Co-funded by the
Erasmus+ Programme
of the European Union

GAMEHUB University Enterprises Cooperation In Game Industry In Ukraine

ТРЕНИНГ
«Разработка компьютерных игр в Unity 3D»

Материалы тренинга подготовили:

к.т.н., доцент кафедры ИТ Кирийчук Д.Л.

к.т.н., доцент кафедры ИТ Ляшенко Е.Н.

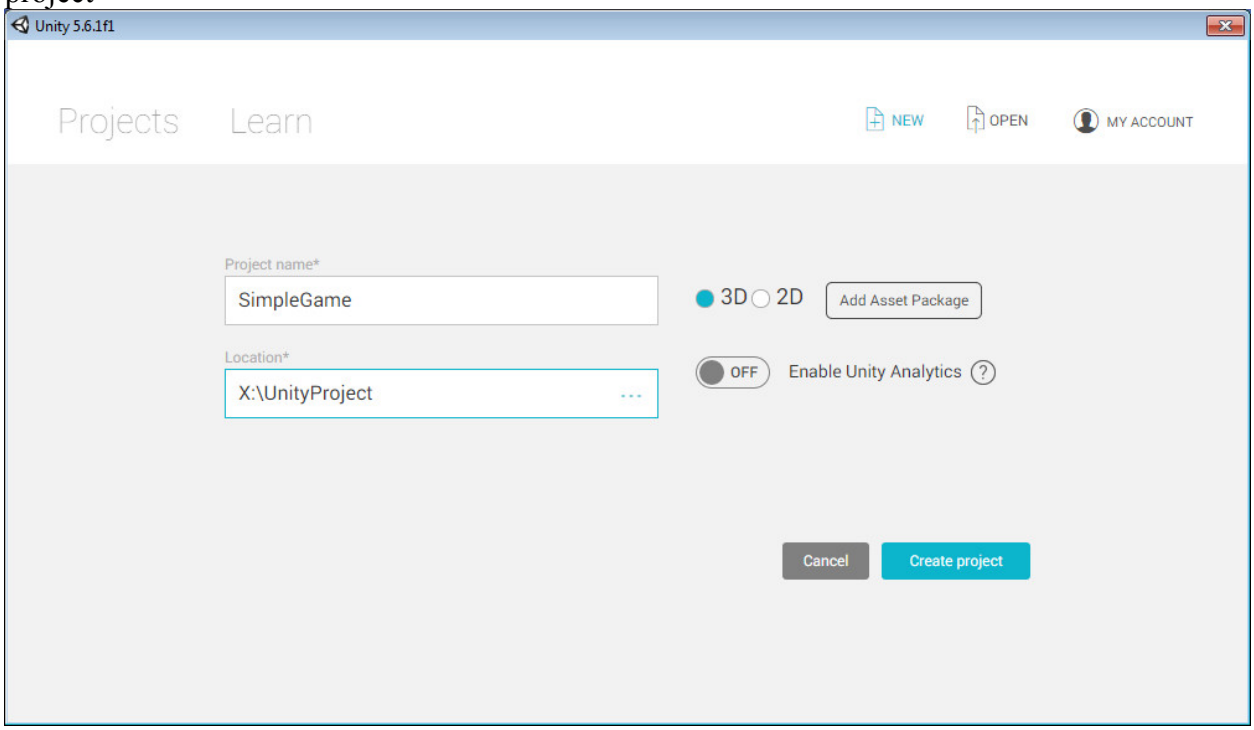
18 - 19 мая 2017 г.
13:00, аудитория 320 (3 корпус) Херсонского национального технического университета.

Erasmus+ №561728-EPP-1-2015-1-ES-EPPKA2-CBHE-JP
«GameHub: University-enterprises cooperation in game industry in Ukraine»

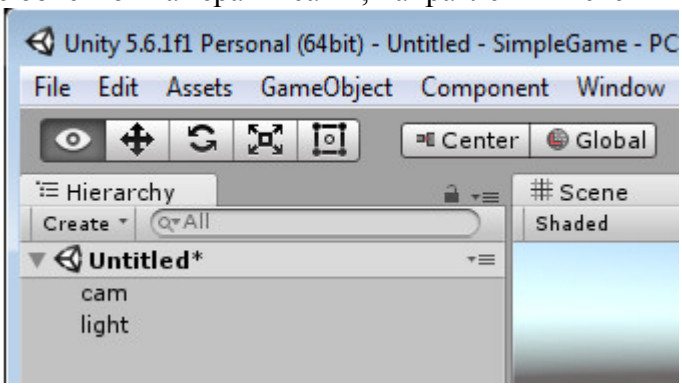
При разработке материалов тренинга использовались обучающие уроки с официального сайта https://unity3d.com/ru/learn/tutorials

ЭТАП 1 СОЗДАНИЕ ПРОЕКТА

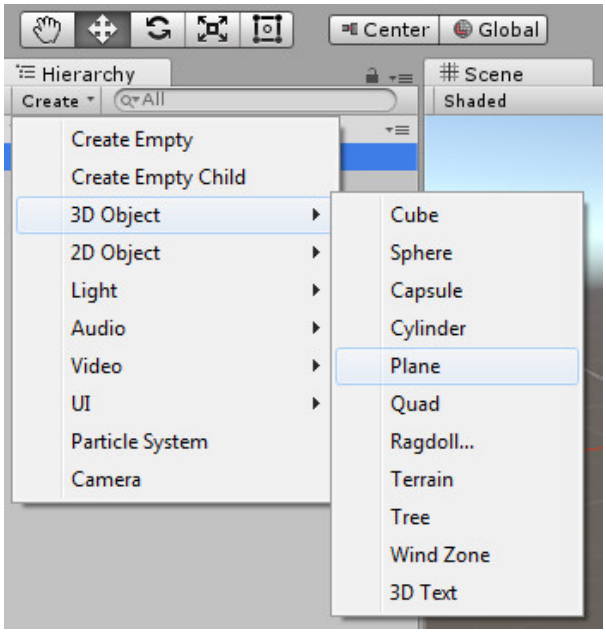
1. Задать имя, каталог для хранения файлов и формат игры - 3D. Затем нажать Create project



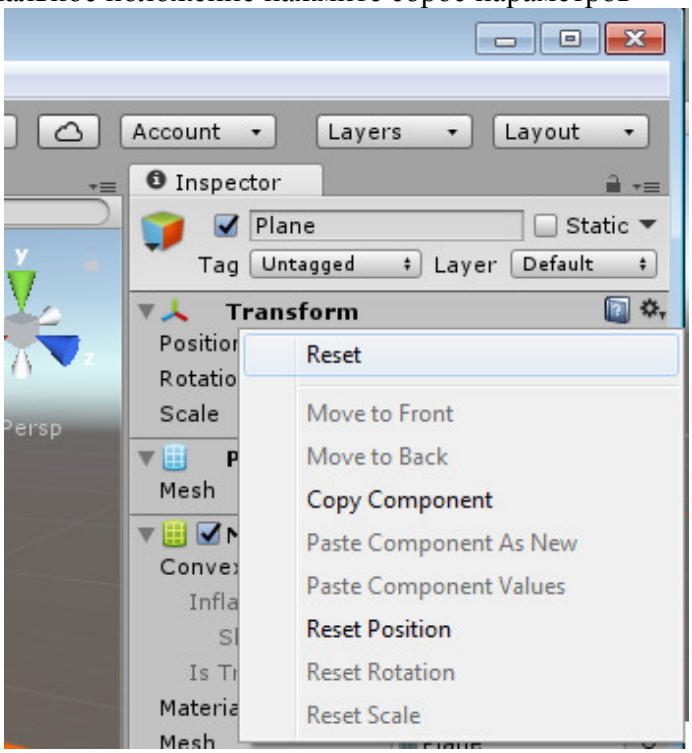
2. Изменить название объектов камера – «cam», направленный свет – «light»



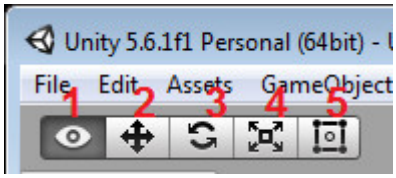
3. Добавить объект плоскость «plane» в дерево иерархии и назовите его «ground»



4. Переместить данный объект по направляющим (красной, зеленой и желтой стрелам).
В окне инспектора (Inspector) будут изменяться координаты, их можно задавать вручную.
Верните координаты объекта (Position) x=1, y=2, z=3
Чтобы вернуть в начальное положение нажмите сброс параметров

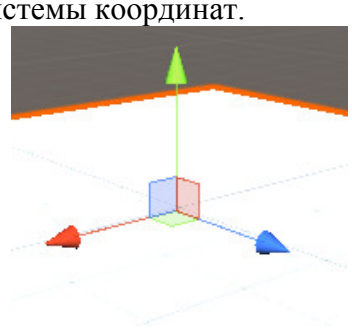


5. Объект можно не только перемещать в плоскости. Обратите внимание на кнопки на панели инструментов под главным меню программы.

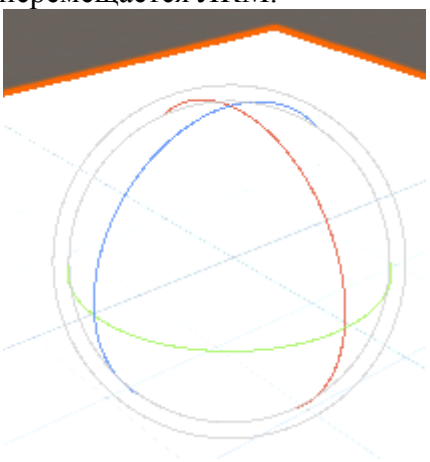


Первая включает режим полета наблюдателя:
Зажатая ЛКМ перемещает сцену относительно наблюдателя.
Зажатая ПКМ в зависимости от выбранного режима «Persp» или «Iso» позволяет поворачивать или камеру наблюдателя или все объекты относительно центра экрана соответственно. Также последний можно получить, зажав клавишу Alt и ЛКМ при перемещать мышь.

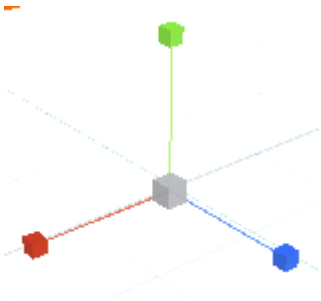
Вторая включает перемещение выделенного в дереве иерархии (Hierarchy) 3D объекта по направляющим стрелкам, достаточно схватить ЛКМ за выбранную стрелку и переместить в заданном направлении. Также можно переносить по выбранной плоскости схватив один из параллелограммов вначале системы координат.



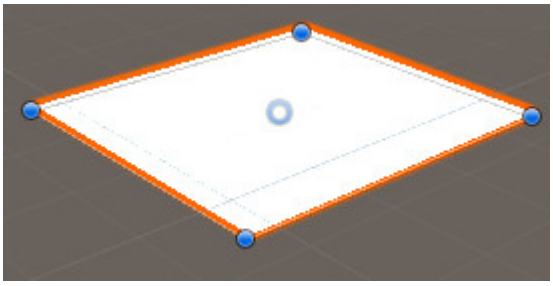
Третья – повороты выполняются аналогично, только выделяется соответствующего цвета направляющая окружности и перемещается ЛКМ.



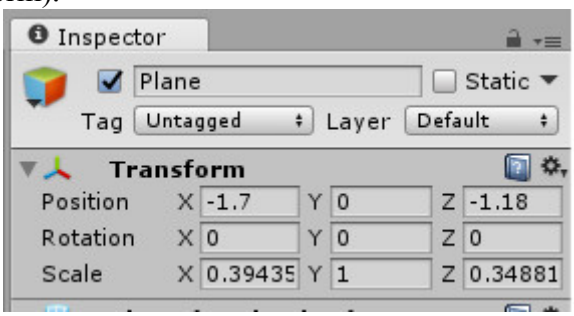
Четвертая – режим изменения масштаба относительно центра объекта, аналогично ЛКМ по цветным направляющим.



Последняя – масштабирование относительно края объекта, изменения производятся перемещением узла.

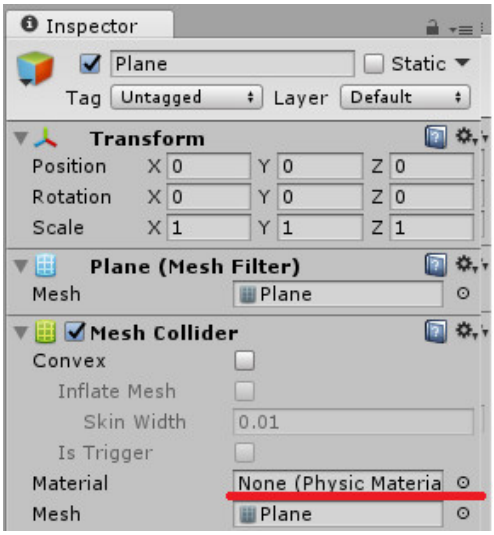


Аналогичные операции можно производить в инспекторе (Inspector) в разделе трансформаций (Ttransform).



6. Клавиша F переносит камеру так, чтобы выделенный объект расположился в центре экрана.

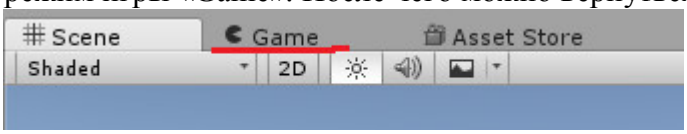
7. Любому объекту можно задать цвет, например, изменим плоскость.
Для этого создадим в дереве ресурсов (Assets) каталог материалов (materials), а в нем новый материал. Выбрав материал изменим его свойство Albedo в инспекторе.
Применим свойство материала к объекту плоскость, через инспектор свойство «Mesh render», «Materials» или просто перенести материал из ресурсов на объект.



8. Создайте в дереве иерархии ещё один объект – сферу (см.пункт 3), назовите его «egg» и также измените его цвет, но на любой другой.

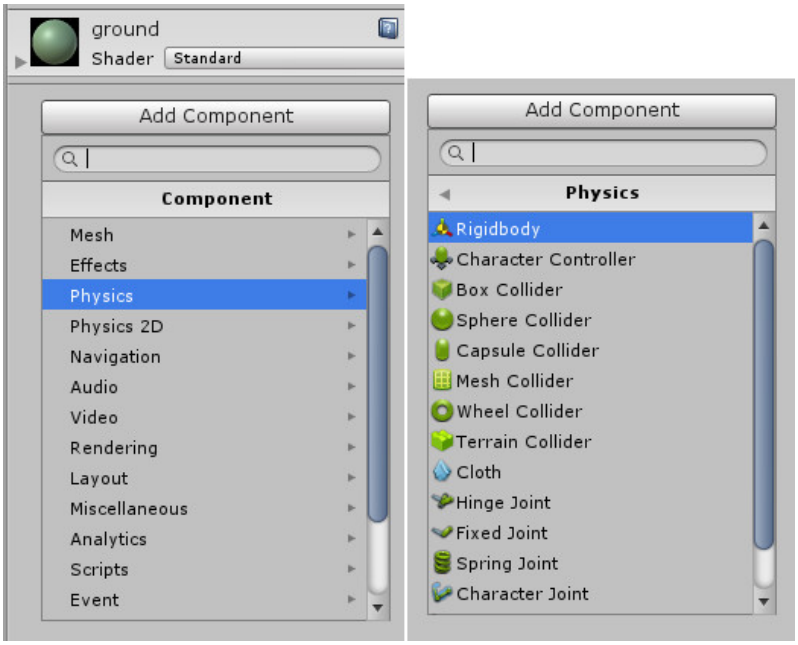
9. Можно управлять отображением теней объектов в разделе «Mesh render» свойство «Receive Shadows»

10. Переключимся в режим игры «Game». После чего можно вернуться в режим «Scene»



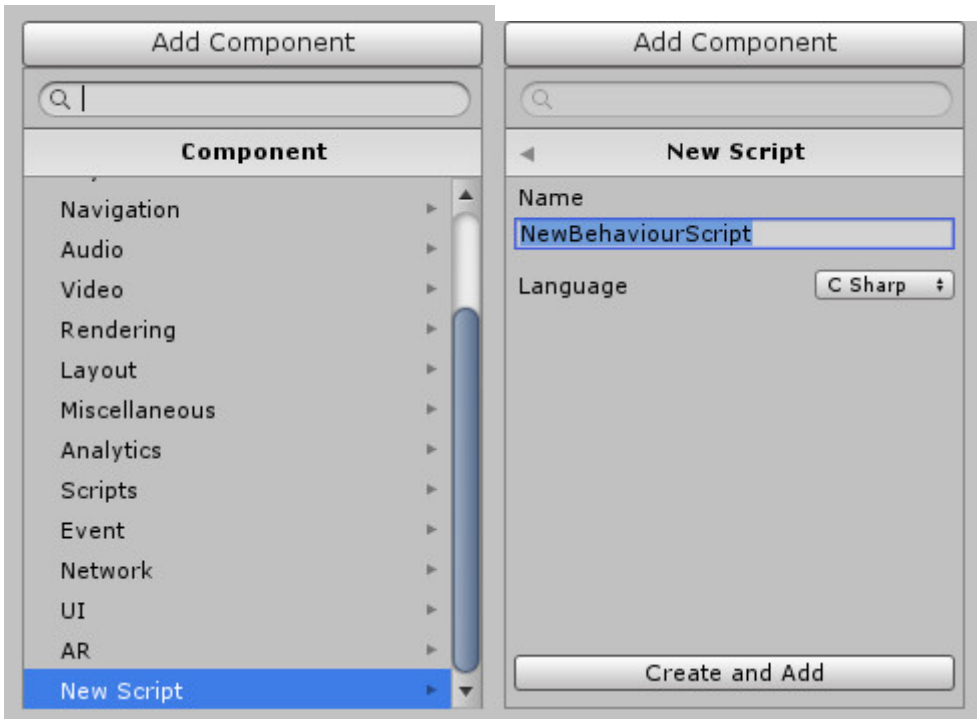
11. Сбросьте свойства «Transform» объектов сфера и плоскость, что переместит их в начало координат. Затем задайте позицию и направление для камеры, чтобы в режиме игры было видно всю плоскость.

12. Добавим свойство физических тел для объекта сфера.

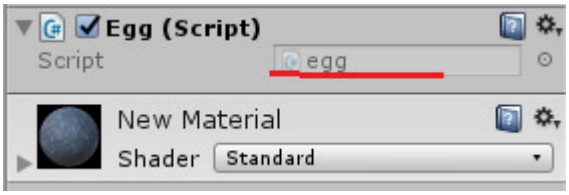


13. Просмотрите изменения в режиме игры. Задайте уклон плоскости, затем верните обратно.
14. Найдите и добавьте в дерево ресурсов «Assets» текстуры земли и гранита.
15. Создайте соответствующий материал.
16. Примените к объектам egg и ground соответствующие текстуры.

17. Добавьте для объекта egg компонент «Script», зададим имя скрипта также egg и язык «C #».



18. Создадим в дереве ресурсов каталог «scripts» в который переместим наш скрипт.
19. Изменим скрипт, для чего можно дважды нажать на него в дерево ресурсов либо в свойствах объекта egg.



20. Добавим в код объявление приватного объекта отвечающего за физику твердого тела
```
private Rigidbody rb;
```
затем в инициализацию добавим получение экземпляра этого объекта
```
rb = GetComponent<Rigidbody>();
```
теперь в методе обновления можно изменять свойства твердого тела (ТТ)
```
float h = Input.GetAxis("Horizontal");
rb.position += new Vector3(h, 0, 0)
```
Для чего сначала считываем с устройства ввода и изменяем свойство ТТ
После чего проверим перемещение объекта в режиме «game»

22.Добавим зависимость от скорости, для чего добавим новое свойство в этот объект
```
public float speed;
```
Затем в окне свойств ТТ egg -> «RigidBody» изменим значение поля speed с 0 на 10. Затем проверим изменения.

23. Аналогично – для второй координаты в событии обновления соответственно
```
float v = Input.GetAxis("Vertical");
...
rb.position += new Vector3 (h, 0, v)*speed;
```

24. Изменим перемещение объекта, на перемещение физического тела.
```
rb.AddForce( new Vector3(h, 0, v)*speed );
```
Проверить изменения.

25. Добавим пустой объект «Create» «Create Empty»

26. Добавим «стену», для чего добавим объект куб «Cube» и изменим свойства размера на следующие 10,1,0.1 и смещение на 0,0,-5
Аналогично для всех стен.

27. Перемещение камеры
```
    public GameObject player;
    private Vector3 offset;

    // Use this for initialization
    void Start () {
        offset = transform.position - player.transform.position;
       }

       // Update is called once per frame
```

```
    void LateUpdate () {
     transform.position = player.transform.position + offset;
     //transform.LookAt(player.transform.position);

}
```

28. Изменение капсул

```
    void Update () {
     transform.Rotate(new Vector3(100, 180, 120) * Time.deltaTime);
    }
```

29. Изменение egg

```
public float speed;
public Text countText;
private Rigidbody rb;
int count = 0;


void SetCountText()
{
    countText.text = "Найдено капсул: " + count.ToString();
}

void Start () {
    rb = GetComponent<Rigidbody>();
    SetCountText();
}


void FixedUpdate () {
        if (rb.position.y < 1e-5)
        {
            if (Input.GetKey(KeyCode.LeftShift))
                rb.velocity = new Vector3(0, 0, 0);
            else
                if (Input.GetKeyDown(KeyCode.Space))
                rb.AddForce(speed * new Vector3(MoveH, 20, MoveV));
            else
                rb.AddForce(speed * new Vector3(MoveH, 0, MoveV));
        }
}

void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag("capsule"))
        {
            other.gameObject.SetActive(false);
            count += 1;
            SetCountText();
        }
            //Destroy(other.gameObject);
    }
```