

Veliki svet



Veoma je važno da igrač ima osećaj velikog prostora u okviru igre

U okviru ovog dela ćemo omogućiti da naš igrač može da izabere preko mape druge mesta koje može posetiti



Tipovi mapa



Postoje dva glavna pristupa u radu sa mapama

1. Fixne mape - U ovom slučaju slike su nacrate od strane umetnika i imaju veliki nivo detalja o svetu koji okružuje igrača, i postepeno otkrivaju durge delove kako igrač istražuje.
2. Generisane mape - U ovoj opciji se na sreću bira mesto za odlazak i događaji koji će se desiti.

Fixne mape

- Dobar izvor besplatnih fixnih mapa imao na adresi <http://freefantasymaps.org/>
- Prilikom korišćenja mapa uvek treba proveriti licencu



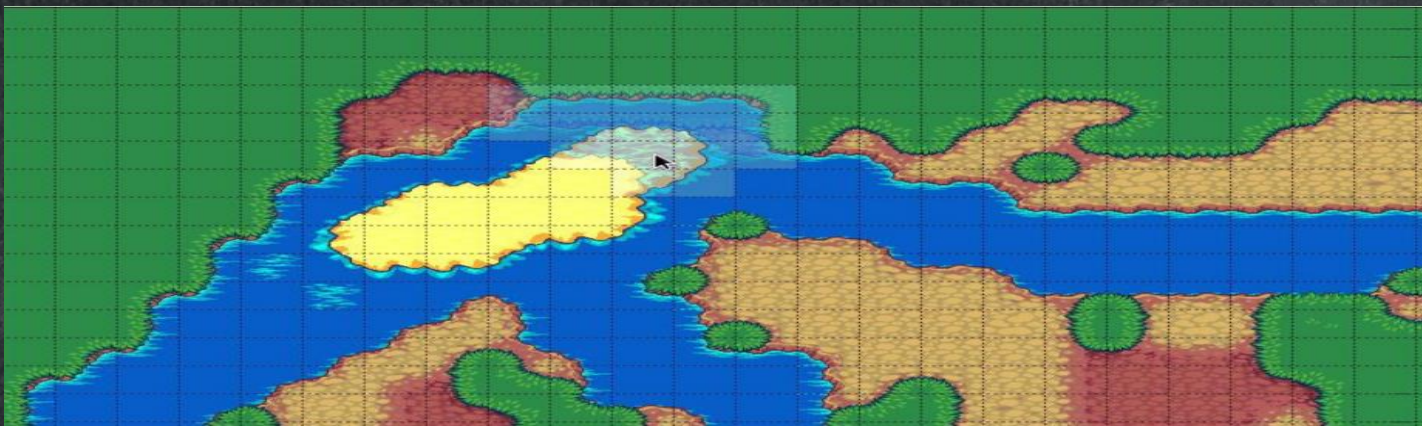
Generisane mape

- Mape možemo generisati na adresi <http://donjon.bin.sh/>



Generisane mape u igri

- Ako želimo veću kontrolu možemo sami praviti mape.
- Externi alati kao što je **Tiled** (<http://www.mapeditor.org>) mogu se koristiti za dizajn
- Projekat uTiled se može pružiti unitiu čitanje Tiled mapa



Prosotor ekrana i prosotor sveta

Kada radimo sa inputom putem miša, ili dodira važno je voditi računa da su koordinate sa kojima Unity radi u prosotru ekrana

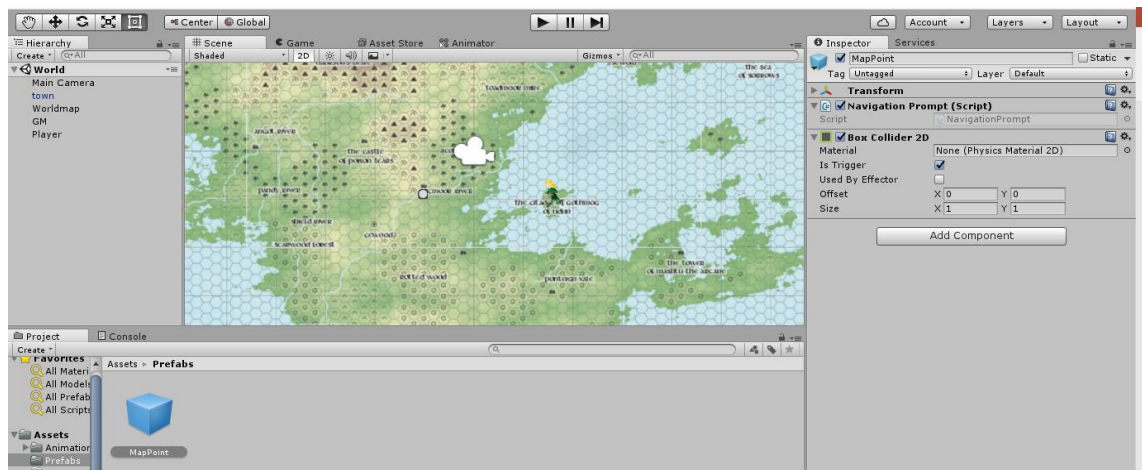
- Prosotor ekrana se odnosi na kordiante relativne u odnosu na ekran, sa početkom u kornjem levom uglu ekrana.
- Prostor sveta se odnosi na kordiante korišćene u okviru unitia
- Kao bismo proverili trenutne kordiante miša možemo koristiti *Input.mousePosition*
- Za dodir možemo koristiti *Input.GetTouch(<touch index>).position*

S obizor da su dobijen koordinate u prostoru ekrana moramo ih prevesti u prostor igre što se može lako uraditi meotdom *Camera.main.ScreenToWorldPoint(<screenCoordinate>)*

Kreiranje mape u igri

Sada kada smo naučili osnove o mapa možemo kreirati našu mapu.

- Prvi korak je kreiranje nove scene u koju ćemo ubaciti mapu
- Na scenu dodajemo:
 - Sliku koja predstavlja našu mapu.
 - Sprite igrača
 - I prazan objekat MapPoint koji će predstavljati neki grad na mapi, od koga pravimo prefab kako bi mogli kreirati više mesta koje ćemo posetiti



U *NavigationManager* skripti je potrebno ubaciti novu rutu
Potrebno je prikazati skriptu *NavigationPrompt* na objekat na mapi

Kreiramo klasu sa proizenim metodama koje nam omogućavaju da uzmemo realne koordinate gde je kliknuto na mapi

```
// Konvertuje Vektor3 u 2D Vektor3
public static Vector3 ToVector3_2D(this Vector3 coordinate)
{
    return new Vector3(coordinate.x, coordinate.y, 0);
}
// Konvertuje koordinate prosotra ekrana u Vektor3 sa vrerdnostima za 2D
public static Vector3 GetScreenPositionFor2D(this Vector3 screenCoordinate)
{
    Vector3 wp = Camera.main.ScreenToWorldPoint(screenCoordinate);
    Return wp.ToVector3_2D();
}
```

This označava da je ovo produžena metoda i omogućava da se pozove normlano:
var clickPoint = WorldExtensions.GetScreenPositionFor2D(Input.mousePosition);



Kretanje igrača po mapi

Kako bi omogućili igraču da se kreće po mapi kreiramo skriptu *MapMovement* U kojoj proveramo uz pomoć metode *Input.GetMouseButtonUp(0)* da li je kliknuto dugme na mišu, i uzimao koordinate gde je kliknuto uz pomoć metode *Input.mousePosition*

```
if (Input.GetMouseButtonUp(0))
{
    StartLocation = transform.position.ToVector3_2D();
    timer = 0;
    TargetLocation = WorldExtension.GetScreenPositionFor2D(Input.mousePosition);
    Ready = true;
}
```

I jednostavno pomeramo igrača uz pomoć

```
if (TargetLocation != Vector3.zero && TargetLocation != transform.position && TargetLocation != StartLocation && Ready)
{
    transform.position = Vector3.Lerp(StartLocation, TargetLocation,
                                     timer);

    timer += Time.deltaTime;
}
```



Kretanje igrača po mapi

Kako bi omogućili igraču da se kreće po mapi kreiramo skriptu *MapMovement* U kojoj proveramo uz pomoć metode *Input.GetMouseButtonUp(0)* da li je kliknuto dugme na mišu, i uzimao koordinate gde je kliknuto uz pomoć metode *Input.mousePosition*

```
if (Input.GetMouseButtonUp(0))
{
    StartLocation = transform.position.ToVector3_2D();
    timer = 0;
    TargetLocation = WorldExtension.GetScreenPositionFor2D(Input.mousePosition);
    Ready = true;
}
```

I jednostavno pomeramo igrača uz pomoć

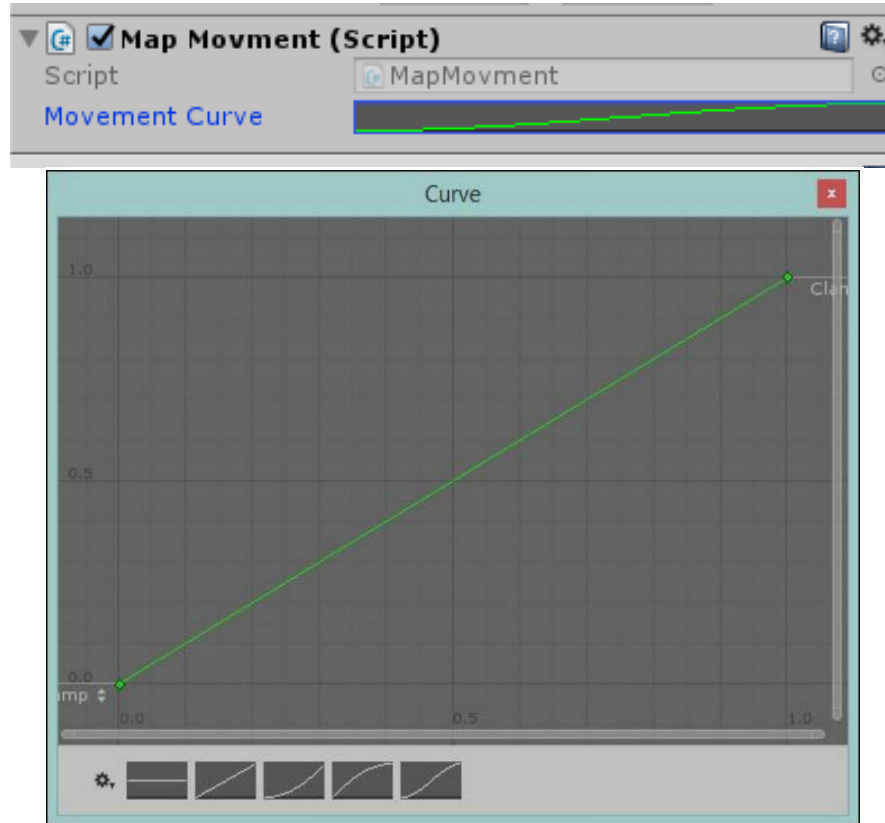
```
if (TargetLocation != Vector3.zero && TargetLocation != transform.position && TargetLocation != StartLocation && Ready)
{
    transform.position = Vector3.Lerp(StartLocation, TargetLocation,
                                     timer);

    timer += Time.deltaTime;
}
```



Animacione krive mogu promeniti kako se sprite menja tokom vremena. Tako da ćemo ih ubaciti u naš *MapMovement* za kretanje igrača na mapi:

```
public AnimationCurve MovementCurve;
```



Postepena tranzicija izmedju nivoa

- Prvo postavljamo prazan objekat „GM“ za koji kačimo skriptu „FadeInOut“ koja nam omogućava postepenu tranziciju izmedju scena
- Metoda OnGUI koja iscrtava teksturu koja ce biti prikazan izmedju scena (Tekstura mozte biti samo crna pozdina)

```
void OnGUI()
```

```
{
```

```
// fade na osnovu smeru, brzine i brzine promene frejma
```

```
alpha += fadeDir * fadeSpeed * Time.deltaTime;
```

```
// setovanje alfa vrednosti na silu izmedju 0 i 1
```

```
alpha = Mathf.Clamp01(alpha);
```

```
// podesava boju GUI-a (u nasem slucaju teksture) sve vrednosti ostaju i ste samo se alfa setuje na alfa vrednost
```

```
GUI.color = new Color(GUI.color.r, GUI.color.g, GUI.color.b, alpha); // setovanje alfa vrednosti
```

```
GUI.depth = drawDepth; // crdna tekstura na vrhu
```

```
GUI.DrawTexture(new Rect(0, 0, Screen.width, Screen.height), fadeOutTexture); // postaviti teksturu preko celog ekrana
```



```
// poziva se nakon učitavanja nove scene  
void OnLevelWasLoaded(int level)  
{  
    // poziva metodu koja zapocinje traniziciju  
    BeginFade(-1);  
}
```

```
// postavlja smer tranizicije izmedju scena  
public float BeginFade(int direction)  
{  
    fadeDir = direction;  
    return fadeSpeed;  
}
```



- U okviru skripte *NavigationPromt* potrebno je postaviti korutinu koja će se aktivirati prilikom promene scena i izvršiti tranziciju

```
IEnumerator ChangeLevel()  
{  
// pronalzi objekat za koji je zakacena tekstura i skripta za tranziciju izmedju s  
cena  
    float fadeTime = GameObject.Find("GM").GetComponent<FadeInOutMan  
ager>().BeginFade(1);  
    yield return new WaitForSeconds(fadeTime);  
// promena scene  
    UnityEngine.SceneManagement.SceneManager.LoadScene(level);  
}
```

