과목 명: 기초컴퓨터그래픽스

담당 교수 명: 임인성

<4차 프로젝트>>

**서강대학교 컴퓨터학과**

**[20130956]**

**[장동욱]**

camera 번호 요약[viewport 위치]

camera6: 주 카메라, 건물을 하늘에서 바라보는 시점[우측, 기본값]

camera1~3: 상면도 측면도 정면도(순서 무관)[좌측 최하단부터 3개]

camera4: 동적 카메라[좌측 최상단, 기본값]

camera5: 호랑이 시점[우측, 기본 상태에서 ‘D’ 한 번눌러서]

**자기 채점 관련:**

주어진 요구사항을 모두 부합하였다고 생각하기에 스스로 모든 항목에 대해 만점을 주겠습니다. 추가 구현 사항은 조교님들의 공정한 심사를 기다리도록 하겠습니다. 감사합니다.

**구현 여부 관련**

모든 항목을 구현하였으므로, 별도의 구현 표시는 생략했습니다.

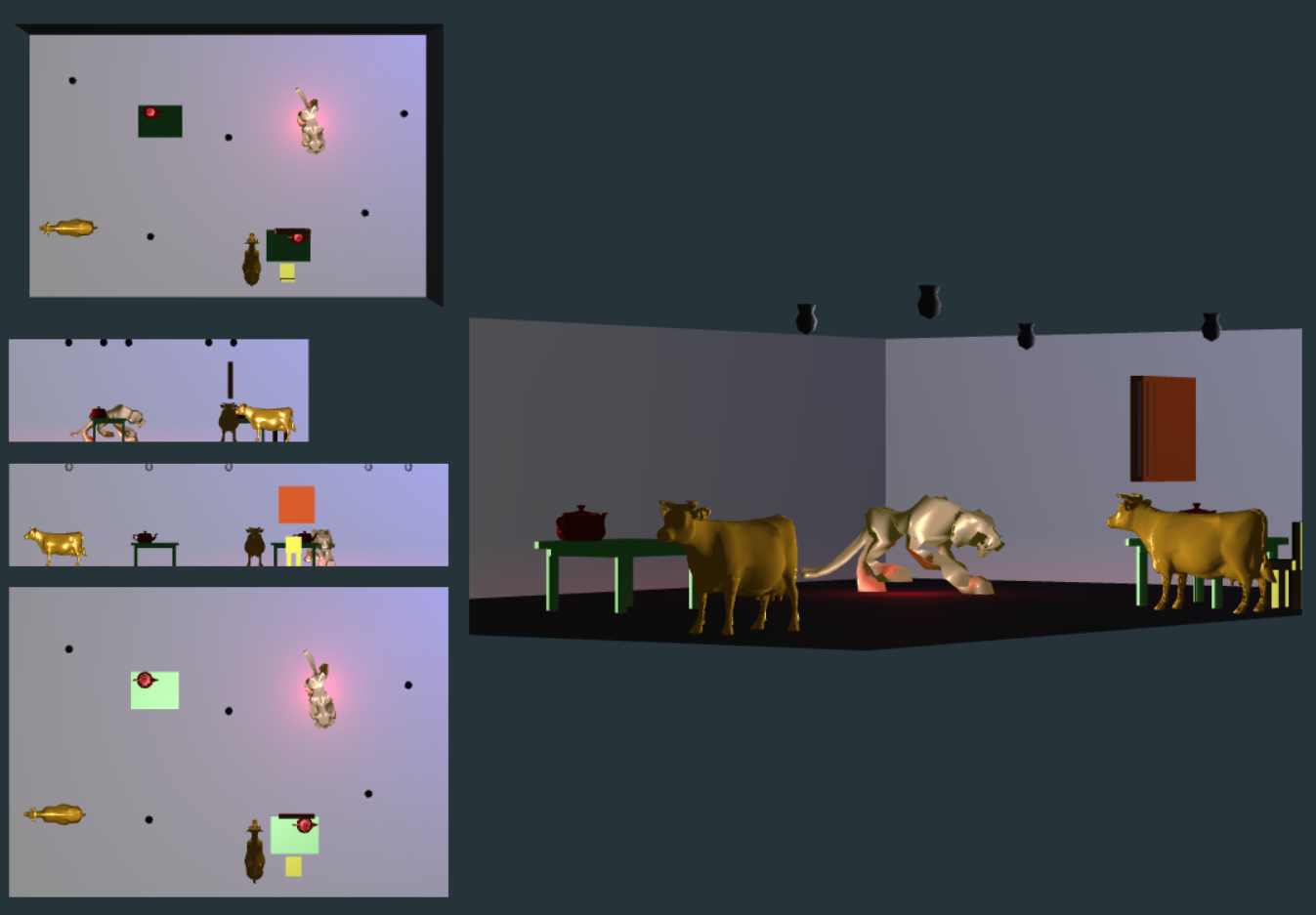
**키보드 입력 관련**

키보드 대소문자를 명확히 구분해주셔야 합니다.

# 다음과 같은 기본 요구사항을 만족시켜주는 프로그램일 작성하라. 쉐이딩 효과 이전까지의 구현은 자신이 HW2에서 작성한 프로그램을 사용하는 것으로 가정함.

## *프로그램이 수행되면 초기에는 기본적으로 Phong shading에 기반을 둔 쉐이딩 방법이 적용되어 있다.*

기본 화면에서 ‘w’키를 두 번 눌러 벽을 제거하여 아래와 같은 출력을 얻을 수 있습니다. 다시 ‘w’키를 누르면 벽이 생성됩니다.



## *사용자가 쉬프트 키와 오른쪽 버튼을 동시에 누를 경우, 누르고 있는 동안 가장 기본적인 Gouraud shading 방법에 기반을 둔 쉐이딩이 적용된다.*

주어진 입력 조합 외에 ‘S’키를 눌러 Gouraud shading을 적용할 수 있습니다. ‘S’키를 다시 누르면Phong shading으로 돌아옵니다. 두 shading 간의 차이가 분명하지 않을 경우, 지난 프로젝트에서 구현한 컨트롤을 사용하여 더 자세히 볼 수 있습니다. 윗 방향키를 눌러서 확대, 아래키를 눌러 축소, 드래그 앤 드롭으로 카메라 위치를 변경할 수 있습니다. 카메라 회전은 컨트롤 키를 누르고, 윗키, 아래키, 드래그 앤 드롭입니다. 혹은 키보드 ‘D’를 눌러서 호랑이 시선으로 전환합니다. 벽에 가로막히거나, 호랑이의 시야에 아무것도 없을 경우, 검은 화면이 나오는데, 전자의 경우 ‘w’를 눌러 벽을 없애거나, 후자의 경우, 조금만 기다리시면 물체가 나타납니다.



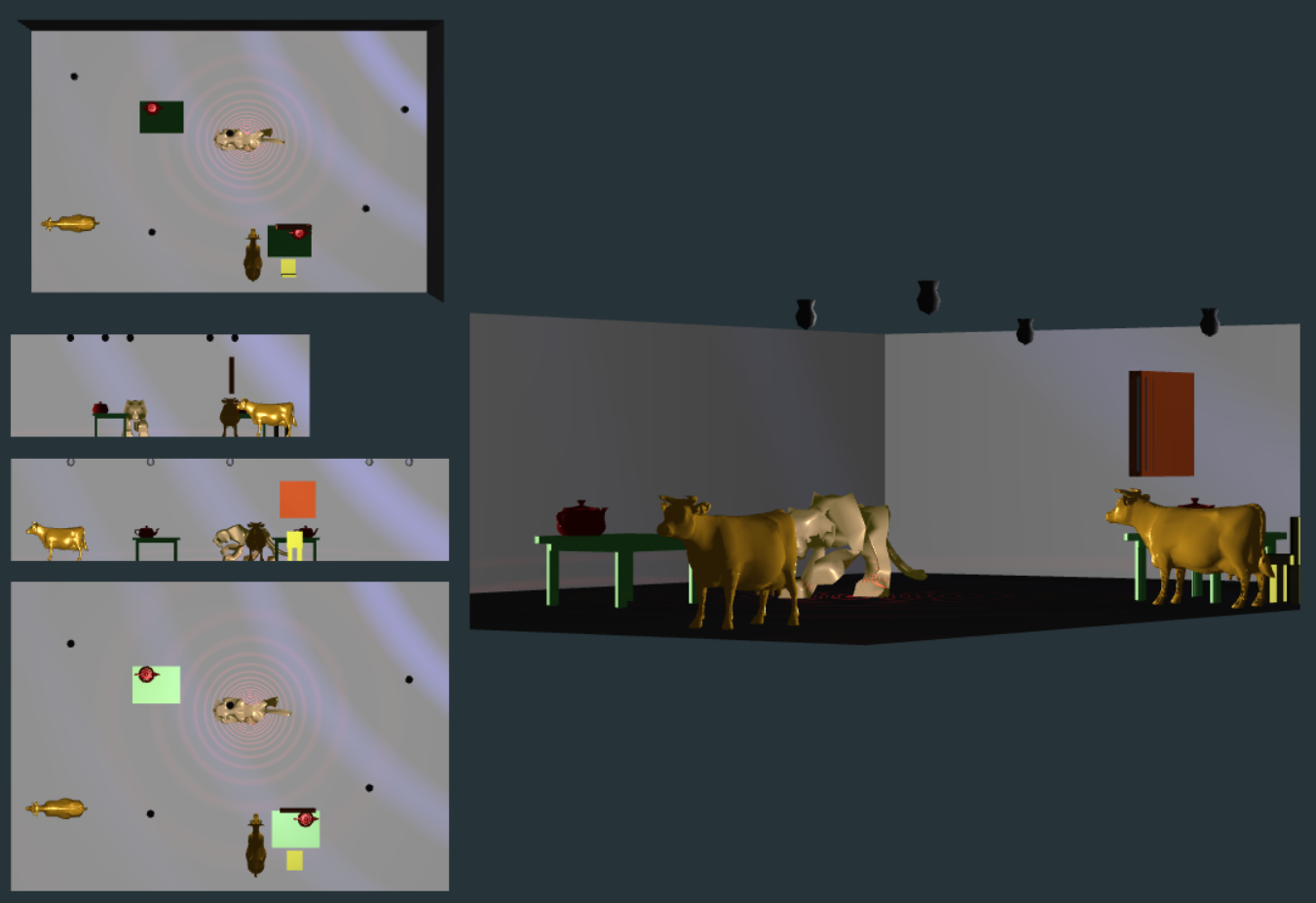
## *사용자가 쉬프트 키와 오른쪾 버튼에서 손을 떼면 다시 원래대로 Phong shading 방법이 적용된 상태로 돌아간다.*

주어진대로 구현 완료헀습니다.

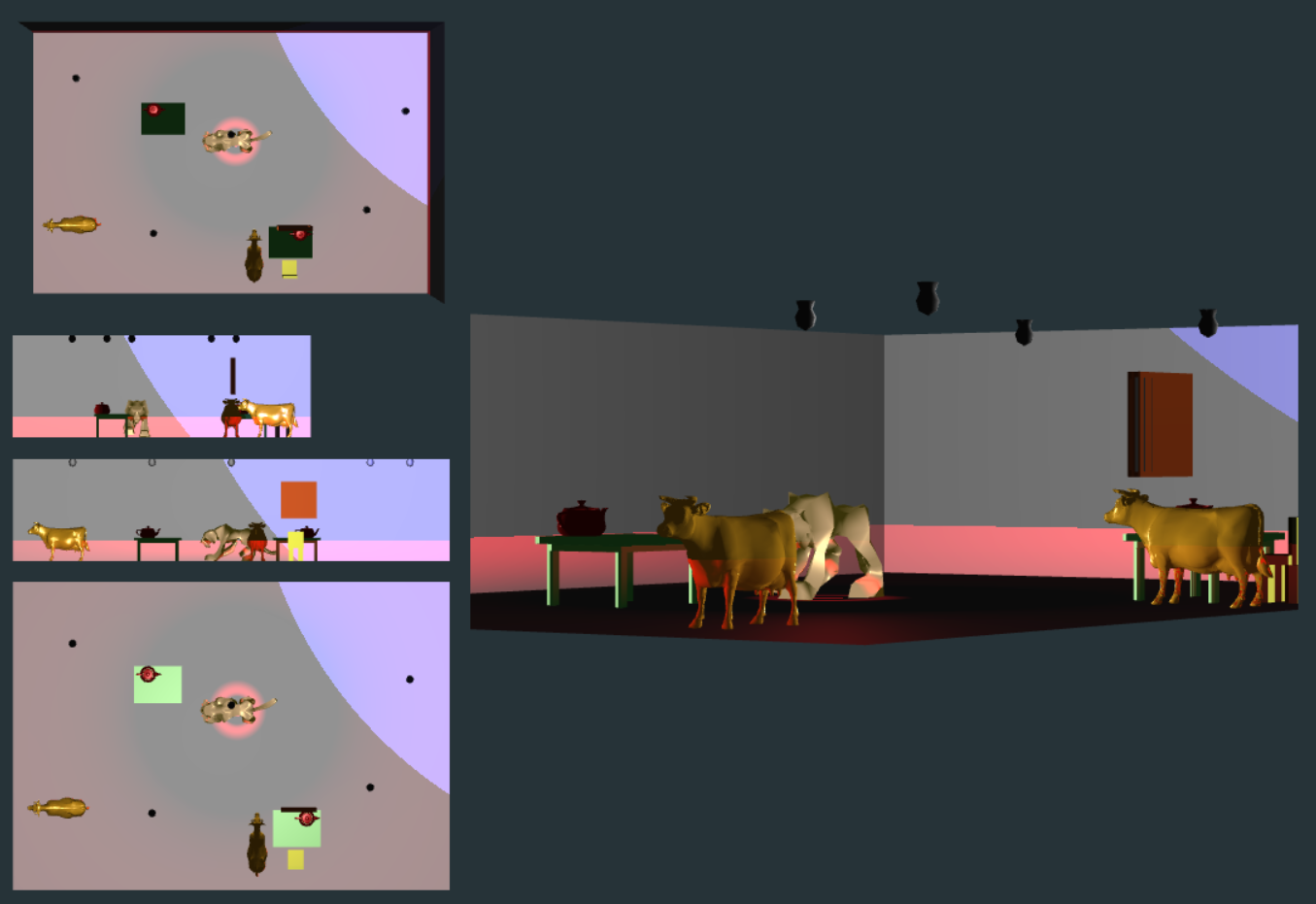
# Phong shading 모드에서 자신만의 재미있는 광원 관련 쉐이딩 효과를 생성하라.(이 효과는 Gouraud shading 모드에서는 생성할 필요가 없음). 자신이 구현한 각각의 기능을 키를 사용하여 토글할 수 있어야 하며, 필요시 쉐이딩 인자를 조절할 수 있어야 한다.(보고서에 해당 기능과 키 조작법에 대한 설명을 정확히 기술할 것)

## *수업 시간에 보여준 스폿 광원에 대한 블라인드 효과를 생성하라(다음 장 그림 참조). 이 기능 구현 시 최소환 광원을 가리키는 블라인드의 촘촘함을 “자신만의 방법으로” 키조작을 통하여 조절할 수 있어야 한다.*

‘b’키를 눌러 블라인드 효과를 활성화합니다. 그 상태에서 ‘b’키를 두 번 더 누르면 원래대로 돌아옵니다. ‘u’키를 눌러 더욱 촘촘한 효과를 ‘i’키를 눌러 성긴 효과를 낼 수 있습니다. 조절키는 코사인 함수의 frequency를 조절하는 세타의 계수를 각각 5만큼 증가, 감소시킵니다. 초기값은 90입니다.



## *마우스 위의 광원 관련 쉐이딩 효과 외에 그 이상의 복잡도를 가지는 조명과 관련한 쉐이딩 효과를 생성하라. (여기서 조명 효과는 광원에서 빛이 발광하는 방식이나 물체 표면에서의 반사 방식에 대한 변형 등을 모두 포함함) 점소는 복잡도와 재미에 따라 0점에서 20점까지 부여함.*



기본화면에서 ‘b’를 두 번, 혹은 2.1에서 ‘b’를 한 번 눌러서 새로운 조명 효과를 적용합니다. time\_scene 함수 내부에 정의되고 함수가 호출될 때마다 1씩 증가되어 최대 10까지 증가하고 다시 0부터 1씩 증가하는 변수 time을 선언하고, 이를 Phong shader에 넘겨주어 주기 10 동안 변화하는 조명 효과를 적용했습니다.

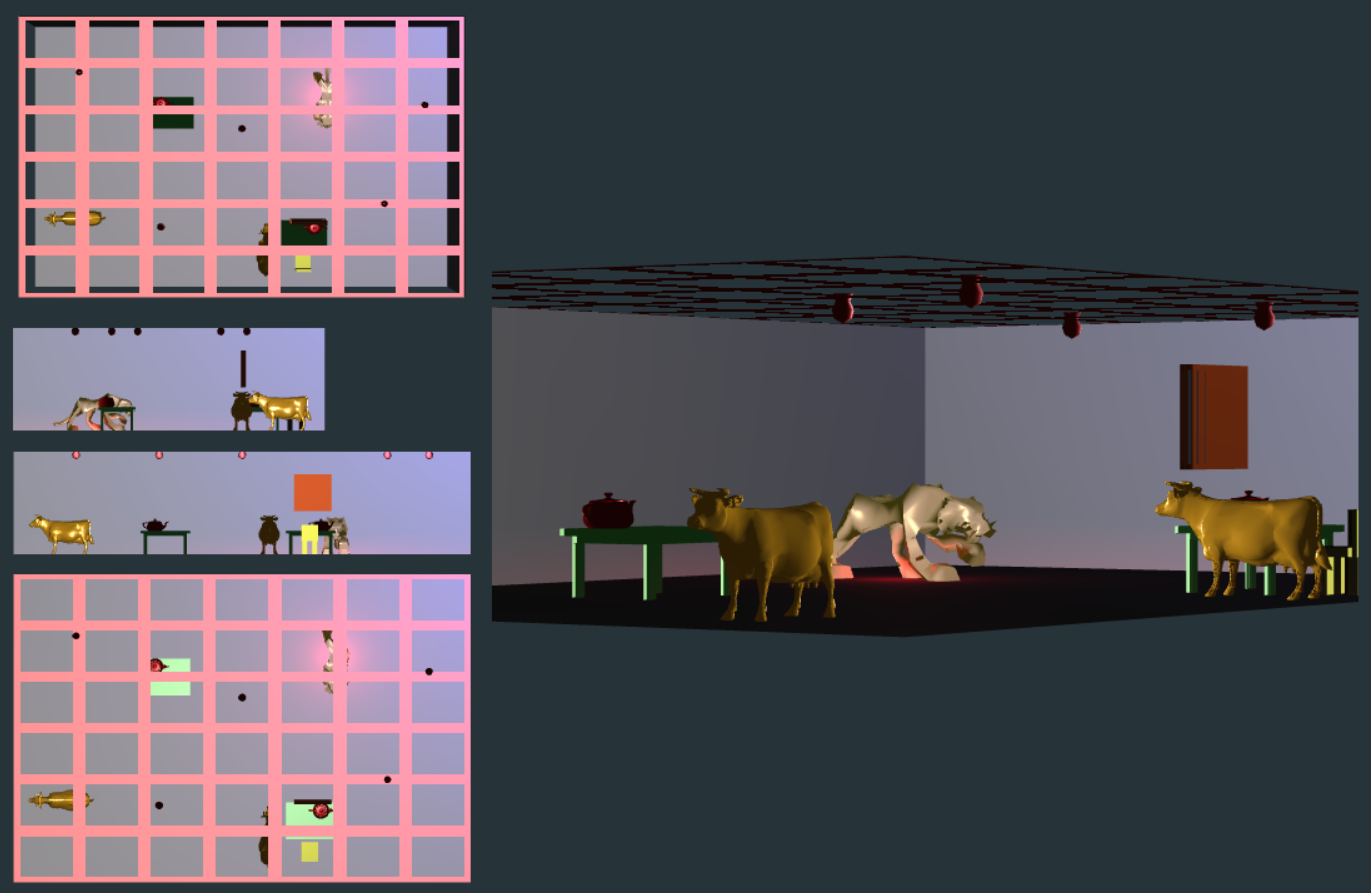
# Phong shading 모드에서 자신만의 재미있는 기하 물체 관련 쉐이딩 효과를 생성하라.(이 효과는 Gouraud shading 모드에서는 생성할 필요가 없음). 자신이 구현한 각각의 기능을 “자신만의 방법으로” 키를 사용하여 토글할 수 있어야 하며, 필요시 쉐이딩 인자를 조절할 수 있어야 한다.(보고서에 해당 기능과 키 조작법에 대한 설명을 정확히 기술할 것)

## *수업 시간에 보여준 스크린 쉐이더 효과를 생성하라(그램 참조). 이 기능 구현 시 최소한 스크린의 밀도를 키 조작을 통하여 조절할 수 있어야 한다.*

‘s’키를 눌러 스크린 효과를 활성화합니다. 현재는 별도의 쉐이더 효과는 없는 상태입니다.



‘e’키를 한 번눌러 스크린의 밀도를 조절할 수 있는 상태로 전환합니다. 이 상태에서는 ‘t’키를 눌러서 세로방향 밀도를 증가, ‘y’키를 눌러서 감소시킬 수 있습니다. 마찬가지로 ‘g’키를 눌러서 가로 방향 밀도를 증가, ‘h’키로 감소시킬 수 있습니다. 두 밀도 모두 최솟값이 1으로 설정되어 스크린 상 구멍의 최솟값은 1입니다.

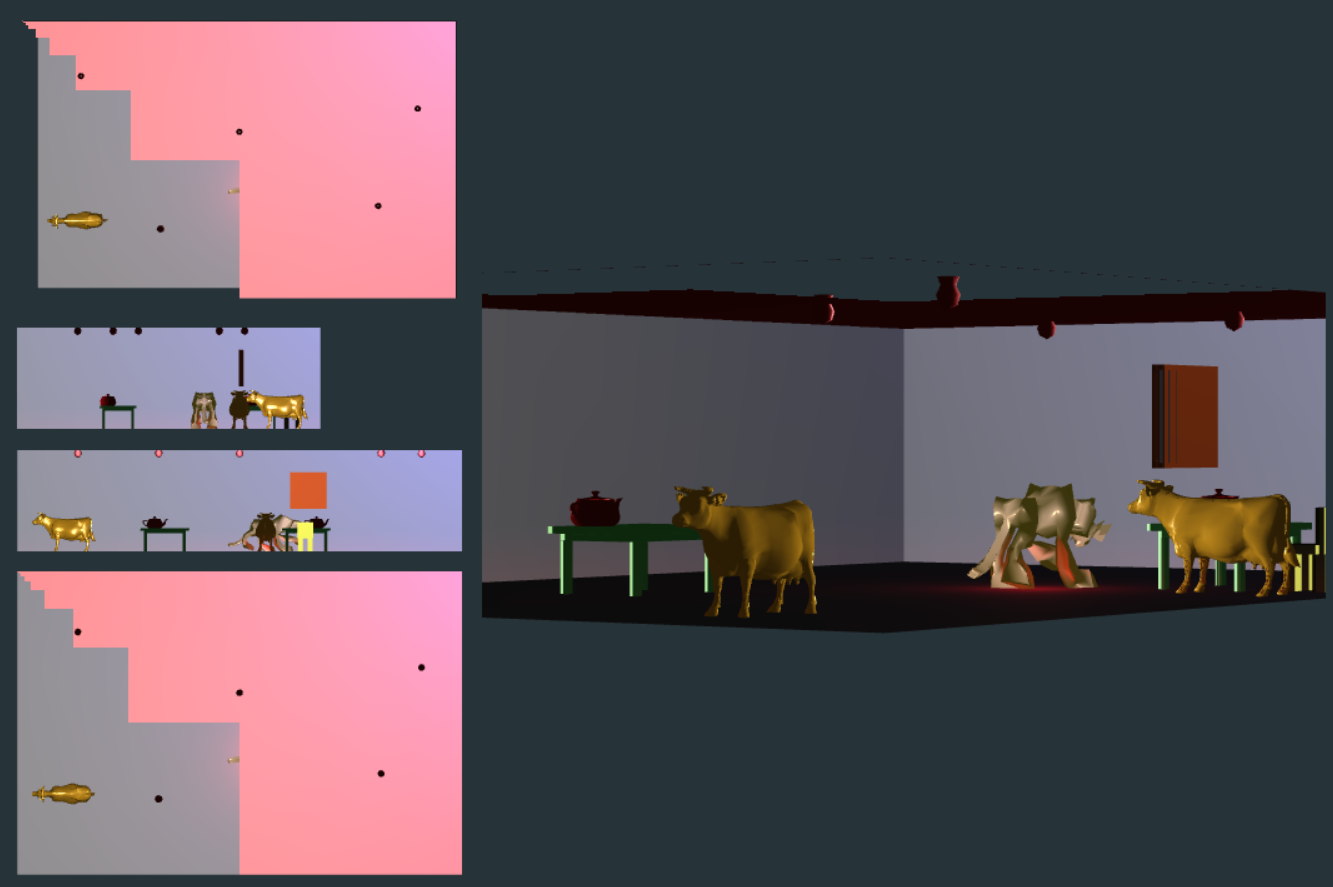


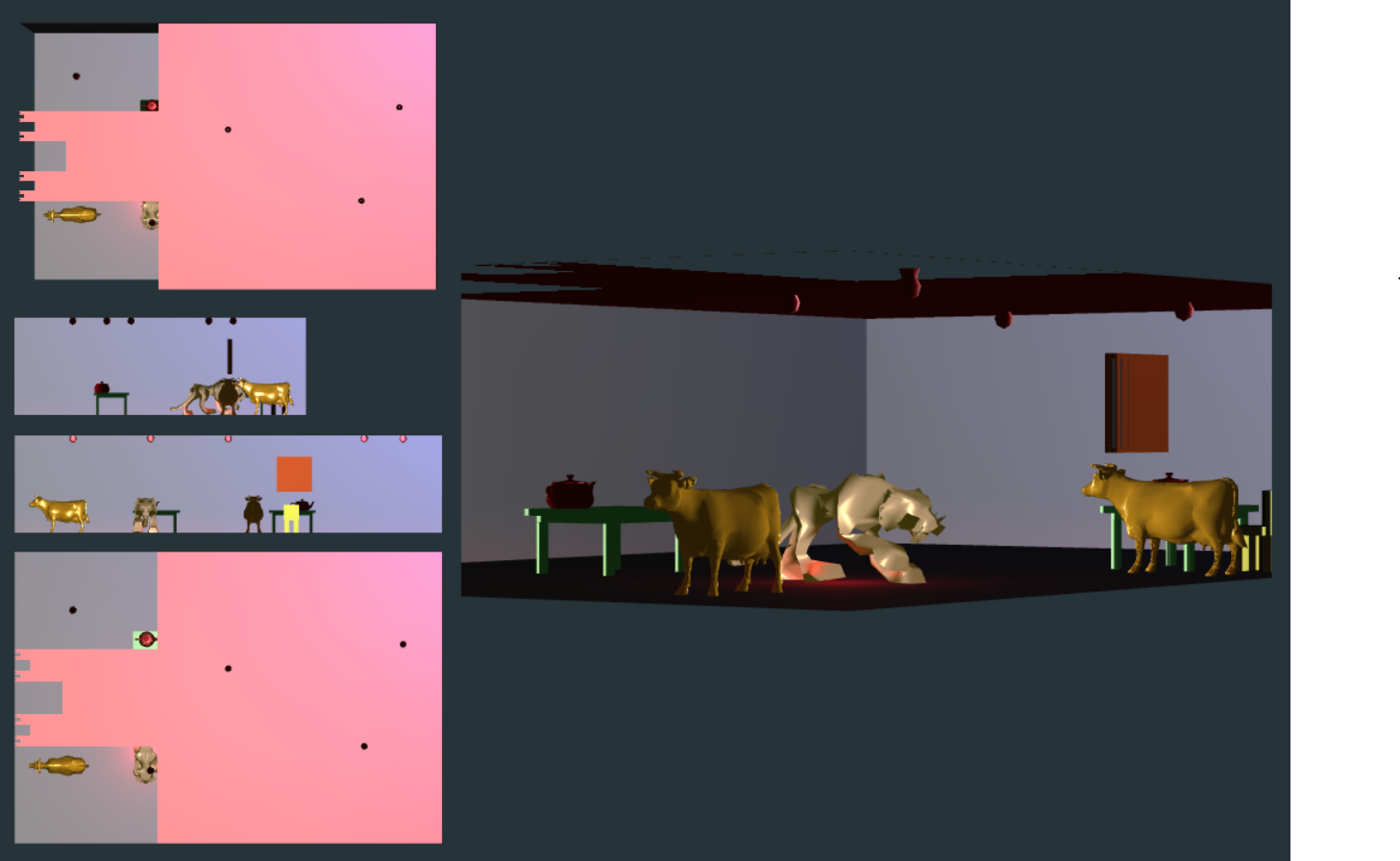
## *위의 기하 물체 관련 쉐이딩 효과 외에 그 이상의 복잡도를 가지는 기하 물체와 관련한 쉐이딩 효과를 생성하라. 점수는 복잡도와 재미에 따라 0점에서 20점까지 부여함.*

3.1에서 ‘e’를 한 번 더 누르거나, 기본화면에서 ‘s’를 누르고 ‘e’를 두 번 누르면 스크린이 벽의 모양에 맞게 바뀌어 집 내부를 볼 수 있게 됩니다.



‘e’버튼을 한 번 더 눌러 재귀적으로 discard된 스크린을 생성할 수 있습니다. ‘j’키를 눌러 인자를 증가시키고, ‘k’키를 눌러 인자를 감소시킬 수 있습니다.





# 추가 구현

## 미니맵을 위한 벽 제거

본 프로젝트는 지난 프로젝트 이전의 결과물을 토대로 합니다. 따라서 두 프로젝트들의 구현요소들이 서로 상충하지 않도록 노력했습니다. 대표적인 예가 지난 프로젝트에 구현한 building입니다. line으로 그렸던 building을 full로, 그리고shading 효과를 넣어 그리게 되면 building 내부 풍경이 보이지 않게 됩니다. 그렇다고 벽 전체를 없애면 저번 프로젝트의 결과물이 보이지 않게 됩니다. 따라서 Phong\_sc.frag파일에서 현재 building을 그리고 있는지의 여부를 판단하는 isBuilding과 벽을 어떻게 그릴건지를 나타내는 walls 변수를 받아옵니다. walls는 0, 1, 2 세 가지 값을 가지고 기본값은 1이며, 벽 전체를 보여주는 상태입니다. 2는 벽의 좌표를 보고 화면에 정면으로 보이는 벽을 제거한 상태입니다. 0은 벽을 완전히 제거한 상태입니다.