

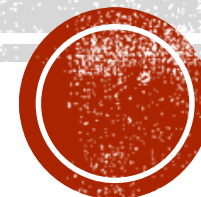


Co-funded by the
Erasmus+ Programme
of the European Union



University-Enterprises Cooperation
In Game Industry In **Ukraine**

«РОЗРОБКА КОМП'ЮТЕРНИХ ІГОР В UNITY3D »



Козіна О. А.

561728-EPP-1-2015-1-ES-EPPKA2-CBHE-JP



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



UNITY – ГНУЧКА, ШВИДКА І ТЕХНІЧНО ДОСКОНАЛА СЕРЕДА



Потужний
рендеринг



2D і 3D



C #, Javascript



Інтуїтивно
зрозумілі
робочі процеси



Глибока
оптимізація





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



University-Enterprises Cooperation
In Game Industry In **Ukraine**

ПРИСКОРЯТЬ ПРОЦЕС СТВОРЕННЯ ІГОР ЗНАННЯ:

- принципи об'єктно-орієнтованого програмування (ООП);
- знання однієї з мов програмування;
- знання однієї з програм 3D-моделювання (об'єкти, моделі, матеріали, текстури).





ЗАВДАННЯ

- ⊕ Розробити перший рівень комп'ютерної тривимірної гри в жанрі шутер від першої особи. Умова проходження рівня — знищення всіх додаткових динамічних об'єктів - терористів.
- ⊕ Ігровий простір реалізувати як сонячний день району покинутого міста.
- ⊕ Розробити і реалізувати механізми взаємодії між усіма динамічними об'єктами.





ОСНОВНІ ЕТАПИ СТВОРЕННЯ ТРИВИМІРНИХ СЦЕН

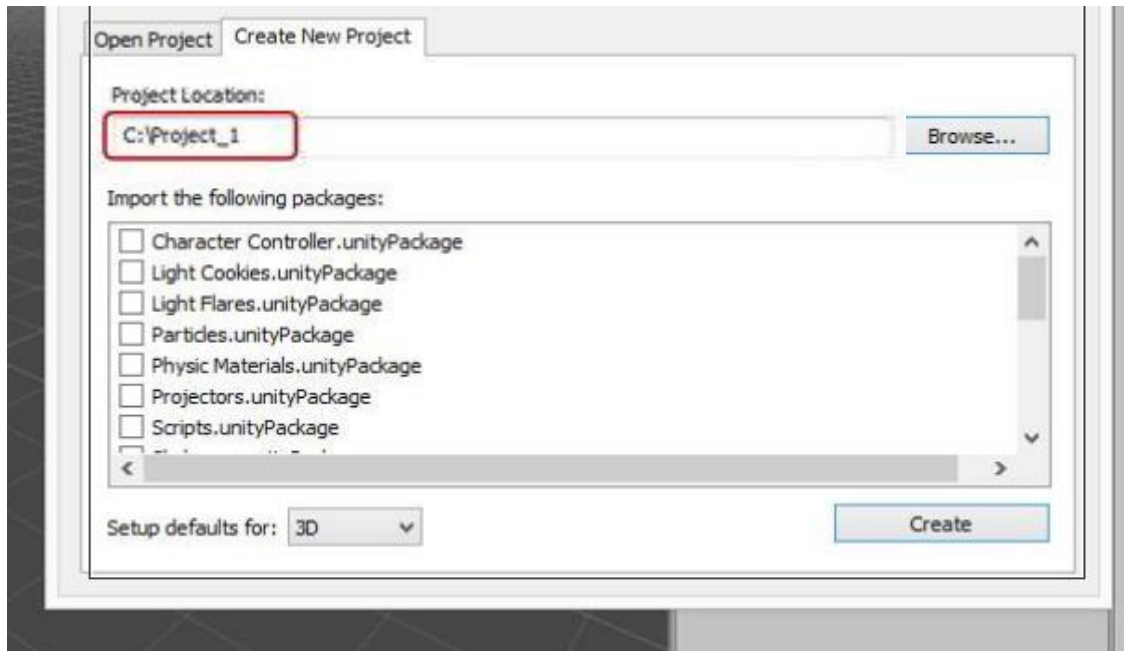
- моделювання — створення тривимірної математичної моделі сцени і об'єктів в ній;
- текстурування — призначення поверхонь моделей растрових або процедурних текстур (настройка властивостей матеріалів — прозорість, відбиття, шорсткість та ін.);
- освітлення — встановлення та налаштування джерел світла;
- анімація (в деяких випадках) — надання руху об'єктів;
- динамічна симуляція (в деяких випадках) — автоматичний розрахунок взаємодії частинок, твердих / м'яких тіл з модельованими силами гравітації, вітру, виштовхування та ін., а також один з одним;
- рендеринг (візуалізація) — побудова проекції відповідно до обраної фізичної моделі





СТВОРЕННЯ ПРОЕКТУ

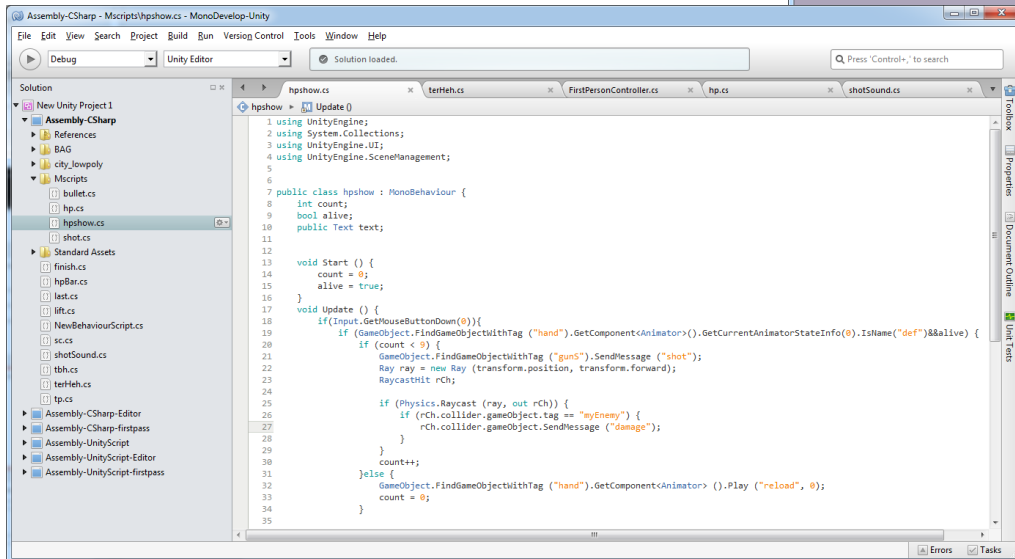
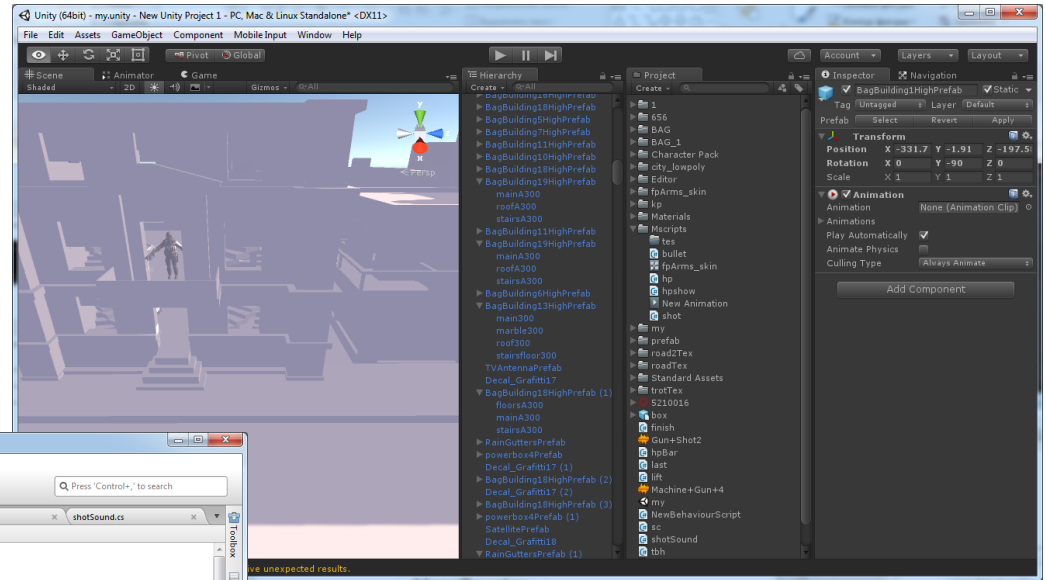
У верхньому куті головного меню обираємо
«**File -- New Project**» (файл – створення проекту).
Далі обираємо місце на жорсткому диску, де має бути
збережений проект, наприклад: **C:/Project_1**





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

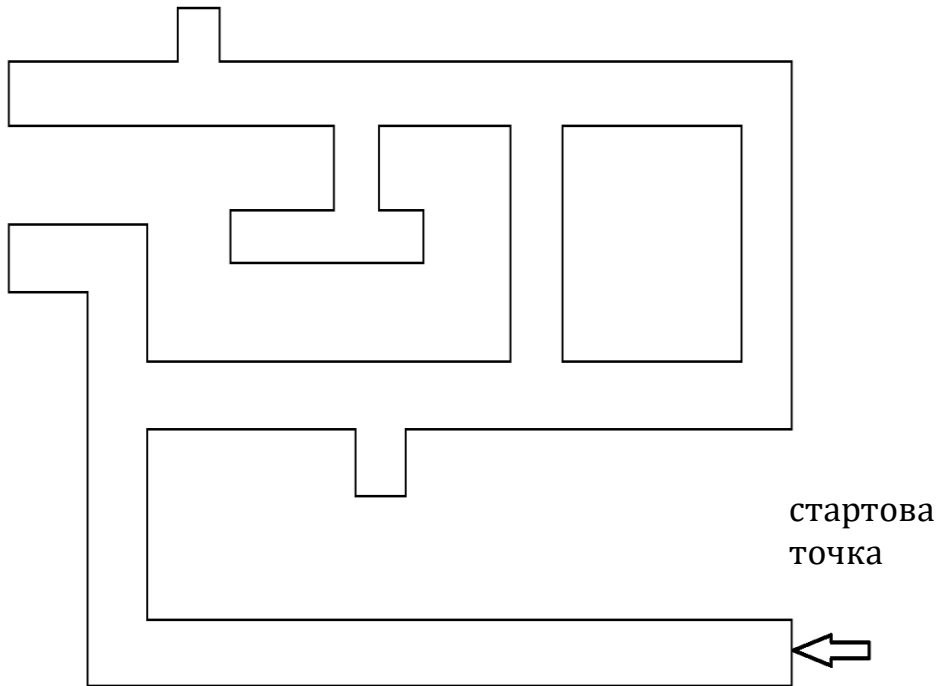
ЗАСОБИ РОЗРОБКИ





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

МОДЕЛЮВАННЯ ІГРОВОГО ПРОСТОРУ



Потрібні такі статичні об'єкти:

- моделі будівель
- моделі дороги і перехресть
- моделі тротуарів, огорож
- модель заднього фону (skybox)





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

МОДЕЛЮВАННЯ ІГРОВОГО ПРОСТОРУ

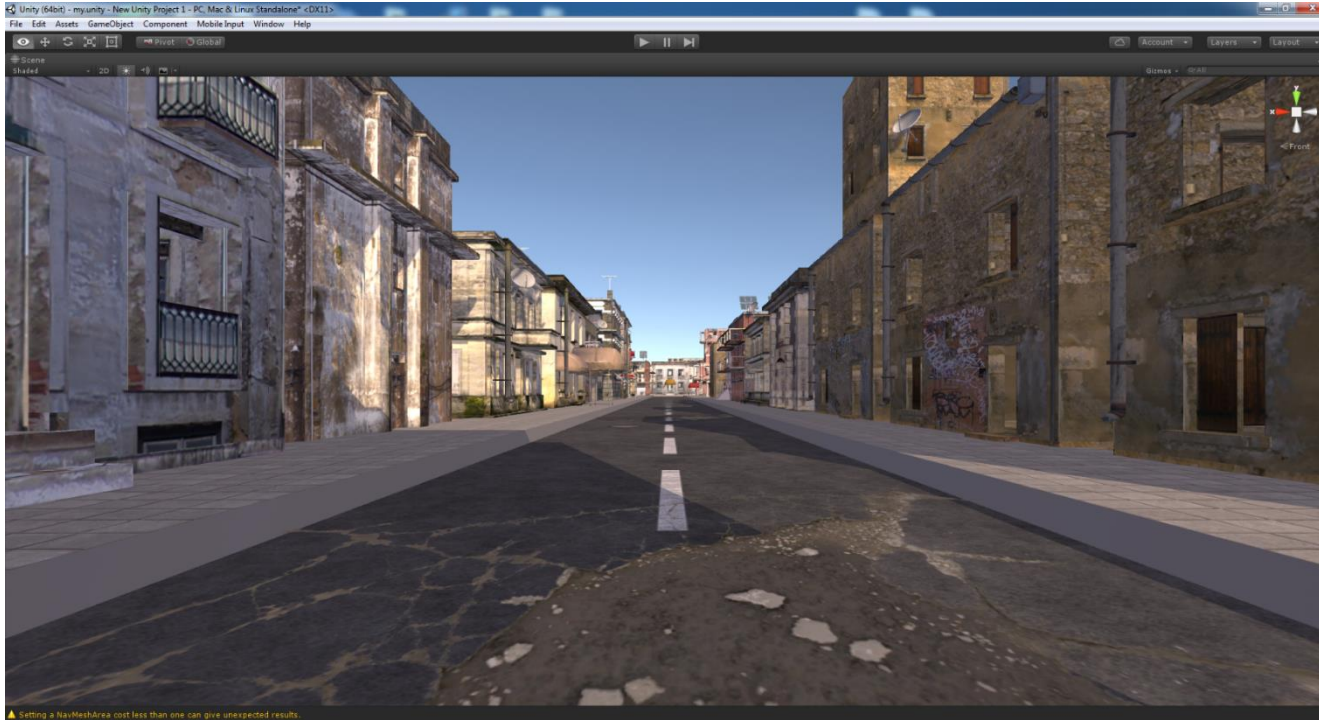
Квартал з текстурами





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

МОДЕЛЮВАННЯ ІГРОВОГО ПРОСТОРУ



Текстури всередині сцени

- покинутий квартал
- моделі невисоких будинків
- літній сонячний день





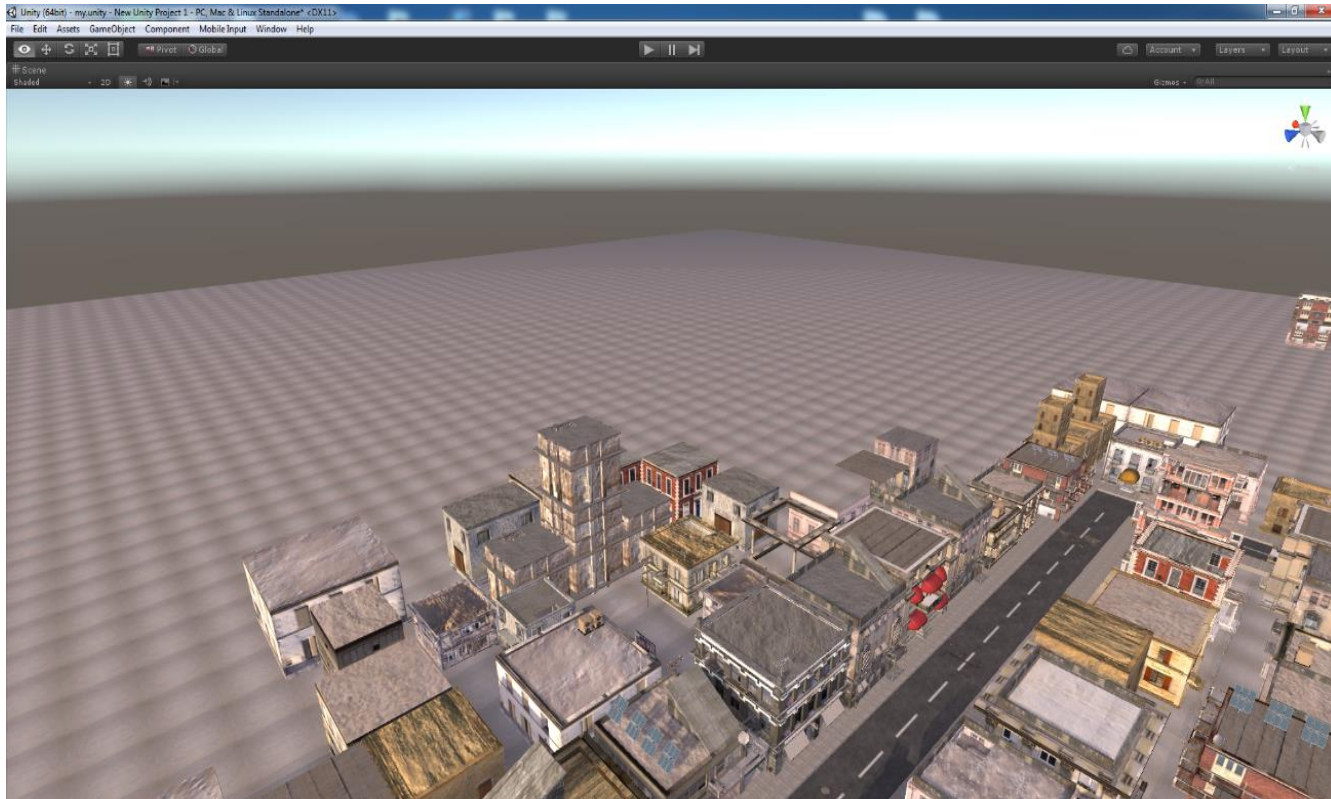
Co-funded by the
Erasmus+ Programme
of the European Union



University-Enterprises Cooperation
In Game Industry In **Ukraine**

МОДЕЛЮВАННЯ ІГРОВОГО ПРОСТОРУ

Текстури за межами корисної області





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

МОДЕЛЮВАННЯ ОБРАЗУ І ПОВЕДІНКИ ГОЛОВНОГО ГЕРОЯ

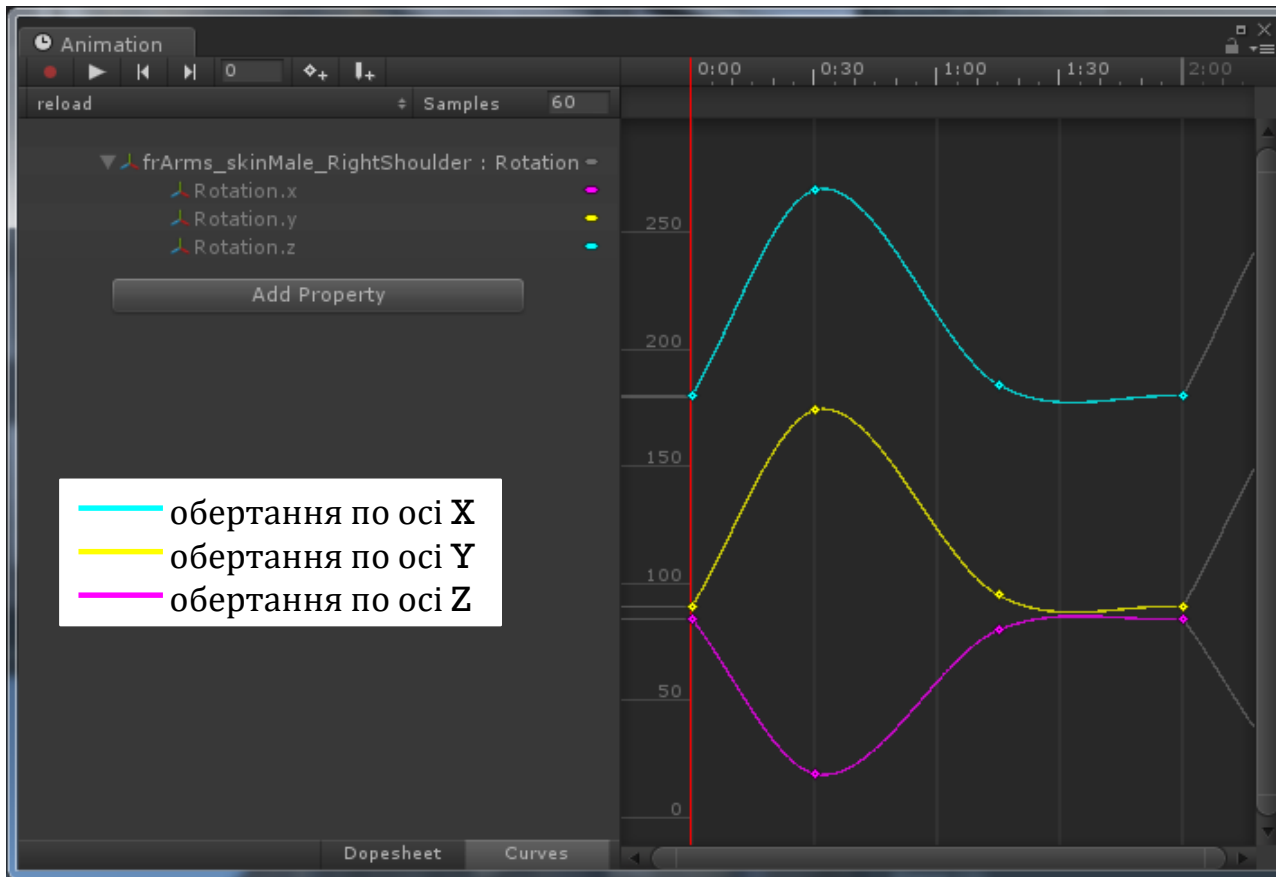
- біг, стрибки
- стрільба
- перезарядка зброї
- облік отриманого урону





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

ПРОЦЕС СТВОРЕННЯ АНІМАЦІЇ ПЕРЕЗАРЯДКИ ЗБРОЇ

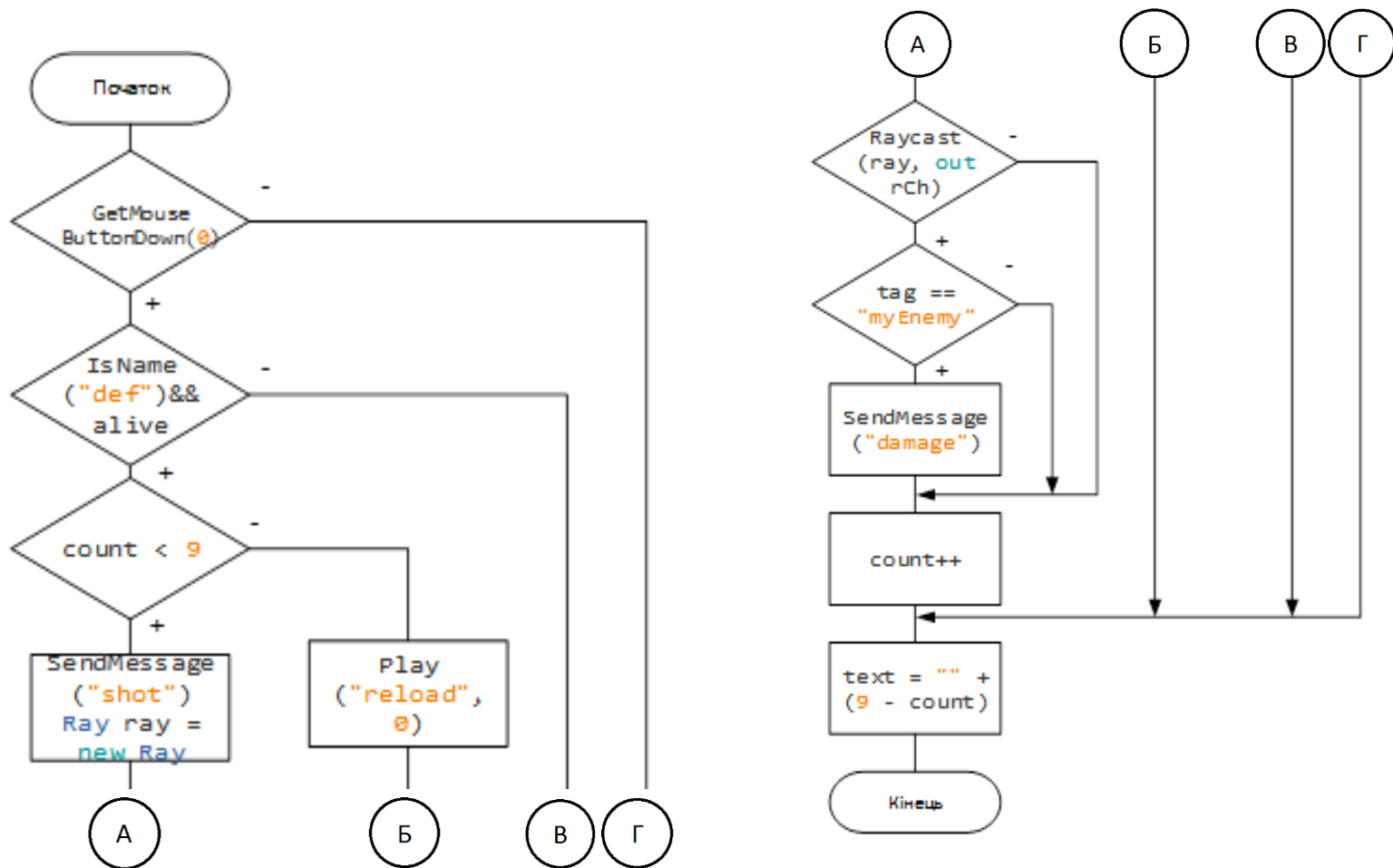


Анімація
перезарядки
застосовується
до руки
персонажа після
того як
кількість
патронів стає
рівним нулю





АЛГОРИТМ ПРОЦЕДУРИ СТРІЛЬБИ ГОЛОВНОГО ГЕРОЯ





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



АНІМАЦІЯ ПЕРЕЗАРЯДКИ ЗБРОЇ ГОЛОВНОГО ГЕРОЯ

за допомогою функції hpShow:

```
if(Input.GetMouseButtonDown(0)){
    if (GameObject.FindGameObjectWithTag ("hand").
        GetComponent<Animator>().
        GetCurrentAnimatorStateInfo(0).
        IsName("def")&&alive) {
        if (count < 9) {
            GameObject.FindGameObjectWithTag ("gunS").
            SendMessage ("shot");

            Ray ray = new Ray (transform.position, transform.forward);
            RaycastHit rCh;

            if (Physics.Raycast (ray, out rCh) {
                if (rCh.collider.gameObject.
                    tag == "myEnemy") {
                    rCh.collider.gameObject.
                    SendMessage ("damage");
                }
            }
            count++;
        }else {
            GameObject.FindGameObjectWithTag ("hand").
            GetComponent<Animator> ().
            Play ("reload", 0);
            count = 0;
        }
    }
}
text.text = "" + (9 - count);
```





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



University-Enterprises Cooperation
In Game Industry In **Ukraine**

МОДЕЛЮВАННЯ ОБРАЗУ ДОДАТКОВОГО ДИНАМІЧНОГО ОБ'ЄКТУ



Доступні анімації:

- ⊕ Біг
- ⊕ Стрільба
- ⊕ Знищення об'єкта

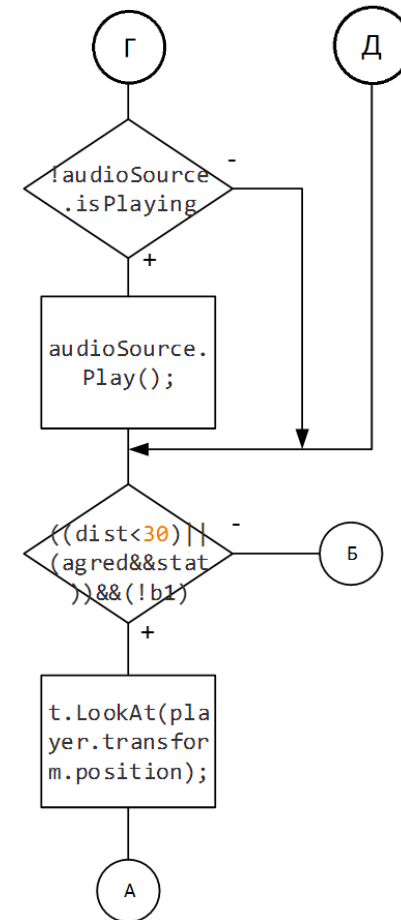
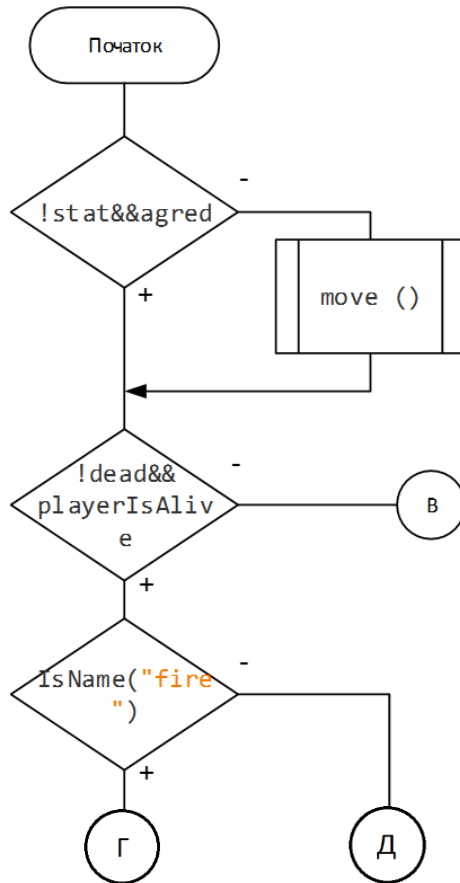
Види активностей:

- ⊕ реакція на дії головного героя
(поява на заданій відстані,
стрільба)
- ⊕ реакція на дії інших динамічний
об'єктів



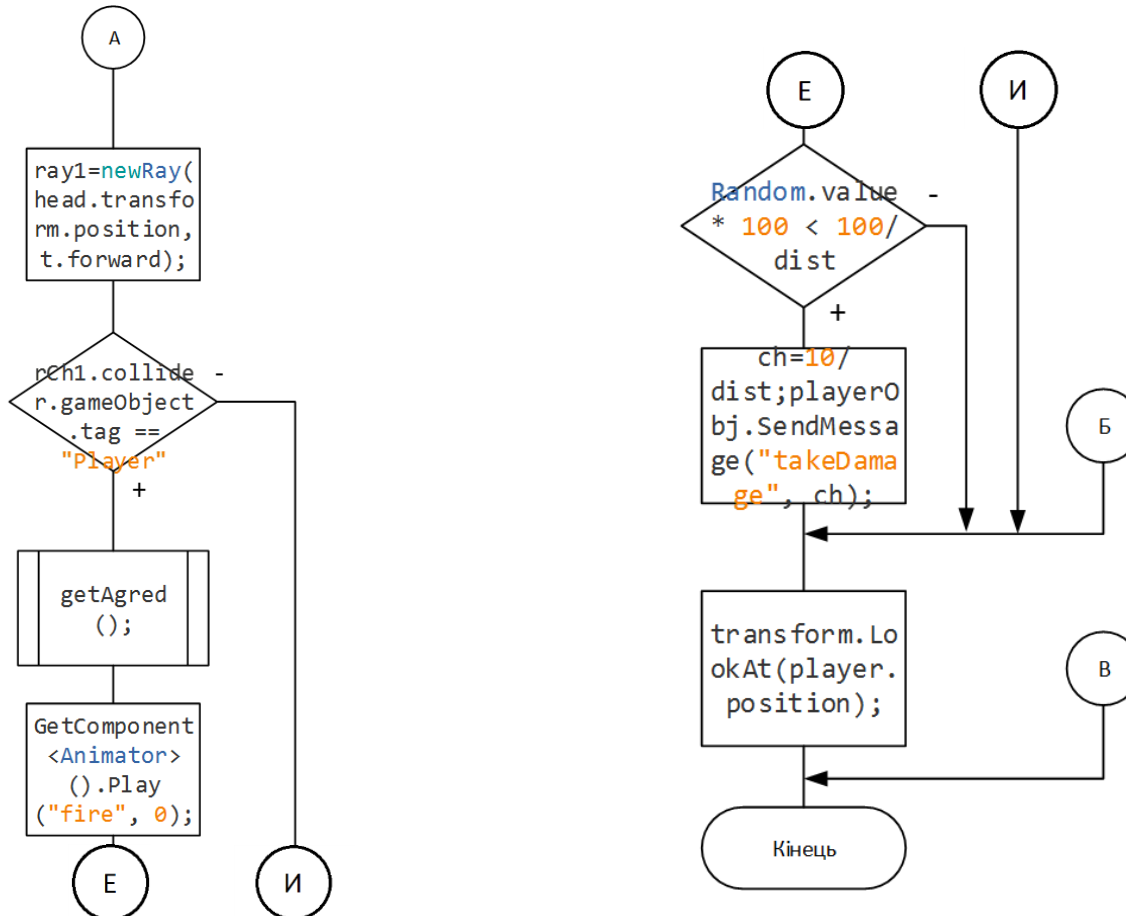


АЛГОРИТМ ПОВЕДІНКИ ДОДАТКОВИХ ДИНАМІЧНИХ ОБ'ЄКТІВ





АЛГОРИТМ ПОВЕДІНКИ ДОДАТКОВИХ ДИНАМІЧНИХ ОБ'ЄКТІВ





РЕАЛІЗАЦІЯ АЛГОРИТМУ ПОВЕДІНКИ ДОДАТКОВИХ ДИНАМІЧНИХ ОБ'ЄКТІВ

Функція `terMeh`:

```
if (!stat&&agred) { move ();  
    }  
if (!dead&&playerIsAlive)  
{  
    if (GetComponent<Animator>().  
        GetCurrentAnimatorStateInfo(0).IsName("fire"))  
        {  
            if (!audioSource.isPlaying)  
                {  
                    audioSource.Play();  
                }  
        }  
    dist=Vector3.Distance (transform.position, player.position);  
    b1 = GetComponent<Animator> ().  
        GetCurrentAnimatorStateInfo (0).  
        IsName ("walkwithweapon");  
    if (((dist < 30) || (agred&&stat)&&!b1))  
    {  
        t.LookAt(player.transform.position);  
        ray1 = new Ray(head.transform.position, t.forward);  
        Physics.Raycast(ray1, out rCh1);  
        // якщо побачив
```

```
        if (rCh1.collider.gameObject.  
            tag == "Player")  
            {  
                getAgred ();  
                GetComponent<Animator> ().  
                Play ("fire", 0);  
                if (Random.value * 100 < 100/dist)  
                    {  
                        ch = 10 / dist;  
                        playerObj.SendMessage ( "takeDamage", ch);  
                    }  
            }  
        transform.LookAt(player.position);  
        transform.Rotate(0, 180, 0);  
    }  
}
```





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



ДЯКУЮ ЗА УВАГУ!

