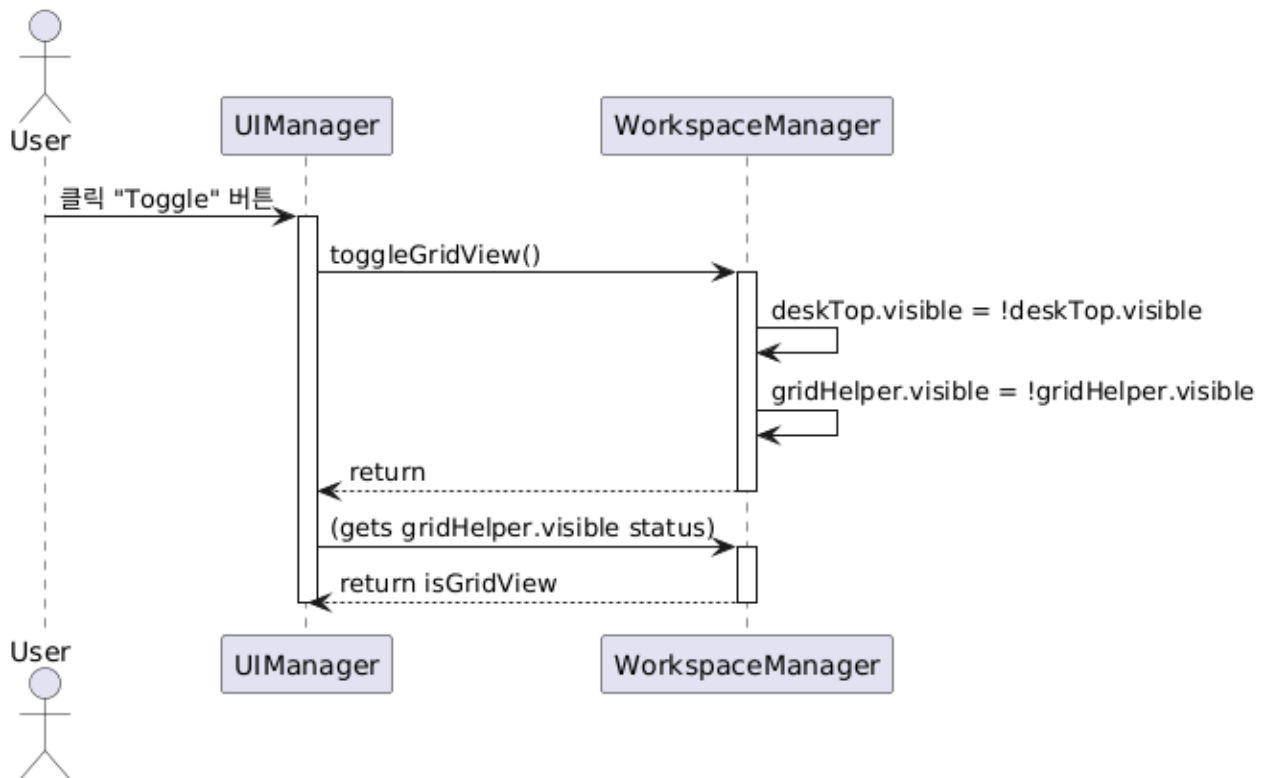
레이아웃 파일저장

사용자가 "Save Layout" 버튼을 클릭하면, UIManager는 먼저 사용자에게 저장할 파일명을 입력받는 프롬프트를 표시합니다. 파일명을 성공적으로 입력받으면, UIManager는 현재 작업 공간의 상태를 수집하기 위해 WorkspaceManager의 getCurrentLayoutData() 메서드를 호출하여 책상 정보를, 그리고 ObjectManager의 getObjectsData() 메서드를 호출하여 모든 오브젝트의 정보를 각각 요청합니다. 데이터를 모두 반환받은 UIManager는 이 정보들을 하나의 layoutData 객체로 결합합니다. 마지막으로, UIManager는 저장할 파일명과 완성된 layoutData를 GoogleDriveService의 saveFile() 메서드로 전달합니다. GoogleDriveService는 전달받은 데이터를 가지고 Google Drive API 서버에 실제 파일 저장(생성 또는 덮어쓰기)을 요청하고, 그 처리 결과를 최종적으로 UIManager에게 반환하여 사용자에게 저장 성공 메시지를 표시합니다.

레이아웃 파일 불러오기 (Load Layout)

사용자가 " Load Layout " 버튼을 클릭하면, UIManager는 GoogleDriveService의 listFiles() 메서드를 호출하여 파일 목록 조회를 요청합니다. GoogleDriveService는 Google Drive API와 통신하여 앱 폴더에 있는 파일들의 목록을 가져와 UIManager에게 반환합니다. 이 목록을 전달받은 UIManager는 파일 선택 UI를 구성하여 사용자에게 보여줍니다. 사용자가 목록에서 특정 파일을 클릭하면, UIManager는 해당 파일의 ID를 GoogleDriveService에 전달하며 loadFileContent() 메서드를 호출합니다. GoogleDriveService는 API로부터 해당 파일의 전체 내용을 가져와 UIManager에게 반환합니다. 최종적으로 데이터를 받은 UIManager는 이 layoutData를 인자로 하여, WorkspaceManager의 applyLayout()을 호출해 책상 크기를 복원하고, ObjectManager의 applyLayout()을 호출해 모든 오브젝트를 재배치함으로써 3D 작업 공간을 불러온 상태로 완벽하게 복원한 뒤 파일 선택 창을 닫습니다.

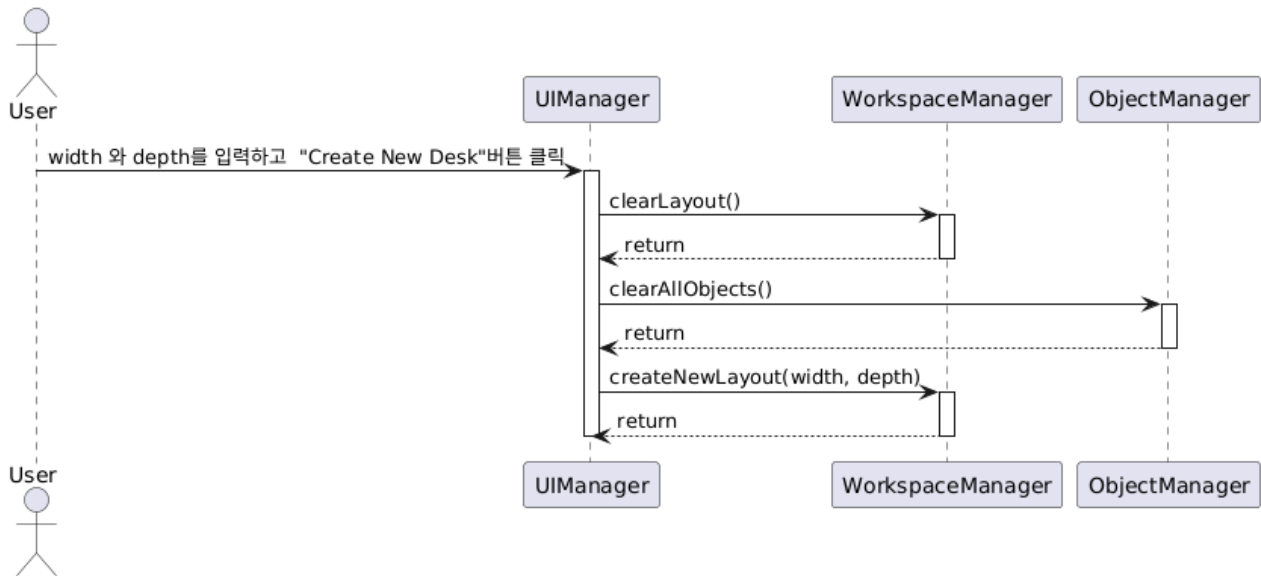
Toggle Desk/Grid View Sequence



책상/격자 뷰 전환 (Toggle Desk/Grid View)

사용자가 Desco 애플리케이션 내에서 "Toggle" 버튼을 클릭하면, UIManager는 WorkspaceManager의 toggleGridView() 메서드를 호출하여 보기 모드 전환을 요청합니다. 요청을 받은 WorkspaceManager는 자신이 직접 관리하고 있는 deskTop(책상 상판)과 gridHelper(격자) 객체의 visible(보이기) 속성을 현재 값의 정반대(true는 false로, false는 true로)로 변경하여 하나의 객체는 숨기고 다른 객체는 보이도록 만드는 핵심적인 전환 작업을 수행합니다. 이 작업이 완료된 후, 화면을 갱신합니다.

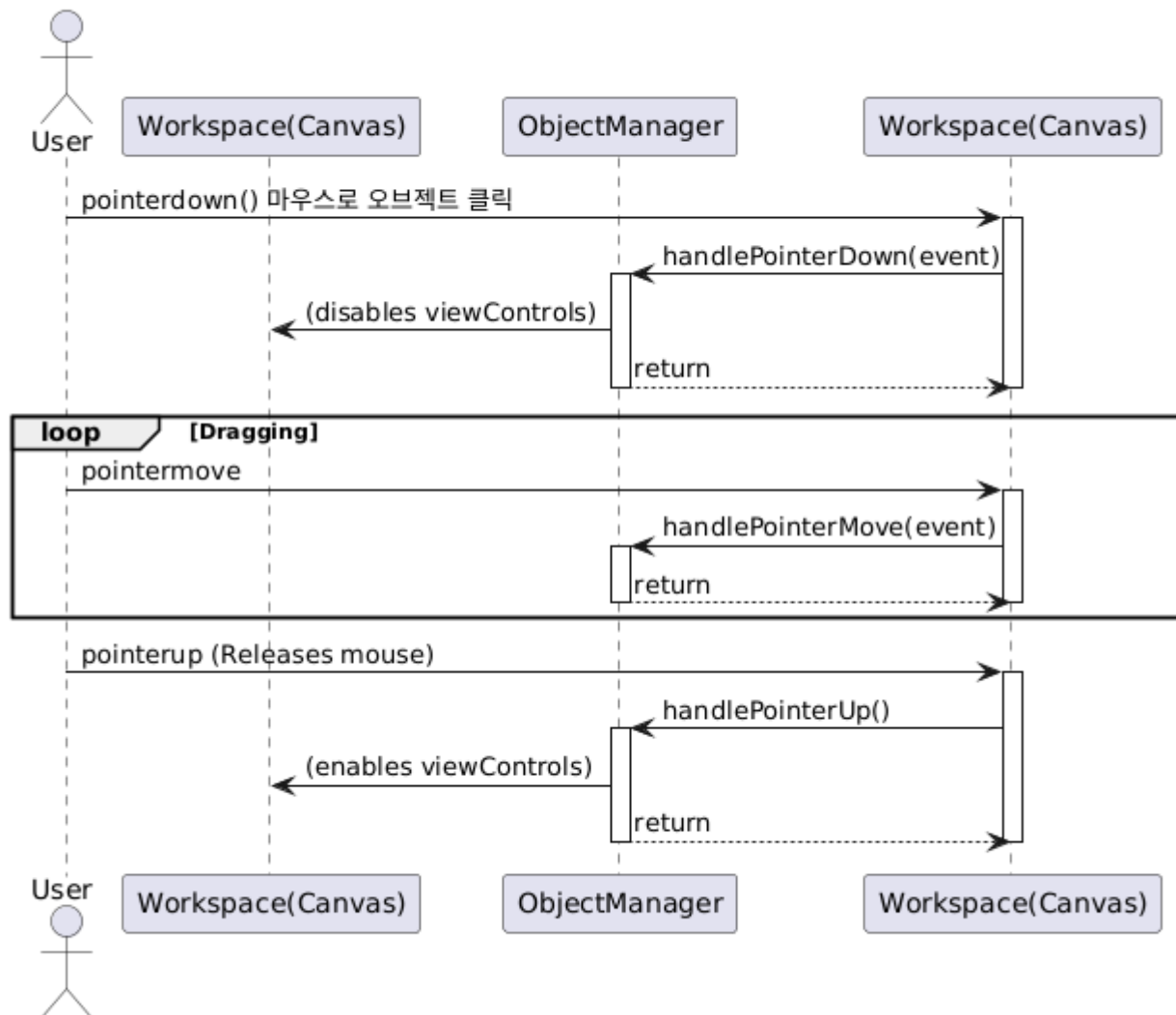
Create New Desk Sequence



새 데스크 생성 (Create New Desk)

사용자가 Desco 애플리케이션 내에서 새로운 책상의 가로, 세로 크기를 입력하고 "Create New Desk" 버튼을 누르면, UIManager는 먼저 WorkspaceManager의 clearLayout() 메서드를 호출하여 현재 씬의 책상을 제거하고, 이어서 ObjectManager의 clearAllObjects() 메서드를 호출하여 기존에 배치된 모든 오브젝트를 제거함으로써 작업 공간을 초기화합니다. 작업 공간이 깨끗해지면, UIManager는 사용자가 입력한 새로운 크기 값을 인자로 하여 WorkspaceManager의 createNewLayout() 메서드를 호출하고, 이를 통해 WorkspaceManager는 새로운 크기의 책상과 격자를 생성하여 씬에 배치합니다. 이후 새로운 데스크 생성 과정을 완료합니다.

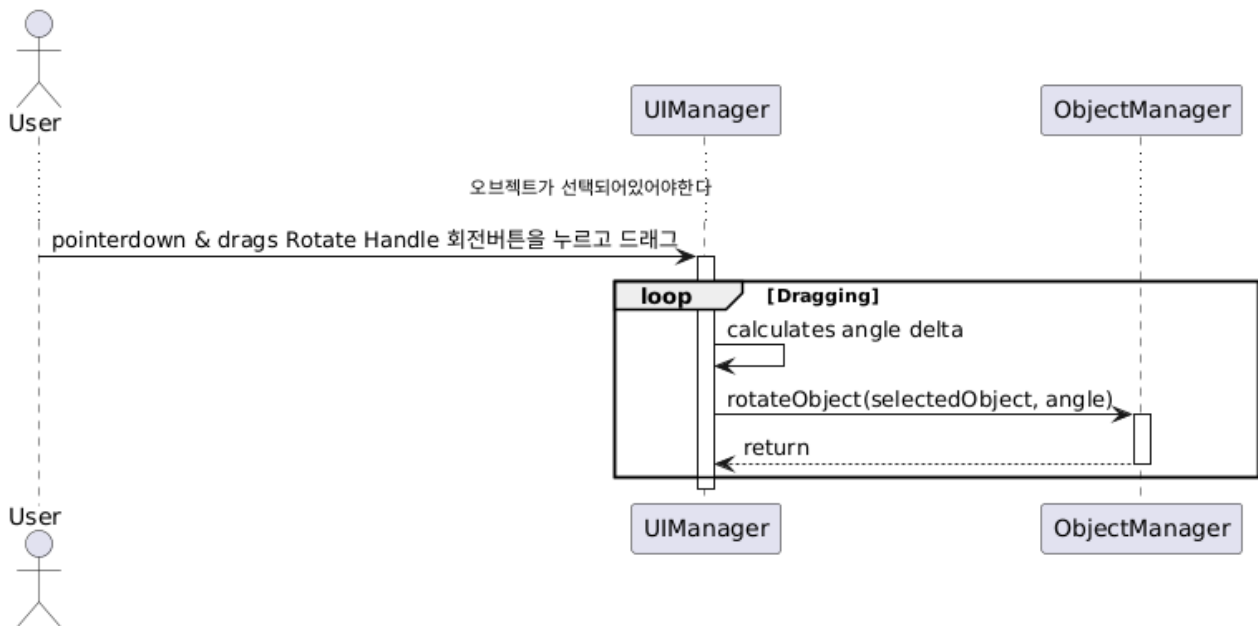
Move Object Sequence



오브젝트 이동 (Move Object)

사용자가 3D 캔버스 위의 특정 오브젝트를 마우스로 클릭하면(pointerdown), 이벤트를 감지한 WorkspaceManager는 ObjectManager의 handlePointerDown() 메서드를 호출합니다. ObjectManager는 오브젝트를 선택하고, 드래그 중 뷰가 회전하지 않도록 WorkspaceManager의 카메라 컨트롤을 비활성화합니다. 사용자가 마우스를 누른 채 움직이면(pointermove), WorkspaceManager는 계속해서 ObjectManager의 handlePointerMove()를 호출하고, ObjectManager는 마우스 위치에 따라 선택된 오브젝트의 위치를 실시간으로 업데이트합니다. 사용자가 마우스 버튼에서 손을 떼면(pointerup), WorkspaceManager는 ObjectManager의 handlePointerUp()을 호출하여 오브젝트 선택을 해제하고 카메라 컨트롤을 다시 활성화합니다.

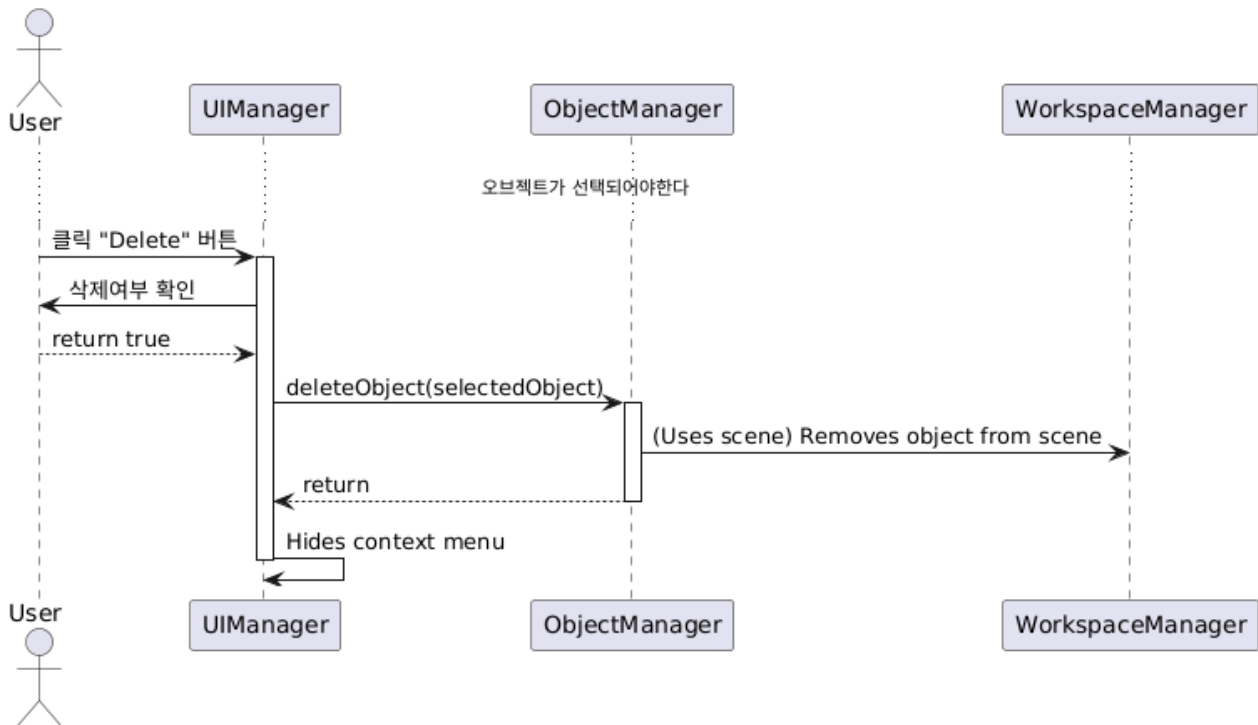
Rotate Object Sequence



오브젝트 회전 (Rotate Object)

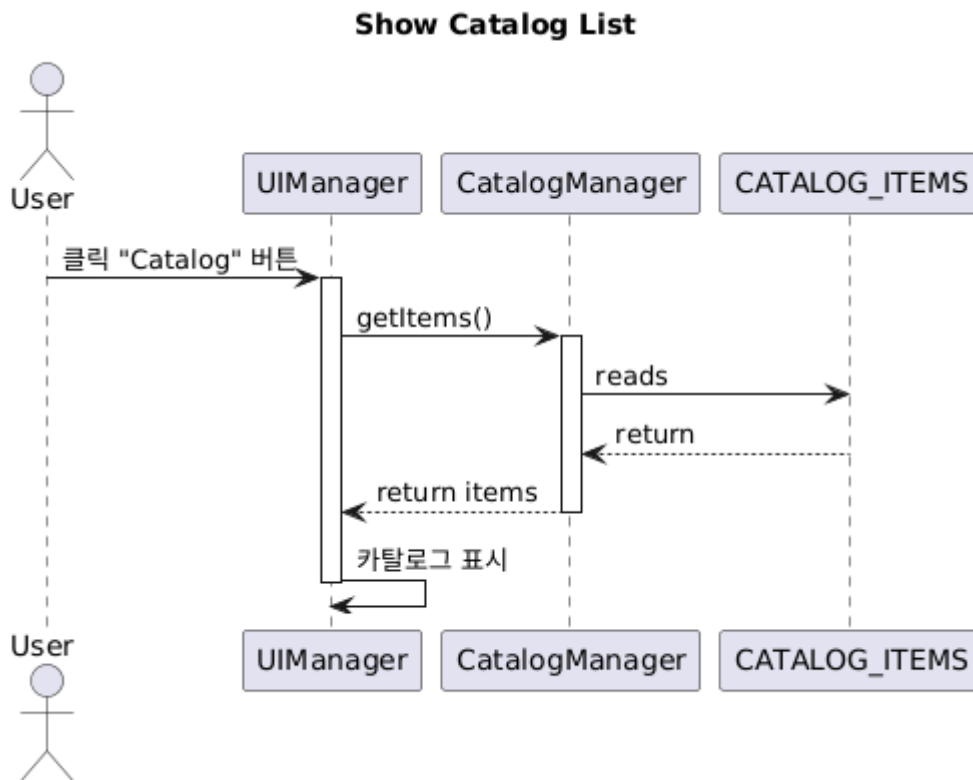
사용자가 특정 오브젝트를 선택하면, UIManager는 해당 오브젝트 주변에 회전 핸들 UI를 표시합니다. 사용자가 이 회전 핸들을 클릭하고 드래그하면, UIManager는 마우스의 수평 움직임 변화량을 감지하여 회전 각도로 계산합니다. UIManager는 이 계산된 각도 값을 인자로 하여 ObjectManager의 rotateObject() 메서드를 지속적으로 호출합니다. 요청을 받은 ObjectManager는 선택된 오브젝트의 회전 속성(rotation)을 업데이트하여, 사용자가 드래그하는 동안 오브젝트가 실시간으로 회전하도록 만듭니다.

Delete Object Sequence



오브젝트 삭제 (Delete Object)

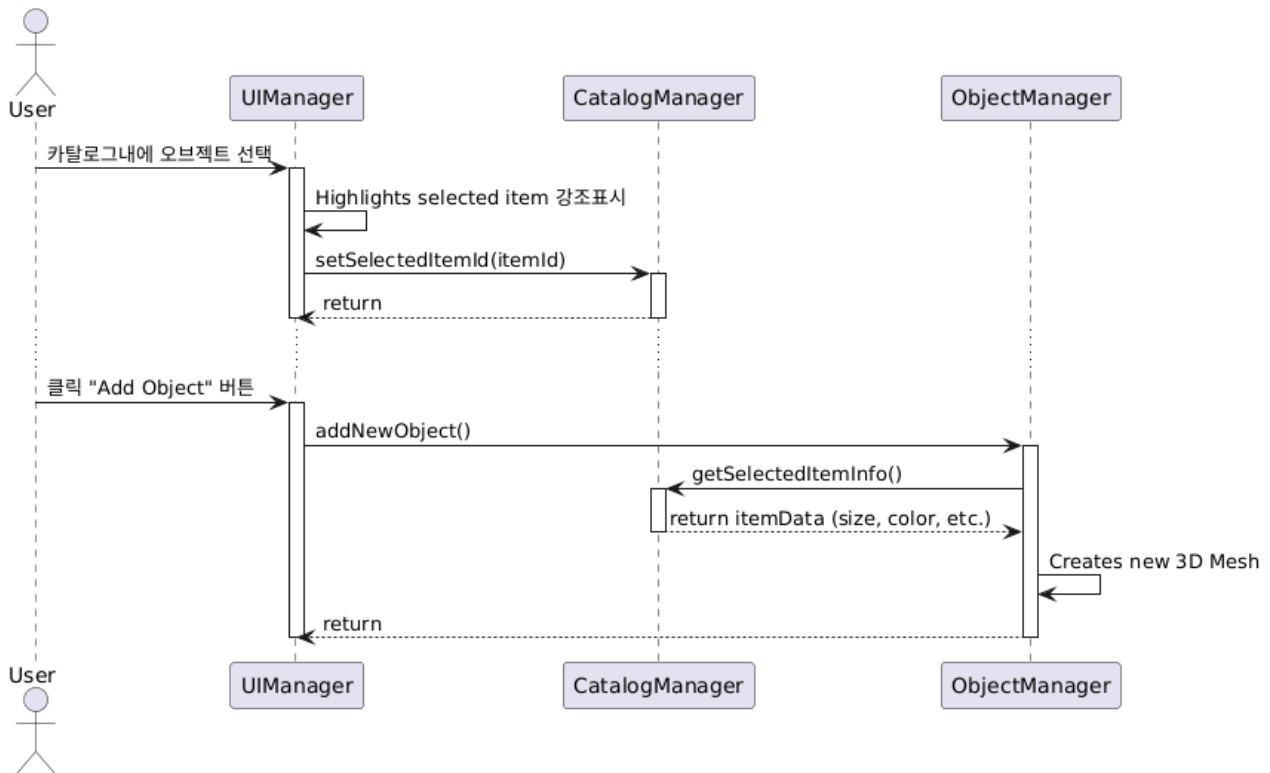
사용자가 특정 오브젝트를 선택하면, UIManager는 해당 오브젝트 주변에 삭제 버튼이 포함된 컨텍스트 메뉴를 표시합니다. 사용자가 이 삭제 버튼을 클릭하면, UIManager는 먼저 사용자에게 삭제 여부를 재확인하는 창을 띄웁니다. 사용자가 확인을 누르면, UIManager는 ObjectManager의 deleteObject() 메서드를 호출하면서 현재 선택된 오브젝트를 인자로 전달합니다. ObjectManager는 전달받은 오브젝트를 3D 씬(Scene)과 내부 관리 목록(placedObjects 배열)에서 모두 제거하고, 마지막으로 UIManager는 오브젝트가 사라졌으므로 컨텍스트 메뉴 UI를 화면에서 숨깁니다.



카탈로그 목록 표시 (Render Catalog)

사용자가 Desco 애플리케이션 내에서 "Catalog" 버튼을 클릭하면, UIManager는 이 이벤트를 감지하고 카탈로그 UI를 화면에 표시하는 프로세스를 시작합니다. 먼저, UIManager는 CatalogManager의 `getItems()` 메서드를 호출하여 추가 가능한 모든 아이템의 목록을 요청합니다. CatalogManager는 미리 정의된 데이터 소스인 CATALOG_ITEMS 배열을 읽어 UIManager에게 반환합니다. 이 아이템 데이터 배열을 전달받은 UIManager는 배열의 각 요소를 바탕으로 카탈로그 패널을 화면에 표시하여 사용자에게 선택 가능한 아이템 목록을 보여줍니다.

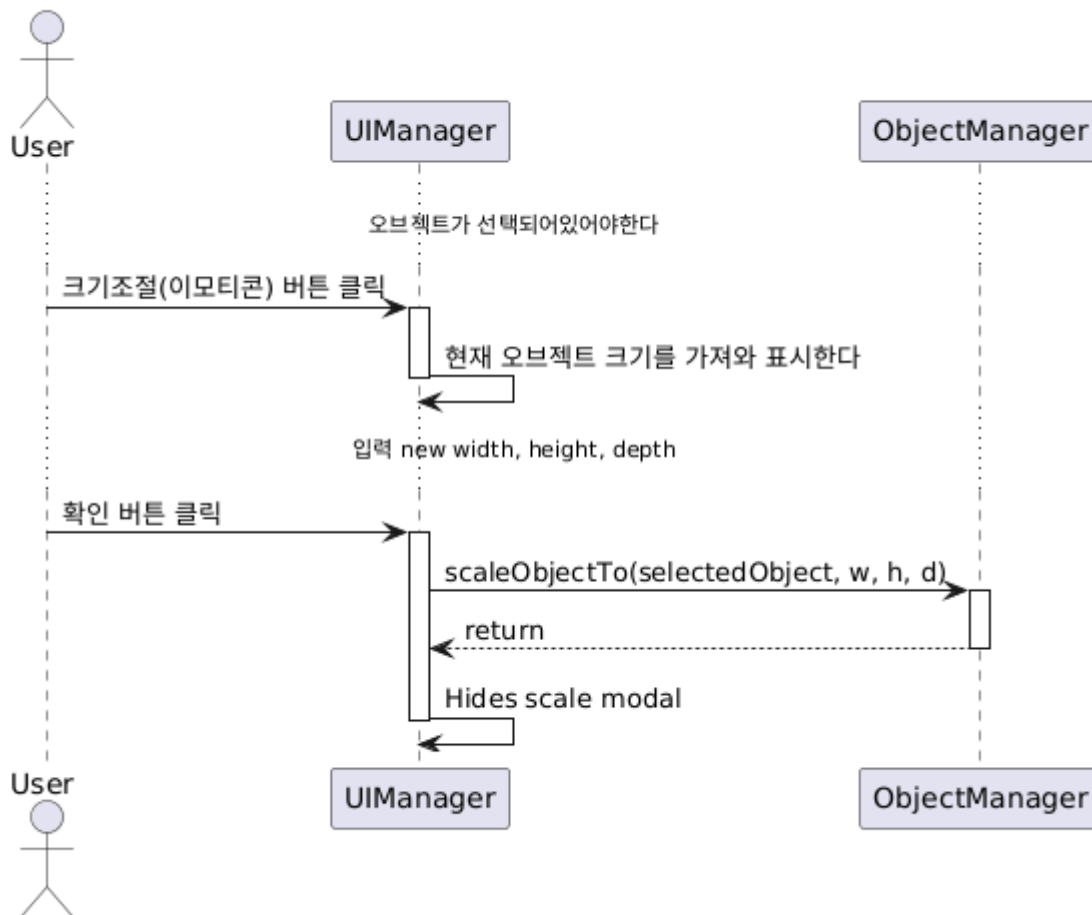
Add Object from Catalog



카탈로그에서 선택하여 오브젝트 추가

사용자가 카탈로그 UI에서 특정 아이템(예: 모니터)을 클릭하면, UIManager는 먼저 해당 UI 아이템을 시각적으로 강조 표시하고, 선택된 아이템의 고유 ID를 CatalogManager의 setSelectedItemId() 메서드에 전달하여 현재 사용자의 선택 상태를 기록합니다. 그 후 사용자가 "Add Object" 버튼을 클릭하면, UIManager는 ObjectManager의 addNewObject() 메서드를 호출하여 오브젝트 추가 프로세스를 시작합니다. addNewObject() 요청을 받은 ObjectManager는 CatalogManager의 getSelectedItemInfo() 메서드를 호출하여, 현재 선택된 아이템의 상세 데이터(크기, 색상 등)를 요청합니다. 이 데이터를 반환받은 ObjectManager는 이를 바탕으로 새로운 3D 객체(Mesh)를 생성하여 작업 공간의 중앙에 배치함으로써, 사용자가 선택한 아이템을 3D 씬에 최종적으로 추가합니다.

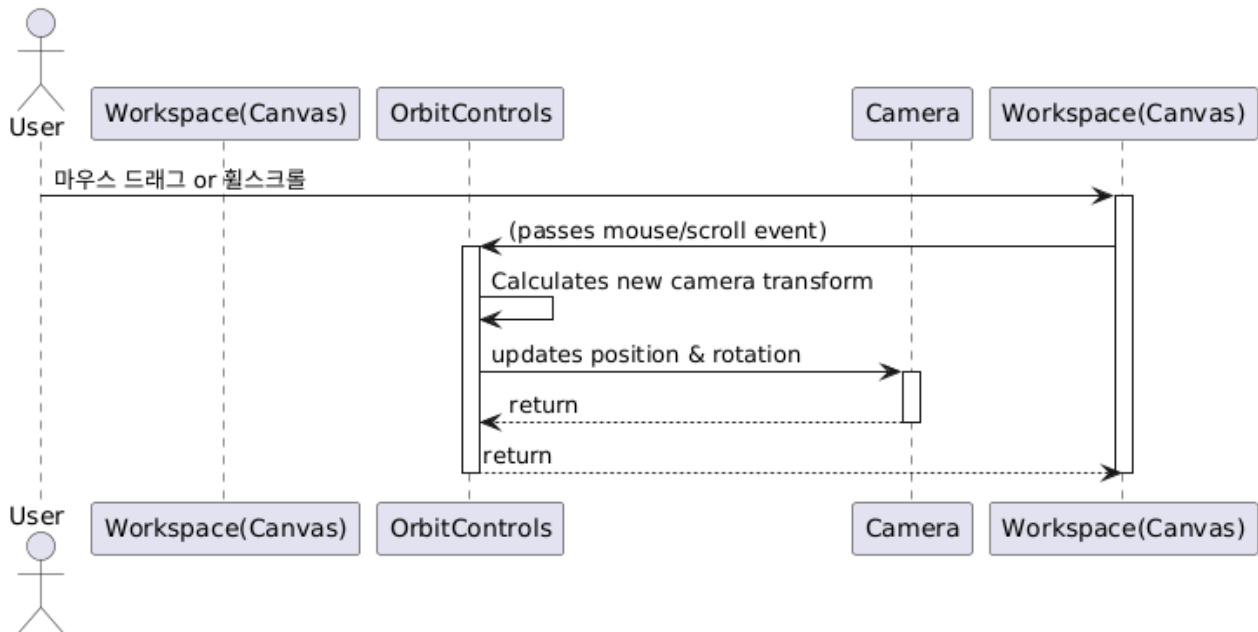
Scale Object Sequence



오브젝트 크기 조절 (Scale Object)

사용자가 선택된 오브젝트의 컨텍스트 메뉴에서 '크기 조절' 버튼을 클릭하면, UIManager는 먼저 현재 오브젝트의 크기 정보를 바탕으로 가로, 세로, 깊이 입력창의 초기값을 설정한 뒤 크기 조절 창을 화면에 표시합니다. 사용자가 이 창에 새로운 크기 값을 입력하고 "Apply" 버튼을 누르면, UIManager는 입력된 값들을 인자로 하여 ObjectManager의 `scaleObjectTo()` 메서드를 호출합니다. ObjectManager는 전달 받은 값으로 해당 오브젝트의 스케일(Scale) 속성을 재계산하여 3D 모델의 크기를 변경하고, 작업이 완료되면 UIManager는 크기 조절 창을 닫습니다.

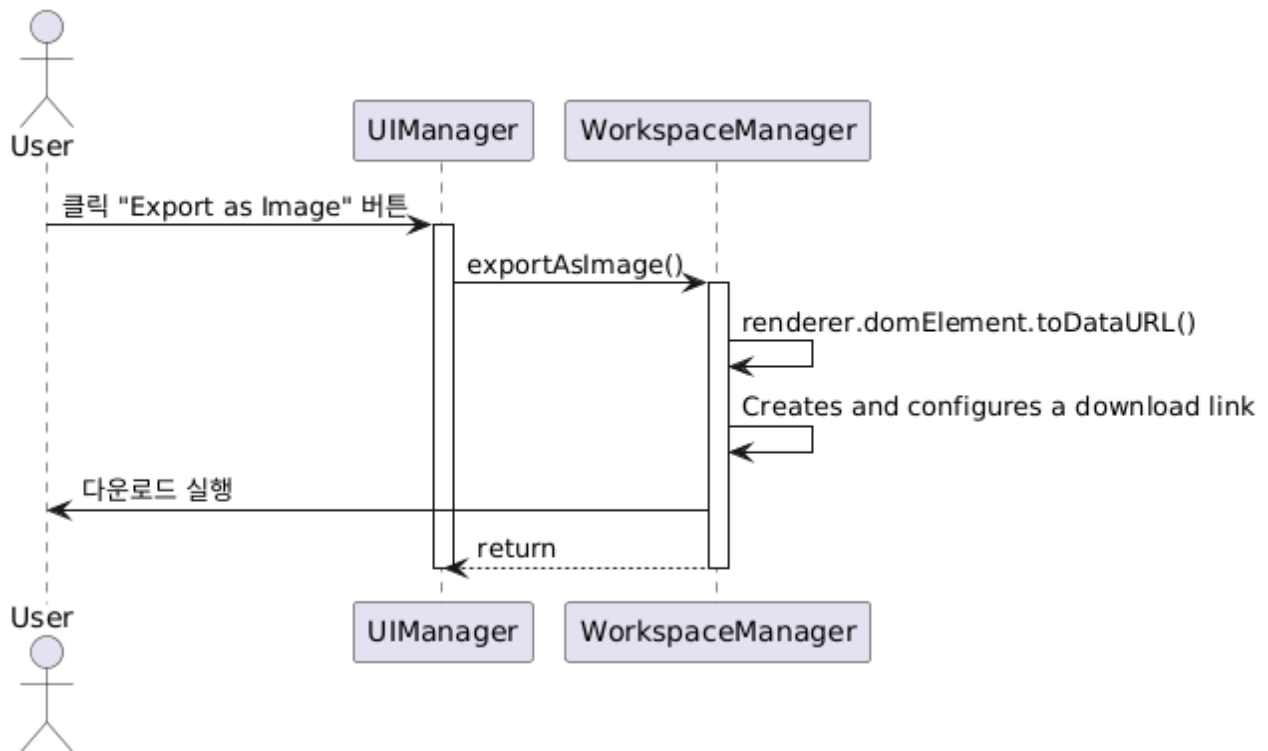
Camera View Control Sequence



카메라 뷰 조절 (View Control)

사용자가 3D 캔버스의 빈 공간을 마우스로 드래그하거나 스크롤 휠을 사용하면, 이 입력 이벤트는 WorkspaceManager에 의해 감지됩니다. 이 이벤트는 OrbitControls 라이브러리에 의해 직접 처리됩니다. OrbitControls는 마우스의 움직임이나 휠 스크롤 값을 바탕으로, 카메라가 어떻게 회전하고 확대/축소되어야 하는지에 대한 복잡한 수학적 계산을 수행합니다. 계산이 완료되면, OrbitControls는 WorkspaceManager가 관리하는 Camera 객체의 position(위치) 및 rotation(회전) 속성을 직접 업데이트합니다. 이 모든 변경 사항은 WorkspaceManager의 애니메이션 루프(_animate) 내에서 매 프레임마다 최종적으로 화면에 렌더링되어, 사용자는 부드러운 뷰 조작을 경험하게 됩니다.

Export as Image Sequence



이미지 내보내기 (Export as Image)

사용자가 Desco 애플리케이션 내에서 "Export as Image" 버튼을 클릭하면, UIManager는 이벤트를 감지하고 WorkspaceManager의 `exportAsImage()` 메서드를 호출하여 이미지 내보내기 프로세스를 시작합니다. 요청을 받은 WorkspaceManager는 자신이 관리하는 renderer를 사용하여, 현재 3D 씬이 그려진 캔버스(canvas)의 내용을 이미지 데이터 URL로 변환합니다. 그 후, WorkspaceManager는 이 데이터 URL을 href 속성으로 가지고, 파일명이 `desco_layout.png`로 설정된 임시 `<a>`(링크) 태그를 동적으로 생성합니다. 마지막으로, 생성된 이 링크로 브라우저의 파일 다운로드 기능을 실행시키고, 사용자는 현재 보고 있는 화면을 PNG 이미지 파일로 자신의 컴퓨터에 저장하게 됩니다.

5. Implementation requirements

- Describe operating environments to implement your system.
- 12pt, 160%.

- OS : Microsoft Window 10이상
- Network : 연결 필요

- System Dependencies: Google 계정 필요

6. Glossary

- Specifically describe all of the terms used in this documents.
- 12pt, 160%.

- Desco(데스코) : 해당 책상 꾸미기 어플리케이션 이름
- 작업공간 : 사용자가 책상을 배치하고 오브젝트를 꾸미는 주요 3D 인터페이스 영역
- Desk : 사용자가 작업 공간 내에 생성하고 오브젝트를 배치하는 기준 가구 모델
- Object : 책상 위에 배치할 수 있는 모든 가상의 아이템
- Sequence Diagram : 객체 간의 동적 상호작용을 시간적 개념으로 모델링하여 나타낸 다이어그램
- StateMachine Diagram : 객체 LifeTime동안의 변화될 수 있는 모든 상태를 정의해 둔 다이어그램
- API (Application Programming Interface) : 응용 프로그램이나 소프트웨어 구성 요소들이 서로 상호작용하고 특정 기능이나 데이터에 접근할 수 있도록 미리 정의해 둔 규칙, 함수, 프로토콜 등의 집합으로 이루어진 인터페이스. 내부 구현의 복잡성을 몰라도 해당 기능을 활용할 수 있게 해줌

7. References

- Describe all of your references (book, paper, technical report etc).
- 12pt, 160%.

-강의자료 : Structural Modeling II, Behavior Modeling I, II

-참조자료 : [Design] Examples