Term Project Report (Computer Graphics, Spring, 2025)

**Title: Room customizing simulator**

학번: 2021147528  이름: 신지환

학번: 2023148074  이름: 임재우

1. **요약**

Room customizing simulator은 제공된 가구를 활용해 방을 꾸밀 수 있는 인테리어 시뮬레이션입니다. 사용자는 마우스를 움직이며 방을 둘러볼 수 있습니다. 우측 리스트에 제공되는 가구를 드래그해 방에 배치하여 방을 꾸밀 수 있습니다. 벽을 클릭해 벽에도 가구를 배치할 수 있습니다.

실내, 전시회이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

1. **사용법**

기본적으로 사용자에게 8x8의 방이 주어집니다. Orbitcontrol을 활용해, 마우스로 방을 둘러볼 수 있습니다. 사용자는 우측의 리스트에 제공되는 가구들을 드래그해 방에 배치하여 방을 꾸밀 수 있습니다. 각 가구는 일정 크기의 타일을 차지하며, 가구가 충돌할 시 Ground Highlight가 빨갛게 표시되며 배치를 금지합니다. 리스트 하단의 빨간 버튼을 통해 방을 초기화할 수 있습니다.

배치된 가구를 클릭할 경우, Edit Mode로 진입합니다. 가구의 크기를 보여주는 Wireframe이 나타나 기존 배치 상태를 표시합니다. M키를 눌러 가구를 이동할 수 있습니다. R키를 눌러 가구를 회전시킬 수 있습니다. Delete키를 눌러 가구를 삭제할 수 있습니다.

벽을 클릭해 Wall view로 진입할 수 있습니다. 벽은 상단 8x4의 Grid를 제공합니다. 우측의 리스트는 벽에 설치 가능한 가구로 전환되며, 일반 view와 동일하게 가구를 드래그해 벽에 배치할 수 있습니다. 이때 Orbitcontrol은 비활성화됩니다. ESC키를 눌러 다시 일반 view로 돌아올 수 있습니다.

1. **기능 구현 표**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 기능 | 구현위치 | 비고 |
| 1 | Document style | index.html (line - ) |  |
| 2 | Furniture list | index.html (line - ) |  |
| 3 | Edit mode index | index.html (line - ) |  |
| 4 | Program Initialization | app.js (line - ) | init() |
| 5 | Light Setup | app.js (line - ) | setupLighting() |
| 6 | Ground highlight Setup | app.js (line - ) | createGroundHighlight() |
| 7 | Wall highlight Setup | app.js (line - ) | createWallHighlight() |
| 8 | Floor grid Setup | app.js (line - ) | createFloorGrid() |
| 9 | Wall grid Setup | app.js (line - ) | createWallGrids() |
| 10 | Grid activation | app.js (line - ) | gridOnOff() |
| 11 | Snap wall coordinates to grid | app.js (line - ) | snapToWallGrid() |
| 12 | Snap floor coordinates to grid | app.js (line - ) | snapToGrid() |
| 13 | Get object size | app.js (line - ) | getObjectDimensions() |
| 14 | Check floor object collision | app.js (line - ) | checkCollision() |
| 15 | Check wall object collision | app.js (line - ) | checkWallCollision() |
| 16 | Load fbx models | app.js (line - ) | loadModels() |
| 17 | Create objects preview | app.js (line - ) | createModelPreview() |
| 18 | User move event | app.js (line - ) | onControlsStart() |
| 19 | User move event ended | app.js (line - ) | onControlsEnd() |
| 20 | Reset camera | app.js (line - ) | springBackCamera() |
| 21 | Event listeners | app.js (line - ) | setupEventListeners() |
| 22 | Drag and Drop | app.js (line - ) | setupDragAndDrop() |
| 23 | Preview for dragging | app.js (line - ) | startDragPreview() |
| 24 | Update dragging preview | app.js (line - ) | updateDragPreview() |
| 25 | Delete dragging preview | app.js (line - ) | endDragPreview() |
| 26 | Add object | app.js (line - ) | addObjectToScene() |
| 27 | Object placement animation | app.js (line - ) | animateObjectPlacement() |
| 28 | Object placement | app.js (line - ) | addWallObjectToScene() |
| 29 | Mouse click event | app.js (line - ) | onMouseClick() |
| 30 | Select object | app.js (line - ) | selectObject() |
| 31 | Object wireframe | app.js (line - ) | animateSelectionBox() |
| 32 | Unselect object | app.js (line - ) | deselectObject() |
| 33 | Mouse move event | app.js (line - ) | onMouseMove() |
| 34 | Key event | app.js (line - ) | onKeyDown() |
| 35 | Delete object | app.js (line - ) | deleteSelectedObject() |
| 36 | Window resize | app.js (line - ) | onWindowResize() |
| 37 | animate | app.js (line - ) | animate() |
| 38 | Room Reset | app.js (line - ) | clearRoom() |
| 39 | Rotate object | app.js (line - ) | rotateSelectedObject() |
| 40 | Object wireframe | app.js (line - ) | flashSelectionBox() |
| 41 | Enter move mode | app.js (line - ) | enterMoveMode() |
| 42 | Exit move mode | app.js (line - ) | exitMoveMode() |
| 43 | Update preview while moving | app.js (line - ) | updateMovePreview() |
| 44 | Move confirm | app.js (line - ) | confirmMove() |
| 45 | Update ground highlight | app.js (line - ) | updateGroundHighlightSize() |
| 46 | Create Wall Click box | app.js (line - ) | createWallClickBoxes() |
| 47 | Wall view | app.js (line - ) | moveCameraToWall() |
| 48 | Return from Wall view | app.js (line - ) | resetCamera() |
| 49 |  | app.js (line - ) |  |

1. **기타 사용 기능**

주요 함수 설명:

1) createGroundHighlight()

* 바닥에 하이라이트 효과를 생성하는 함수이다.
* 바닥을 강조하는 초록색 반투명한 Highlight를 생성한다.
* 해당 타일에 이미 가구가 있을 경우 빨간색 HIghlight로 표시된다.

2) createFloorGrid()

* 바닥에 Grid를 생성하는 함수이다.
* 바닥 크기(8x8)에 맞는 Grid 메시를 생성한다.

3) snapToGrid(position)

* 객체의 위치를 Grid에 맞추는 함수이다.
* 입력된 위치를 가장 가까운 타일로 조정한다.
* Grid 간격은 1 단위로 설정된다.

4) checkCollision(object, position)

* 객체가 다른 객체나 벽과 충돌하는지 확인하는 함수이다.
* 입력된 위치에서 객체의 경계 상자를 계산한다.
* 다른 객체나 벽과 겹치는지 확인한다.

5) createModelPreview(type)

* 드래그 중인 객체의 미리보기를 생성하는 함수이다.
* 미리보기의 색상은 반투명한 파란색으로 설정된다.

6) addObjectToScene(object)

* 객체를 씬에 추가하는 함수이다.

7) selectObject(object)

* 객체를 선택하는 함수이다.
* 이전에 선택된 객체의 Highlight를 제거한다.
* 새로운 객체를 선택하고 Highlight를 추가한다.

8) rotateSelectedObject()

* 선택된 객체를 회전시키는 함수이다.
* 객체의 타입에 따라 다른 회전 로직을 적용한다.
* 회전 후 객체의 위치를 적절히 조정한다.
* 회전이 완료되면 선택 박스를 업데이트한다.

9) moveCameraToWall(wall)

* 카메라의 위치를 벽 앞으로 이동시킨다.

1. **기타**

Room customizing simulator은 방을 사용자가 원하는 대로 가구를 배치해 꾸미고 감상할 수 있는 프로그램이다. 이 프로젝트가 가지는 독창성은 다음과 같다.

1. 혁신적인 사용자 경험 설계

Room customizing simulator은 매우 기능적인 UI를 제공한다. js뿐만 아니라 html 파일도 활용해 사용자에게 필요한 설명을 UI로 표시하고, 우측에 상호작용 가능한 가구 리스트를 나열한다. 사용자는 마우스로 리스트의 어떤 가구든 클릭 & 드래그로 씬에 배치할 수 있다.

이 시뮬레이션은 이중 뷰 모드를 통해 언제든 클릭만으로 Room view와 Wall view를 오갈 수 있다. 이를 통해 사용자가 방의 모든 각도에서 인테리어를 세밀하게 설계할 수 있다.

또한 자연스럽고 부드러운 애니메이션과 기능적인 상호작용을 제공한다. 사용자는 마우스 컨트롤을 통해 자유롭게 방을 둘러볼 수 있다. 또한 카메라가 멈추고 일정 시간이 지나면 원래 자리로 복귀하는 기능 또한 구현되어 있다. Wall view로 전환할 때도 카메라가 자연스럽게 벽을 향해 이동하고 동시에 FOV가 조정되어, 벽에 딱 맞는 시야가 설정된다. 이 경우에는 OrbitControl과 Camera reset도 자동으로 비활성화된다.

이와 같이 Room customizing simulator가 탑재한 다중 모드와 스마트 카메라 시스템이 사용자에게 혁신적인 경험을 제공할 수 있다.

1. 지능형 배치 시스템

Room customizing simulator은 8x8의 Room Grid와 8x4의 Wall Grid를 제공한다. 사용자는 Room view와 Wall view를 오가며 일반 가구와 벽 가구를 드래그해 배치할 수 있다. 이때 사용자가 가구의 Preview를 볼 수 있는 기능이 구현되어 있다. 리스트의 아이템을 클릭해 씬으로 드래그할 때, 자동으로 Preview model이 생성되어 화면에 표시된다. 사용자는 가구의 Preview를 통해 유연하게 인테리어를 설계할 수 있다.

또한 가구를 배치할 때 충돌 계산과 시각적 피드백 또한 지원한다. 가구를 배치할 때 Preview model 아래에 초록색 Highlight가 생성되는데, 만약 해당 타일에 이미 가구가 있을 경우, 충돌이 발생하여 빨간색 Highlight가 표시된다. 이 경우 해당 타일에는 설치가 불가능하다.

또한 배치된 가구를 클릭해 Edit Mode에 들어갈 때, Wireframe이 나타나 해당 가구가 차지하는 공간을 표시한다. 이는 복잡한 형태의 object가 차지하는 공간을 보여준다. 또한 가구를 옮길 때 기존 위치에 남아 배치가 어떻게 변경되었는지를 파악하는 데 쓸 수 있다.

이 외에도 가구를 배치할 때의 애니메이션이나, 램프 등의 광원을 설치할 때 Pointlight도 함께 생성되어 사실적인 Lighting을 구현하는 등의 디테일도 구현되어 있다.

Room customizing simulator은 Three.js와 Blender, html을 종합적으로 활용해 개발한 인테리어 설계 시뮬레이터이다. 이 프로젝트는 사용자가 인테리어를 직접 설계하고 꾸미며 느끼는 재미와 즐거움뿐만 아니라, 실생활에도 사용할 수 있을 만큼 유용한 기능을 제공한다. Room customizing simulator는 매우 완성도 높은 배치 시스템과 활용도 높은 다양한 도구를 가지고 있다. 지속적인 개발을 통해 가구의 종류를 늘리고 방의 크기를 직접 변경할 수 있는 등의 기능들을 추가할수록, 실무에도 활용 가능한 수준으로 발전할 수 있는 가능성을 품고 있다.