# ApplicationSettings

The settings fort he Application are persistenly stored in between excutions of the Application, in ApplicationSettings. ApplicationSettins is composed of GraphicsSettings and ControlsSettings.

Every main scene in the Application needs to have an active ‘empty’ root gameobject with the ‘ApplicationSettings’ script attached to it.

The Applicationsettings is a singleton. It is also persistent during the execution of the Application: Once it is created, it sets itself as ‘DoNotDestroyOnLoad’, which makes that this instance stays alive when transitioning between scenes.

# Layers

The scene content is ordered in several layers:

1. Sky
2. Celestial objects
3. Horizon
4. World
5. UI-VR
6. UI

The layers are rendered in the order in which they are listed above.

## Layer ‘Sky’

Contains the skydome.

Clears both dept hand color buffer (with a solid color).

## Layer ‘Celestial objects’

Contains the celestial objects like sun and moon.

Only clears depth buffer.

## Layer ‘Horizon’

Contains the horizon dome.

Only clears depth buffer.

## Layer ‘World’

Contains the virtual environment and objects in it.

* Terrain
* Buildings
* Furniture
* ? World-space interactive items
  + Light switches
  + …
* …

Only clears depth buffer.

## Layer ‘UI-VR’

Contains the world-space UI components that are attached to the player for VR mode.

Only clears depth buffer.

## Layer ‘UI’

Contains the screen-space UI components for non-VR mode.

Only clears depth buffer.

# Procedures

## Procedure: Add a construction project

1. Import assets
   1. model of building (Sketchup v2015), into Assets/KS/Model/ProjectXXX folder.
   2. Project preview image, as 2D/Sprite, into Assets/Resources/ProjectPreview/ folder.
2. Create project scene
   1. Named ‘ProjectXXX’, in folder ‘Assets/Scenes’
3. Add Lighting to project scene
   1. World/Lighting
      1. LightgroupXXX
      2. Light01
      3. Light02
      4. …
4. Add POI to project scene
   1. POI.Default
      1. L0Leefruimte
      2. …
5. Add project selection button to the Home state’s Project Selection menu
   1. This could be automated, so project selectionn scene buttons are dynamically creadted and arranged according tot he project scenes available.

## Procedure: Remove a construction project

Remove the project scene

Remove the project preview Sprite asset.

Remove the project selection button from Home state’s Project Selection menu.

# Development tools

## Monitoring application performance

The FPS can be investigated using the FPS counter widget and exported file.

## Developing/debugging a project scene

When making a lot of successive changes to a project scene, the easiest way to review and debug those changes is to set the ‘InitialProjectName’ setting on the PlayApplicationState. This makes the Play state start up with the designated scene loaded right away.

# TODO:

* Make it so that the ‘selected project’ in Home menu is the project that was last played.
  + Add ‘active project’ to application settings.
* Implement a key to generate screenshot:
  + While in ‘Play’ mode
  + Shortcut key = ?
  + Of specified resolution (1920\*1080)
  + Filename = Project name + datetime stamp.
  + (optionally) adjust rendering temporarily:
    - Hide all UI elements
    - Activate best available graphics quality setting.
  + Generated images will be usable, for example, as project preview images.
* Properly implement and document a way to generate POI.
  + Shortcut key = ?
  + To File (path:???)
* Add proper lighting to P006
* Align P006 With World Axes. This will enable using terrains more easily
  + Fix terrain for Tuin Achter
  + Add Terrain(s) in tuin voor
  + Add terrain(s) for tuin links/rechts)
    - Zet beplanting als ‘mesh’ details
* Add proper lighting to P008
* Add proper lighting to P011
* Add proper lighting to P024
* Add proper lighting to P025
* Add proper furniture to P011
* Review and finalize reusable lighting components as prefabs
* Initialize lighting in project 008 and 001 properly
* Load reusable ‘furniture’ components as prefabs.
  + Set material settings.
* Create ‘Furniture-less’ version of model for each project (skp v2015)
* Use furnitureless model in project scenes
* Prepare furniture in projects, using tweaked unity furniture
* Sky: Implement Clouds
  + Based on perlin noise?
  + Add heightmap for extra visual ‘depth’
* SkyDome : Implement Fog
* SkyDome : Implement Finegrained control over Sky gradient.
  + Shader: add property \_SkyLight1InfluenceRangeAngle
* SkyDome Shader: implement properly support for arbitrary number of celestial objects.
* Sky behavior: use SkyDome Shader support for arbitrary number of celestial objects, to properly represent both sun and moon.
* SkyDome : Implement making rendering ground optional.
* SkyDome : Implement timed ‘Ground Colors’