«Talento Tech»

Desarrollo de Videojuegos

Unity 2D

Clase 10

«Talento Tech»

Buenos Aires Ciudad

Clase N° 10 | NavMesh

Temario:

- Navmesh
- PathFinding

Navmesh.

En Unity, un NavMesh (o Malla de Navegación) es una representación del espacio de navegación accesible en una escena, utilizada principalmente para que los “Agents” en un juego puedan moverse de manera autónoma evitando obstáculos.

- **Malla de Navegación:** Es una malla que Unity genera automáticamente, cubriendo las áreas donde los personajes pueden caminar. Cubre únicamente las superficies accesibles, excluyendo las zonas bloqueadas por obstáculos.
- **Agentes de Navegación:** Los agentes son los objetos o personajes en tu juego que se moverán usando el NavMesh. Cada agente puede ser configurado con características específicas, como su velocidad y tamaño, para que se adapte al entorno.
- **Caminos y Obstáculos:** El NavMesh permite que los agentes encuentren el camino más corto hacia su objetivo, evitando colisiones con obstáculos o paredes. Los obstáculos se pueden configurar como objetos estáticos o dinámicos, afectando en tiempo real el recorrido de los agentes.
- **NavMesh Baking:** Es el proceso de generar o “hornear” la malla de navegación. Unity examina los objetos de la escena, detecta superficies y obstáculos y crea la malla de navegación en función de esos datos. Este proceso se realiza generalmente antes de la ejecución del juego y se ajusta si el entorno cambia en tiempo real.

El NavMesh es fundamental para juegos que requieren una IA con movimiento autónomo, como juegos de estrategia, de rol o de acción, donde los personajes deben poder moverse de manera inteligente por el entorno del juego.

NavMesh 2D.

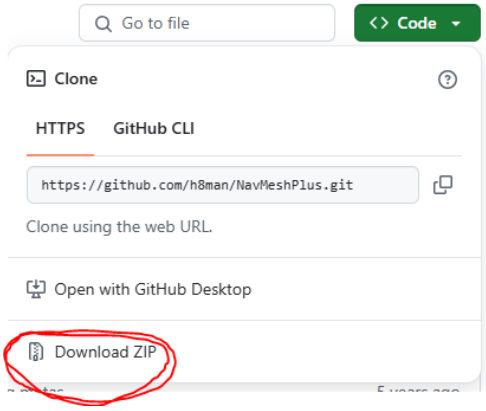
Uno de los problemas con los que nos encontramos es que NavMesh, principalmente está diseñado para la navegación en 3D, por lo tanto, si queremos usarlo en 2D, tenemos que descargar algunos repositorios que nos serán de ayuda.

Descarga.

Pueden descargarlo de aca:

https://github.com/h8man/NavMeshPlus

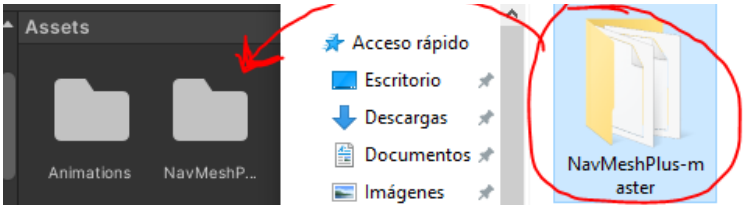
Haciendo Click en “code” y “Download Zip”



O yendo a este link:

https://drive.google.com/file/d/1Oz583-YJte6Nu6yuB8_PBHCyBLO3wNgc/view?usp=sharing

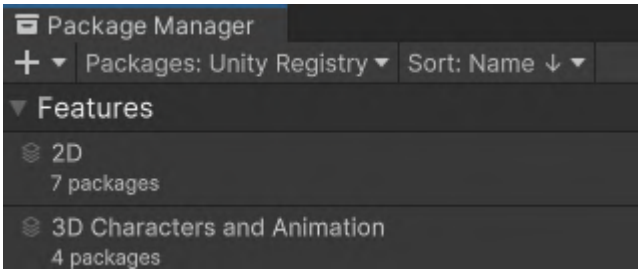
Al descargarlo, deben descomprimirlo y pasar la carpeta resultante “NavMeshPlus- Master” a la carpeta Assets de Unity:



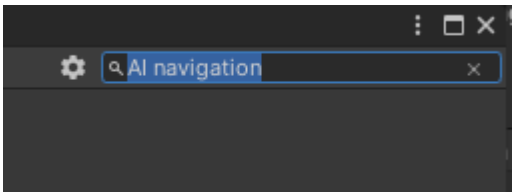
Navigation IA.

También tendremos que descargar el Package de “AI Navigation”, yendo a “**Window->Package Manager**”

Asegúrense de colocar “Unity registry”



Y en el buscador colocar “AI Navigation”

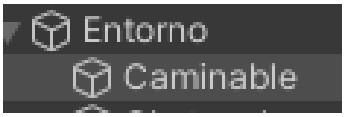


Atención: *Si llega a pasar que no les aparece nada, es posible que ya lo hayan descargado.*

Seteo.

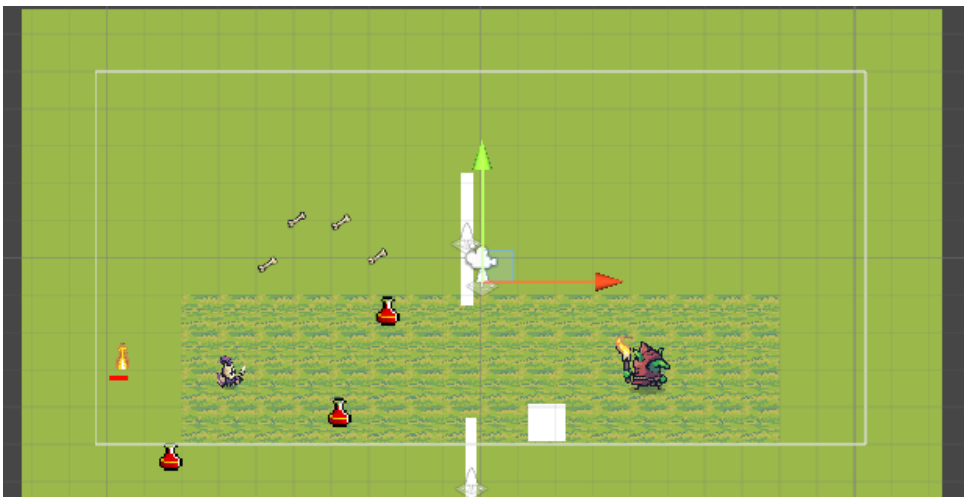
Para empezar a utilizar nuestra nueva IA debemos crear varios objetos.

Caminable.

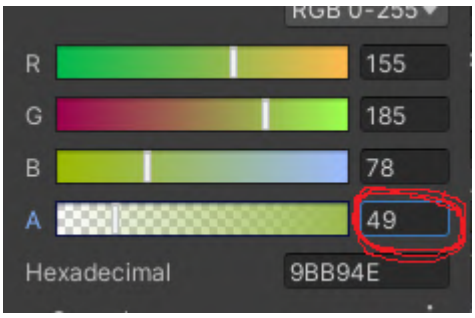


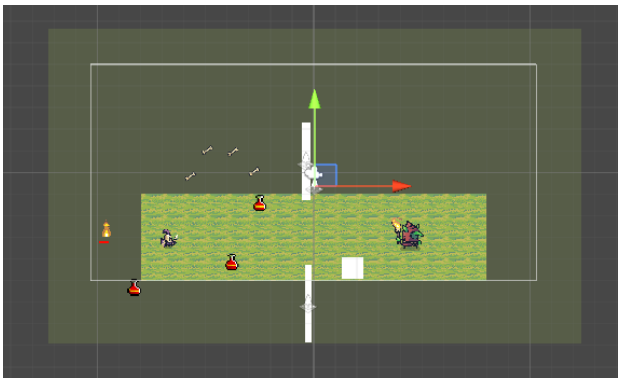
Crearemos un Empty Object llamado “Entorno” y le pondremos un “Square” dentro, con el nombre de “caminable”.

Haremos que este cubra un tamaño considerable, porque será la base por donde nuestro NPC pueda moverse.

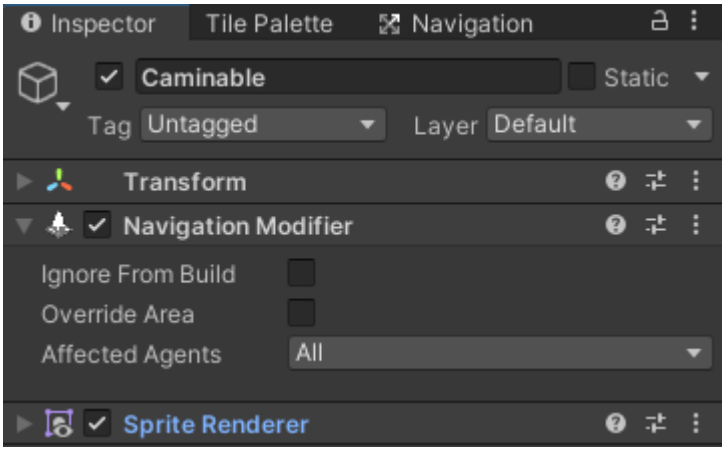


Le reduciremos el Alpha para que sea medio transparente y no nos moleste tanto.



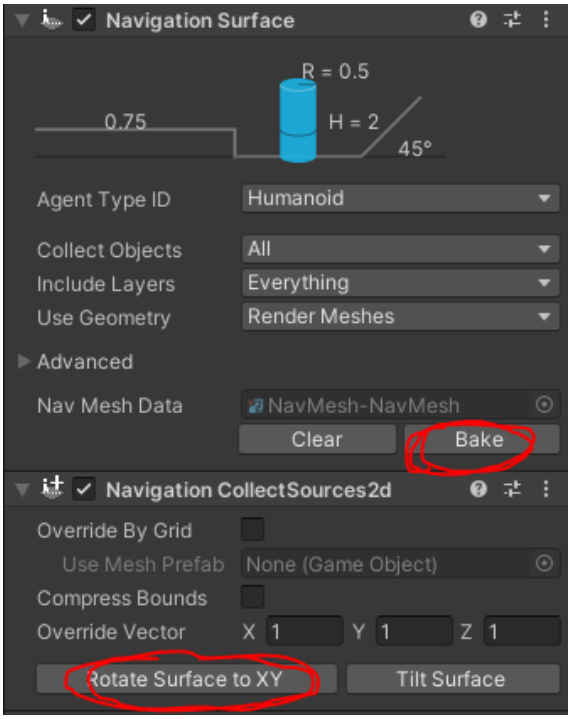


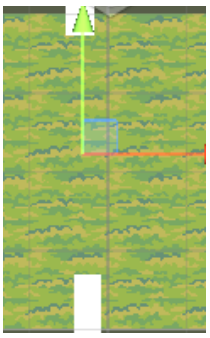
Una vez hecho esto le agregaremos el componente “**Navigation Modifier**”. Siendo un componente nuevo, parte del package que descargamos externamente.

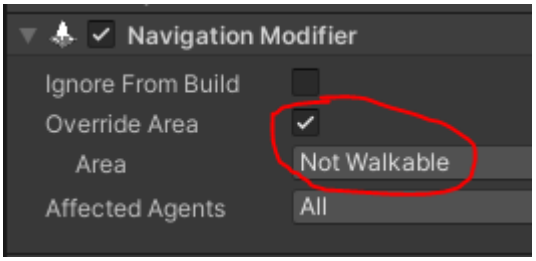


Obstáculo.

Para generar los obstáculos, por ahora, crearemos un simple “Square” y lo acomodaremos como más nos guste y le pondremos el **Component** “Navigation Modifier”. Asegurándonos de Chequear/tildar el “Override area” (SobreEscribir Area) y colocar la opción de “Not Walkable” (No caminable), que le otorgara el comportamiento de obstáculo.





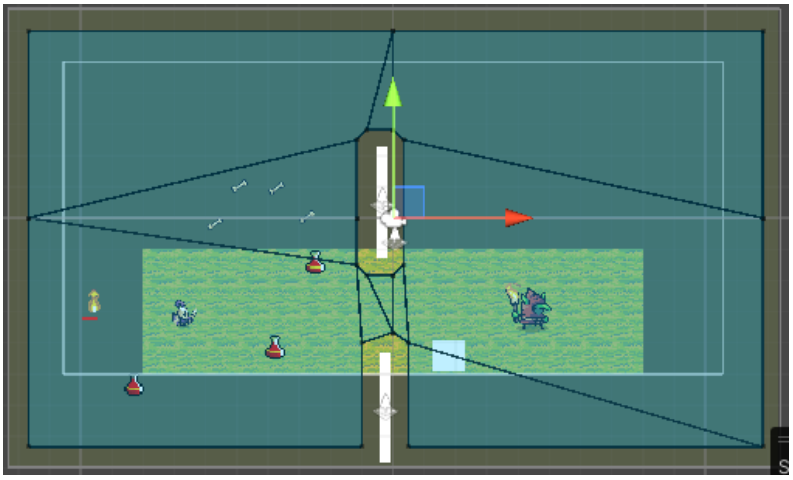


EmptyObject: NavMesh.

Pasaremos a crear otro EmptyObject, lo nombraremos “NavMesh” y le pondremos los componentes “Navigation Surface” y “Navigation CollectorSources2D”

Le daremos Click a “**Rotate Surface to XY**” que “acomodora la superficie” y luego click en “**Bake**”.

Verán como el GameObject “Caminable” se pintó de un celeste, pero evitando los obstáculos creados.



Significa que nuestro NPC tiene TODO ese espacio para moverse, pero **no podrá** avanzar por los obstáculos creados.

Enemy.

Código.

Lo último que necesitamos es setear a nuestro enemigo. Empezaremos brindándole un comportamiento:

```
public class EnemyNav : MonoBehaviour
{
[SerializeField] Transform target;
NavMeshAgent agent;


void Start()
{
    agent = GetComponent<NavMeshAgent>();
    agent.updateRotation = false;
    agent.updateUpAxis = false;

}

void Update()
{
    agent.SetDestination(target.position);
```

```
    }
}
```

Explicación del código.

Definimos las 2 variables a usar

```
    [SerializeField] Transform target;
    NavMeshAgent agent;
```

El “target” será su objetivo y la variable de tipo “NavMeshAgent” es la que nos permitirá usar las herramientas del NavMesh.

En el Start() le asignaremos al valor a “Agent” y pondremos en “false” las opción de actualizar Rotación y el UpAxis. Esto será para que el NavMeshAgent, NO gire nuestro Sprite (Obviamente dependerá de la circunstancia).

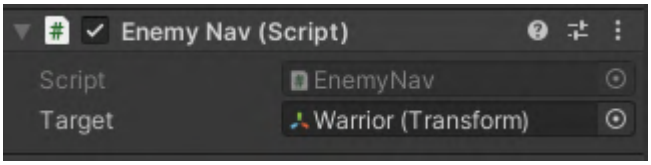
```
    void Start()
    {
        agent = GetComponent<NavMeshAgent>();
        agent.updateRotation = false;
        agent.updateUpAxis = false;

    }
```

En la función Update() le diremos que se mueva al target, usando una funcion propia del NavMeshAgent

```
    void Update()
    {
        agent.SetDestination(target.position);
    }
```

GameObject.

Todo este código lo pondremos en nuestro “Enemigo”. Le **asignaremos el target** por medio del **Inspector**



En mi caso el GameObject del Player se llama “Warrior”.

Le agregaremos otro Component:
El “NavMeshAgent”, para que el código lo utilice.

¡Listo!, al darle Play, veremos que el NPC estará constantemente siguiendo a nuestro personaje.

Ejercicios prácticos:

NavMesh.

Notaran, que si apenas empezamos el enemigo ya nos está persiguiendo, nos puede arruinar un poco el juego. Así que la actividad de hoy es colocar un límite de distancia al comportamiento de “perseguir a nuestro personaje”.

TIPS:

1) Recuerden que ya lo hicimos en la clase anterior.

2) Si desean que el NPC se detenga al perder el rango pueden usar:

```
agent.isStopped = true;
// Recuerden ponerlo en False nuevamente al querer moverlo, sino se quedará por siempre ESTÁTICO.
```


Buenos Aires
aprende
Agencia de Habilidades para el Futuro
Buenos
Aires
Ciudad