У деяких відносинах 3D-камеру концептуально можна розглядати як справжню камеру, щоб налаштувати знімок, вам потрібно правильно розташувати камеру і повернути її так, щоб вона дивилася на те, що ви хочете побачити.

Але на цьому відмінності справді закінчуються.

З традиційним об'єктивом світло камери проходить через об'єктив, що проектується на плоский шматок світлочутливої плівки або матриці.

Ви можете сфокусувати об'єктив для різних ефектів глибини різкості. Ви також можете замінити об'єктив для різних ефектів, таких як телефото-об'єктив, ширококутний об'єктив, риб'яче око і т.д.

В ігровому движку камера насправді просто проекція тривимірної геометрії на двовимірну площину дисплея. Ви можете імітувати такі ефекти лінз, як фокусування, відблиски та риб'яче око, але вони вимогливі до процесора.

Є два різні типи 3D-проекцій, які ви можете створити у своєму проекті, перспективні та ортографічні.

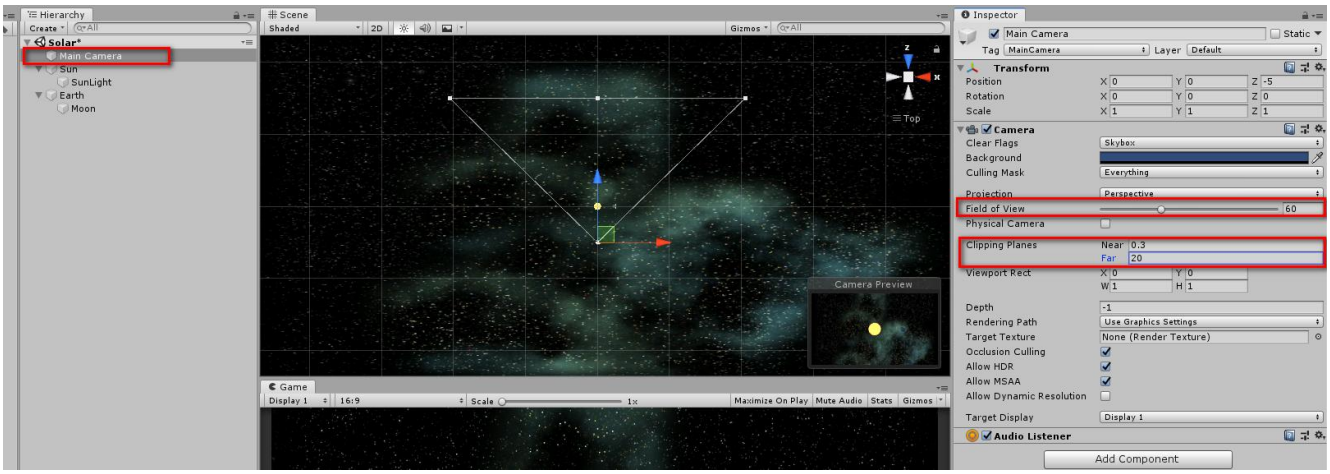
Перспективна проекція найбільше схожа на справжню камеру і демонструє світ, як ми його бачимо. Це найчастіше використовується в іграх від першої та третьої особи. Об'єкти, які знаходяться ближче до камери, більше об'єктів, що знаходяться далі від камери.

Ортографічних проекцій у світі немає. Комп'ютер підтримує той самий відносний розмір об'єкта, незалежно від того, наскільки близько чи далеко від камери. Зазвичай ортогональні камери використовуються для зйомки зверху донизу або ізометричного вигляду на гру.

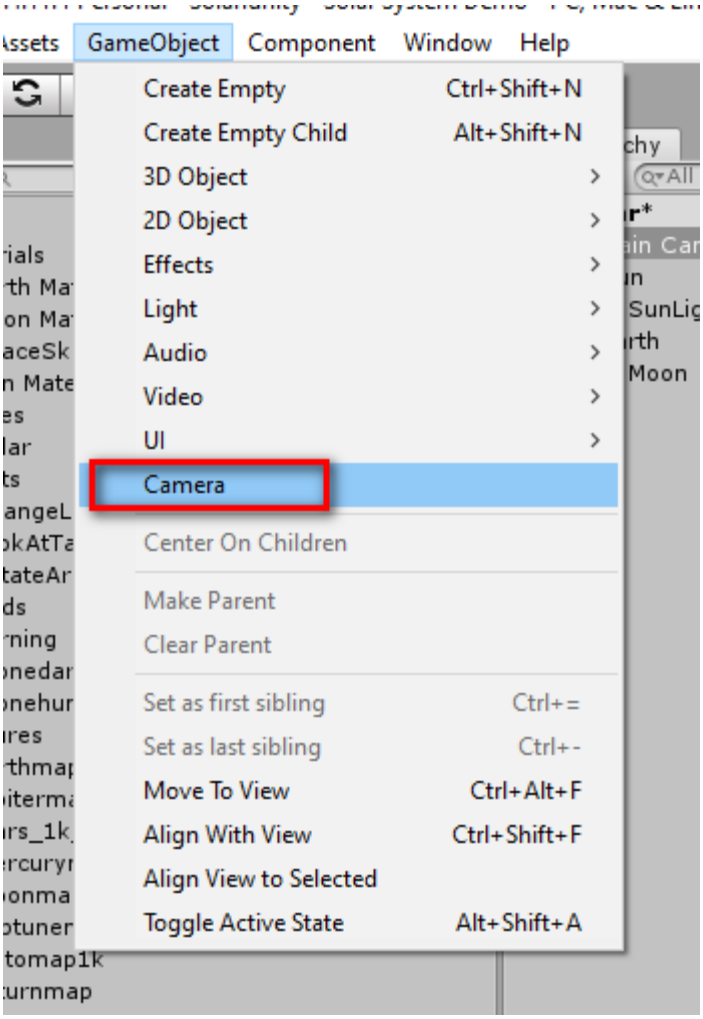
Перспективна камера має зрізану піраміду з вузьким полем огляду.
ближче до камери та ширше поле зору далі від камери. У міру наближення об'єкта до камери він здаватиметься більшим або ближчим.

Ортографічна камера має усічену піраміду прямокутного вигляду.
з постійним полем зору або ортогональним розміром усюди. Об'єкти, де б вони не були перед камерою, матиме такий самий розмір щодо інших об'єктів.

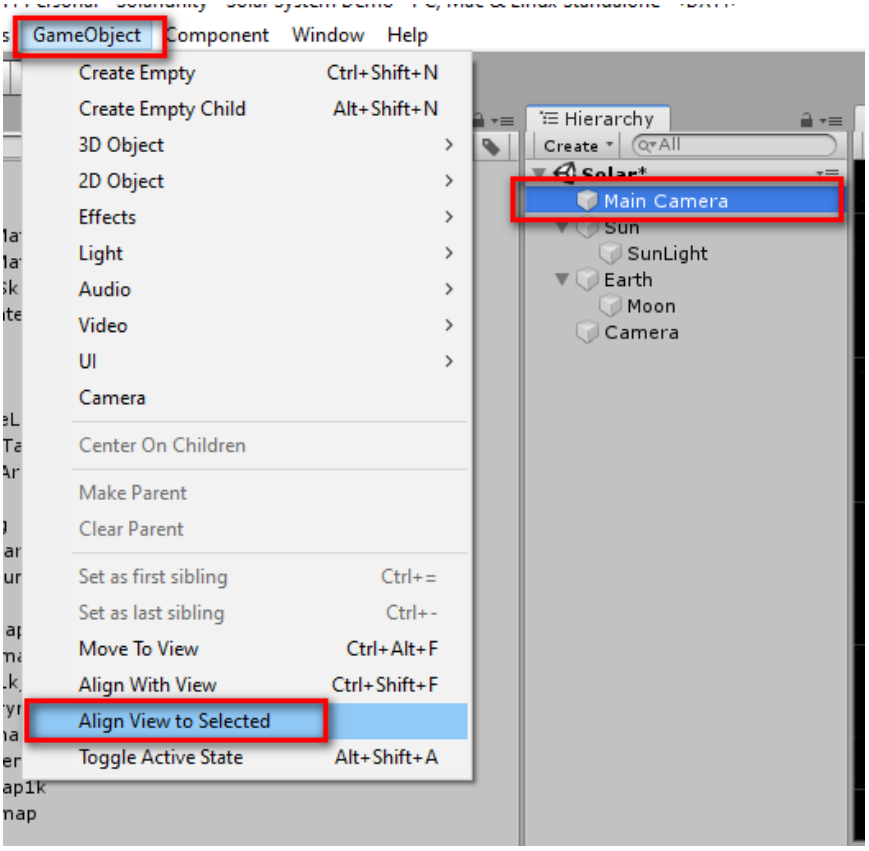
Виділяємо основну камеру та налаштовуємо параметри:



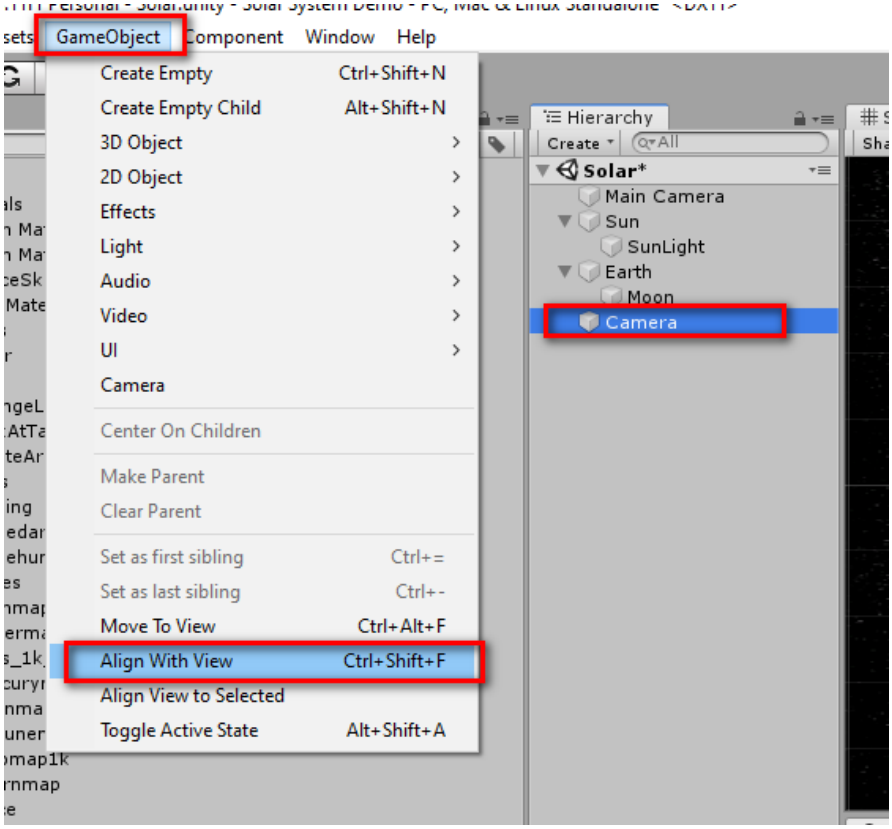
Додаємо ще одну камеру:



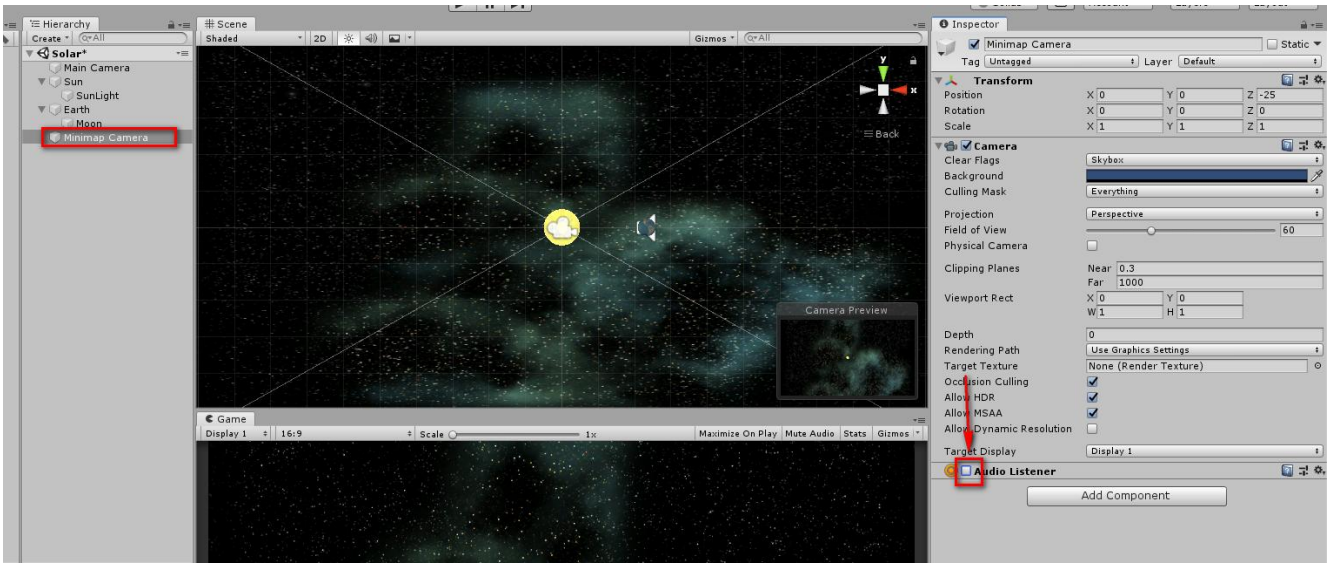
Виділяємо основну камеру і перемикаємося на вигляд із неї:



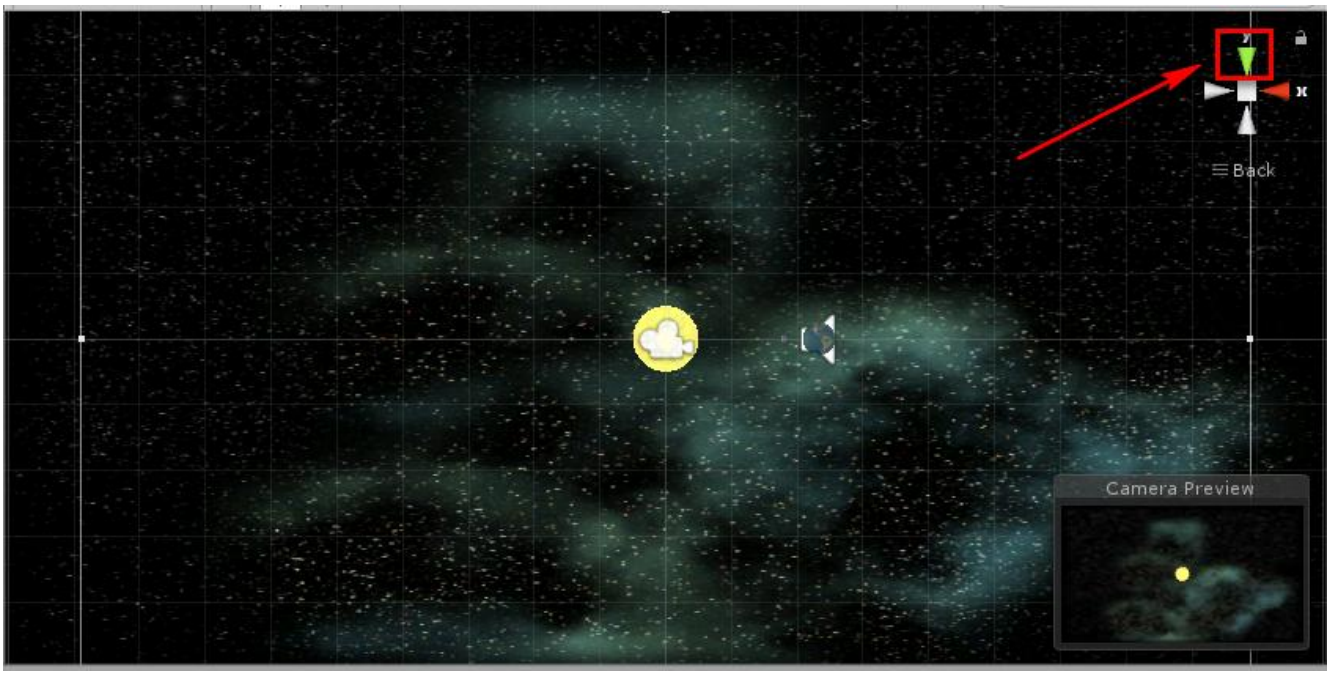
Виділяємо другу камеру та переміщуємо її до цього виду:



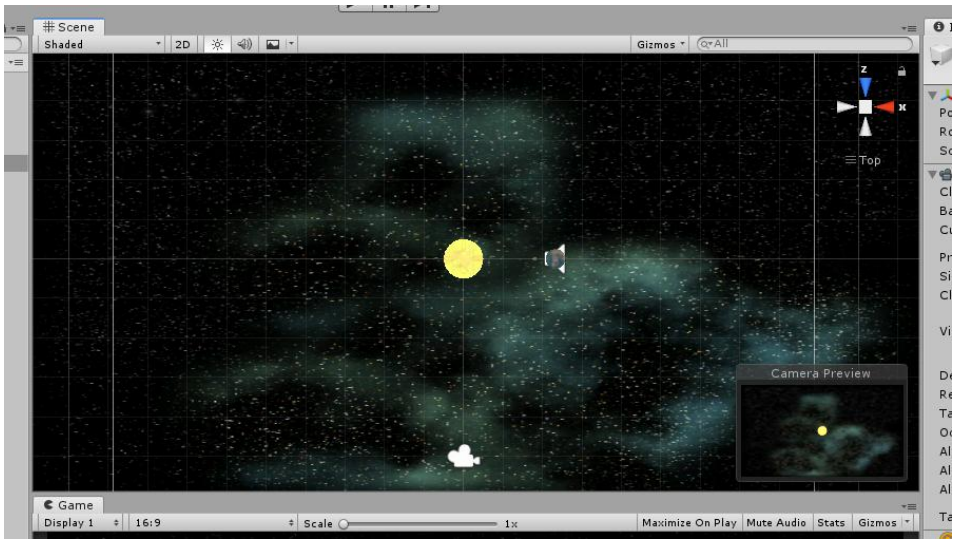
Відключаємо у другої камери аудіо слухач:



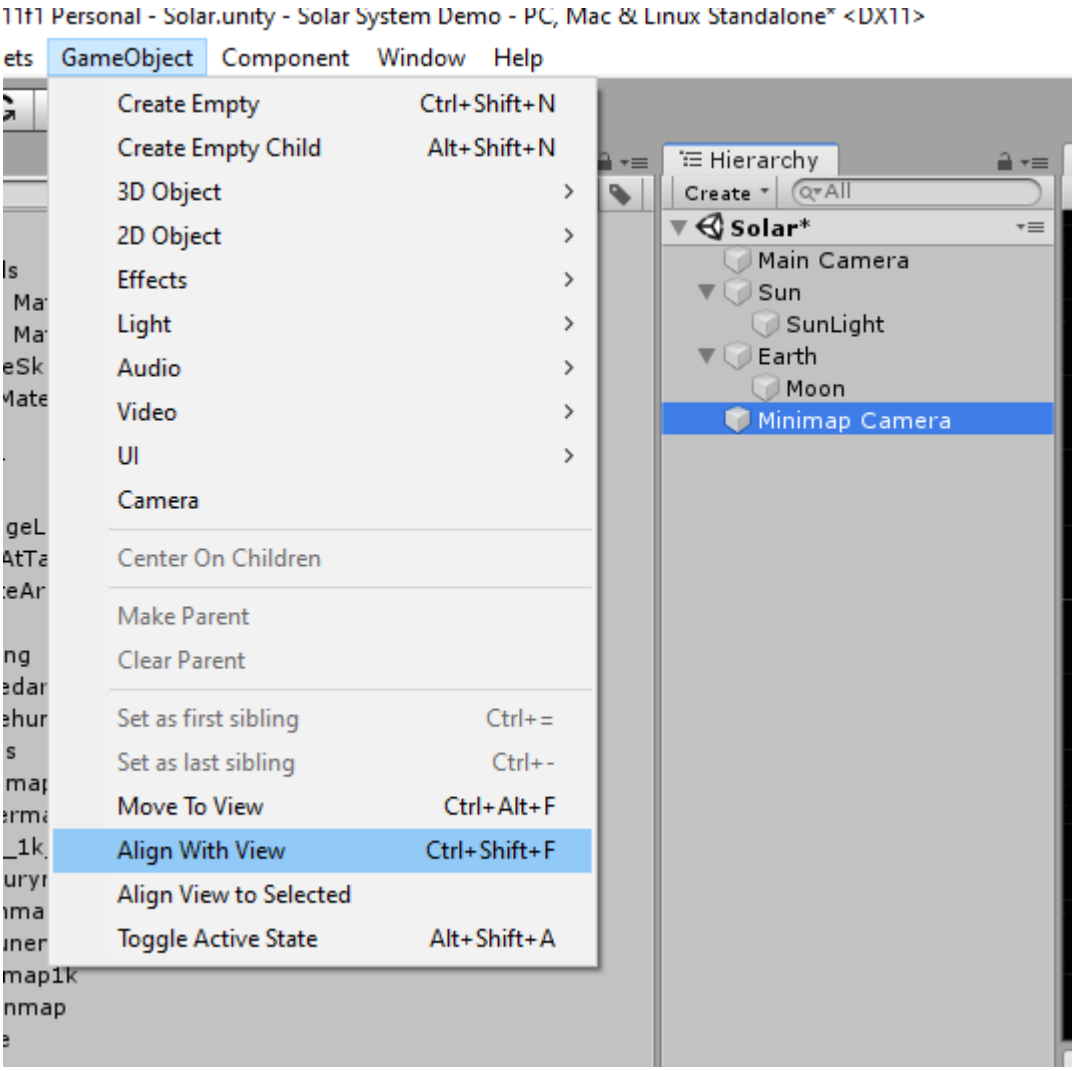
Включаємо вид зверху:



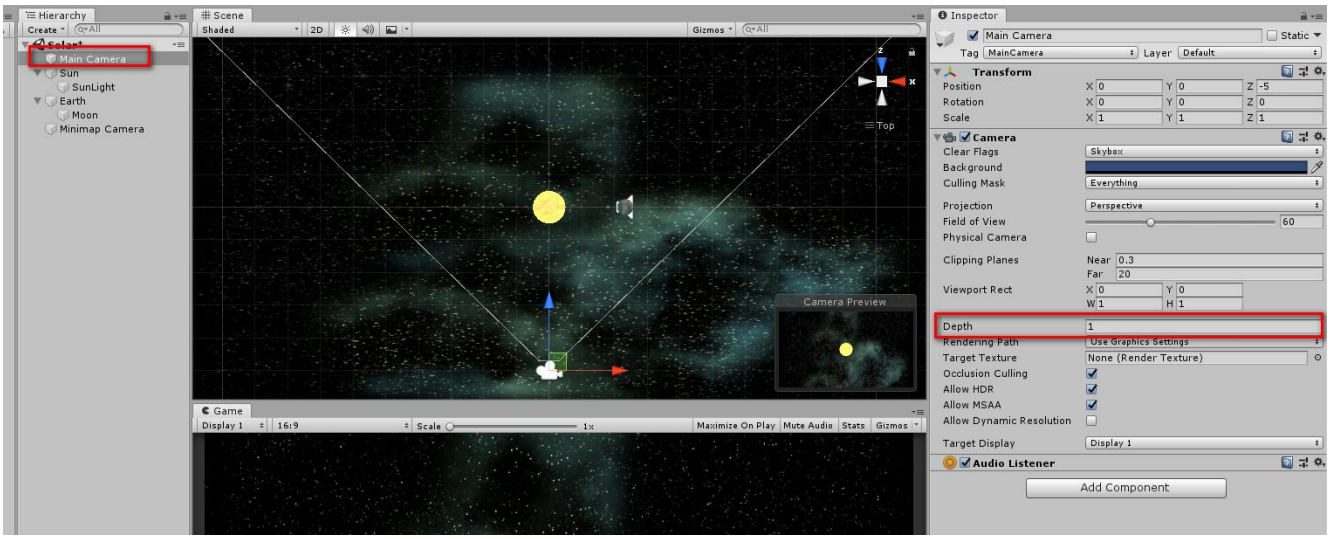
Наближаємо або віддаляємо так, щоб добре було видно всі планети:



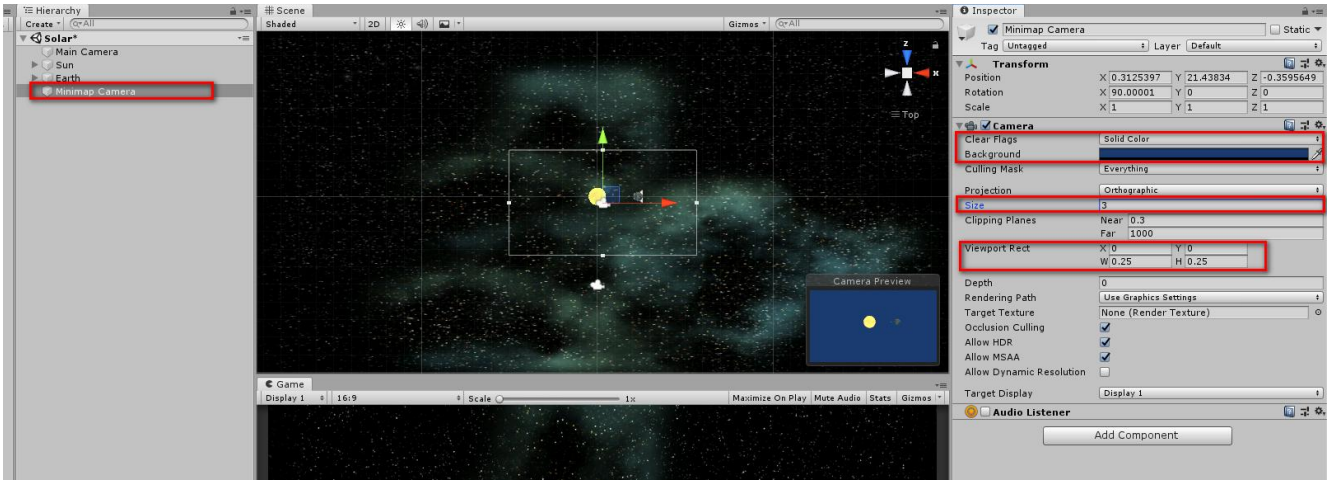
Вирівнюємо положення камери за цим видом:



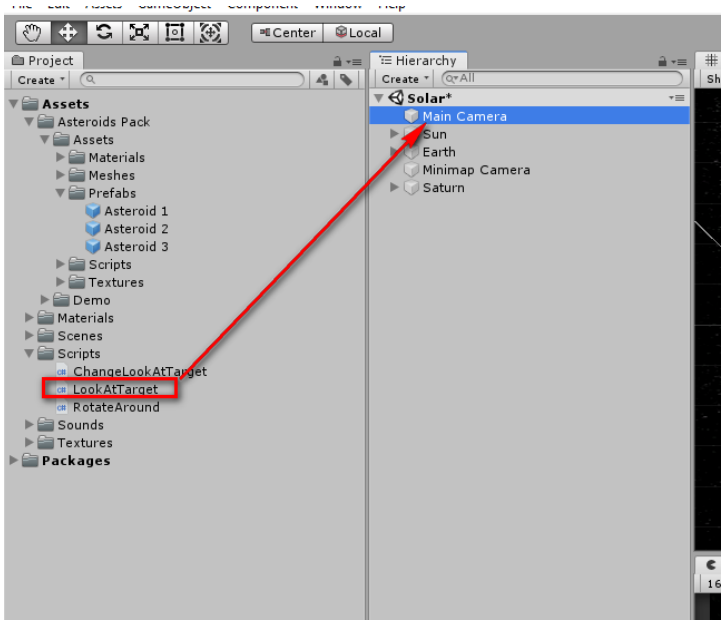
Нова камера зараз основна, саме з неї ми бачимо сцену. Ви можете переключити вигляд, використовуючи глибину. АЛЕ, потрібно повернути основну камеру вниз для подальшого виконання (Спробуйте змінити значення як на картинці та запустити симуляцію,потім поверніть значення -1)



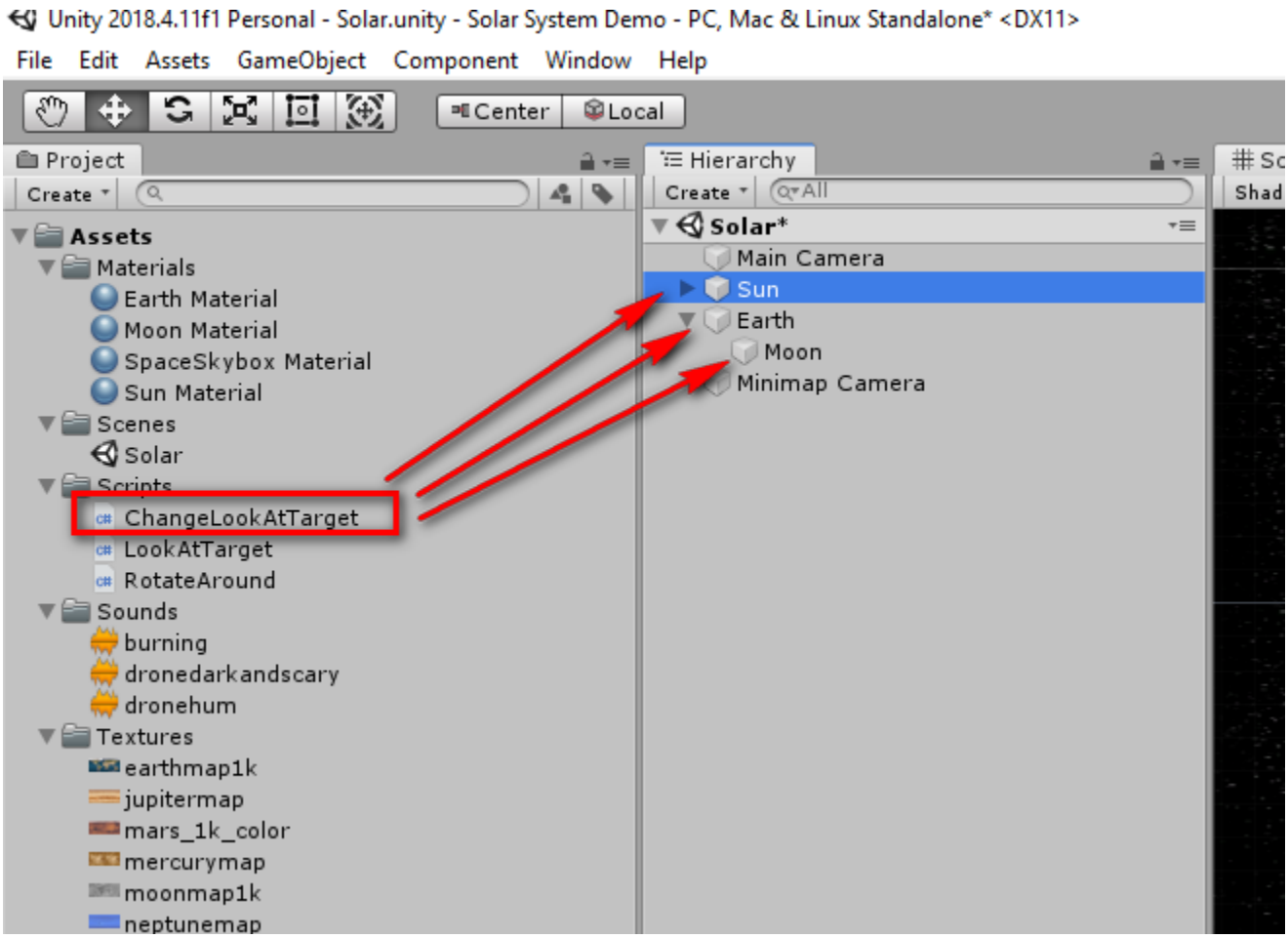
Виділяємо другу камеру та налаштовуємо їй параметри:



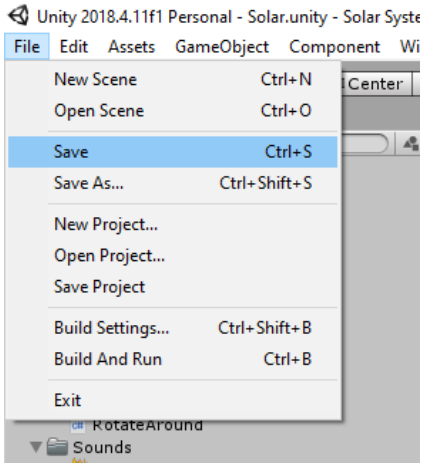
Накладаємо на головну камеру скрипт LookAtTarget. Він зберігає у собі об'єкт, який має концентрувати увагу камери.



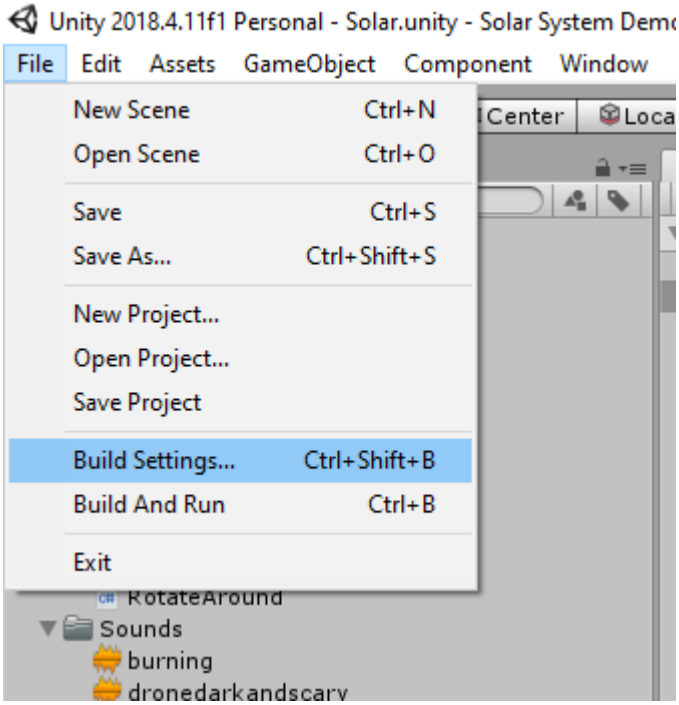
На решту об'єктів накладаємо скрипт, що визначає клацання мишкою для концентрації на об'єкті:



Зберігаємо сцену:

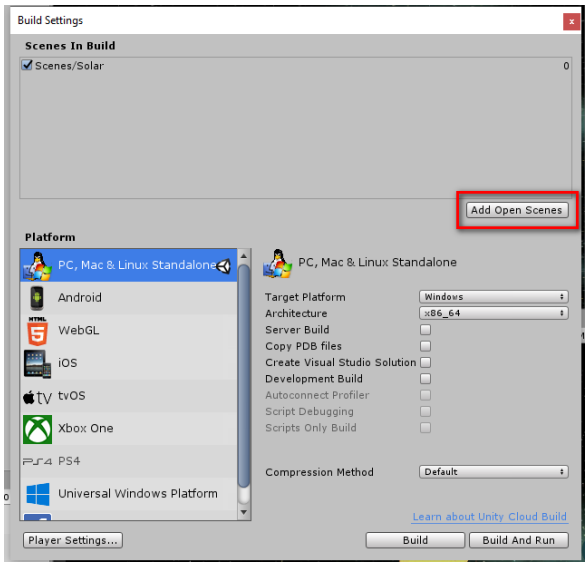


Відкриваємо налаштування білда:

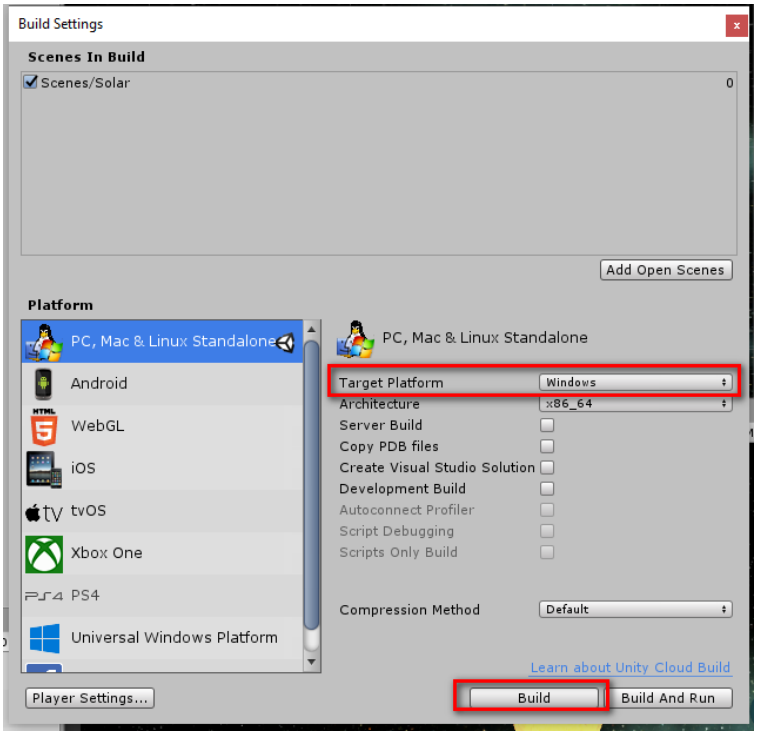


Варто звернути увагу, що те, як файли зберігаються в проекті, також зберігаються і в пам'яті комп'ютера.

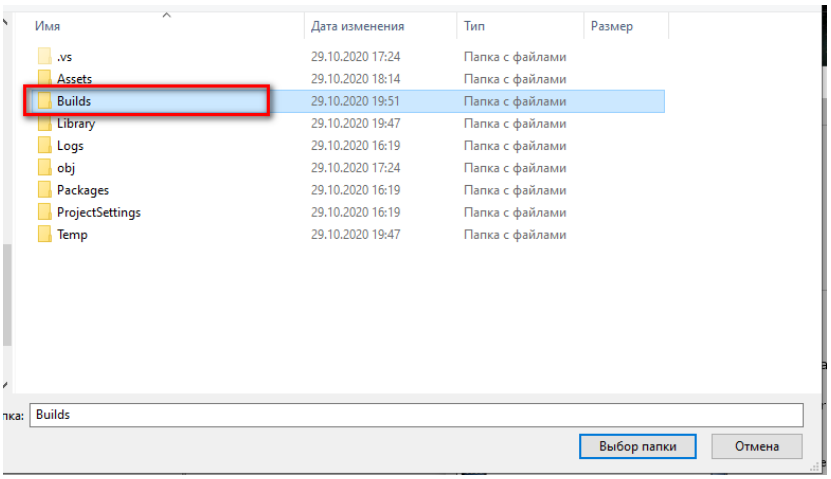
Додаємо поточну сцену до білду:



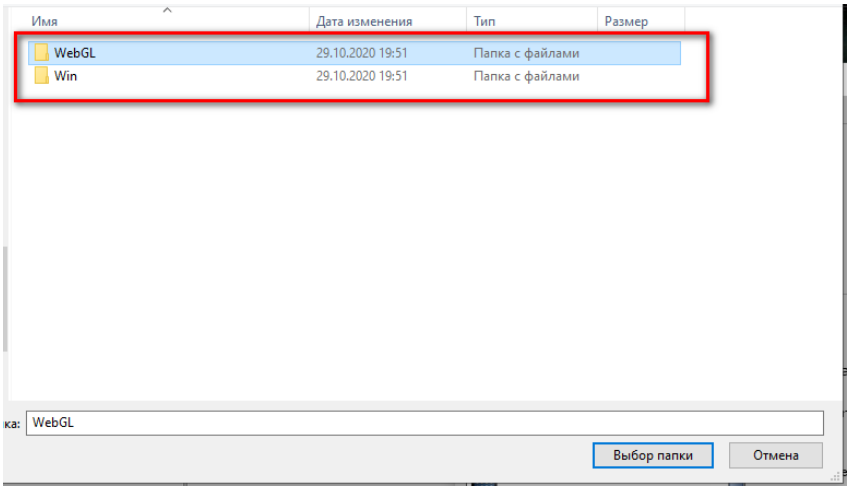
Вказуємо платформу для білда:



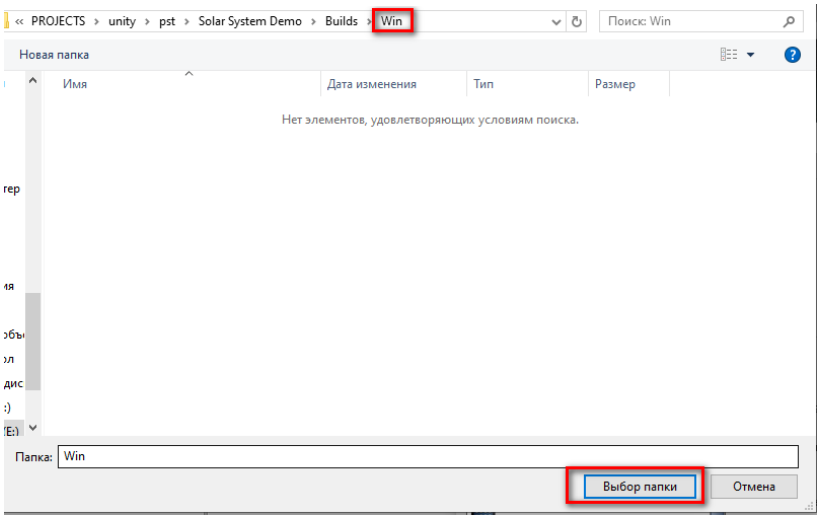
Створюємо папку:



У ній ще дві папки:

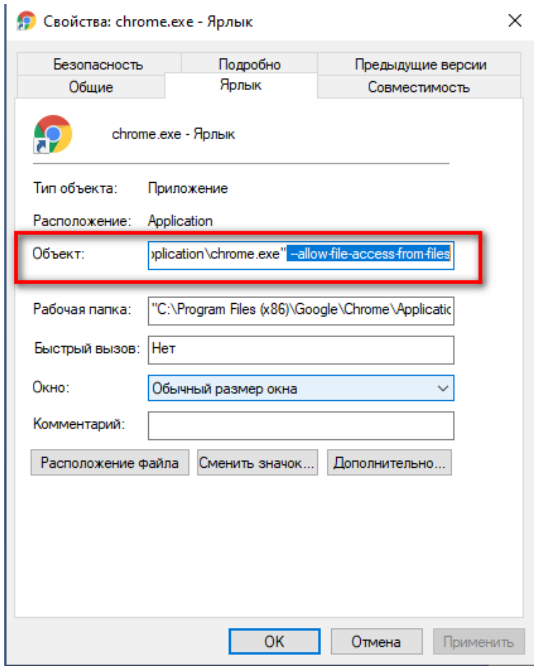


Вибираємо папку:

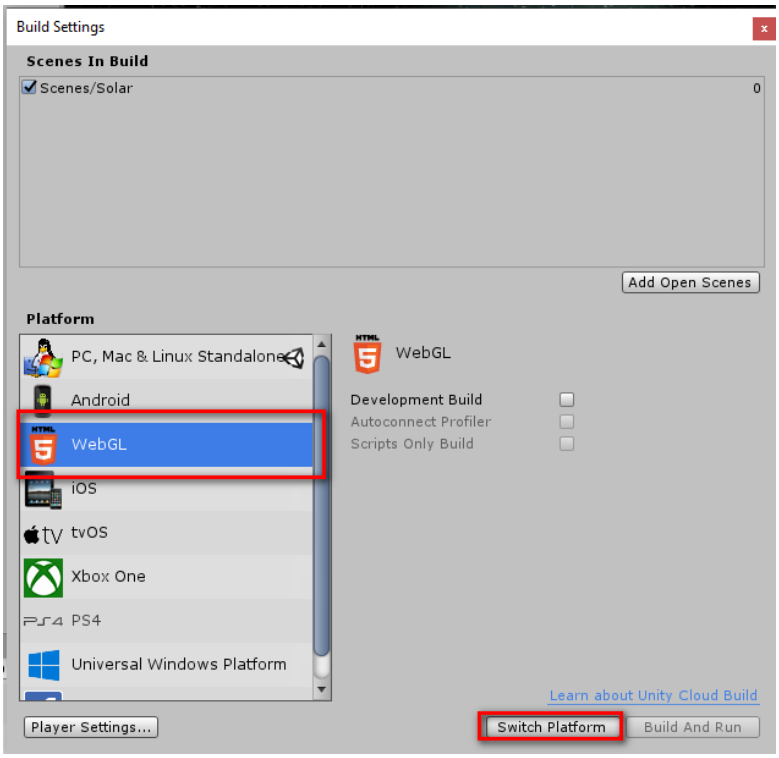


Для запускуWebGL необхідно в ярлику браузера прописати:

--allow-file-access-from-files



Перемикаємо платформу наWebGL



Вибираємо папку для білда і запускаємо його: