

Лабораторна робота №11

Система навігації Unity3D

Мета: : Ознайомитися з процесом створення навігаційної сітки. Розглянути основи використання NavMesh. Навчитися запікати NavMesh і розбиратися в його компонентах як NavMesh Obstacle, Off Mesh Link і NavMesh.

Література

AI Navigation (NavMesh) — самостійний пошук шляху до цілі в Unity:
<https://gamedev.dou.ua/forums/topic/46391/>

Unity-Technologies/NavMeshComponents: <https://github.com/Unity-Technologies/NavMeshComponents>

Creating a NavMesh Agent: <https://docs.unity.cn/cn/current/Manual/Navigation.html>

Зміст роботи

Завдання 1. Створити цільові навігаційні карти в ігровому світі, який був створений на лабораторній роботі №6.

Завдання 2. Налаштувати дії головного героя. Що має робити головний герой:

- Рухатися до антагоніста, оминаючи перешкоди(статичні та динамічні).
- Атакувати антагоніста, коли він буде у зоні досягнення.
- Відійти на безпечну відстань, в разі необхідності.
- Отримати поранення після сутички з антагоністом. В разі смерті закінчити гру.

Налаштуйте дії антагоніста. Що має робити антагоніст:

- Рухатися до головного героя, оминаючи перешкоди.
- Атакувати головного героя, коли він буде у зоні досягнення.
- Отримати поранення після сутички з головним героєм, зникнути в разі смерті.

Завдання 3. Передбачити додаткові ігрові об'єкти, які рухатимуться навігаційною сіткою за власним маршрутом.

Додатково можна отримати 5 балів якщо зробити усе вищезгадане зі звуками.

Методичні рекомендації

Що таке NavMesh у Unity? Система навігації NavMesh дозволяє пояснити ігровим персонажам, як дістатися певної точки рівня, уникаючи всілякі перешкоди та використовуючи створені гравцем механіки.

Для роботи з нею Unity пропонує наступні чотири компоненти:

- NavMesh. Основний компонент, який є структурою даних, що описує всі поверхні ігрового рівня, для якого буде створена навігаційна сітка.
- Звертаючись до неї персонажі можуть прорахувати свій шлях до певної точки.
- NavMeshAgent. Це компонент, який закріплюється за ігровими об'єктами, які мають згодом пересуватися за рівнем. Такі ігрові об'єкти називають агентами.
- Off-mesh Link. Цей компонент необхідний для створення особливих шляхів лише на рівні. Підйом сходами, вхід у портал, стрибок з платформи.
- NavMesh Obstacle. Переміщення, які схильні до фізики або можуть змінювати своє положення будь-яким іншим чином повинні мати даний компонент для того, щоб агенти знали про їх існування і могли їх обійти.

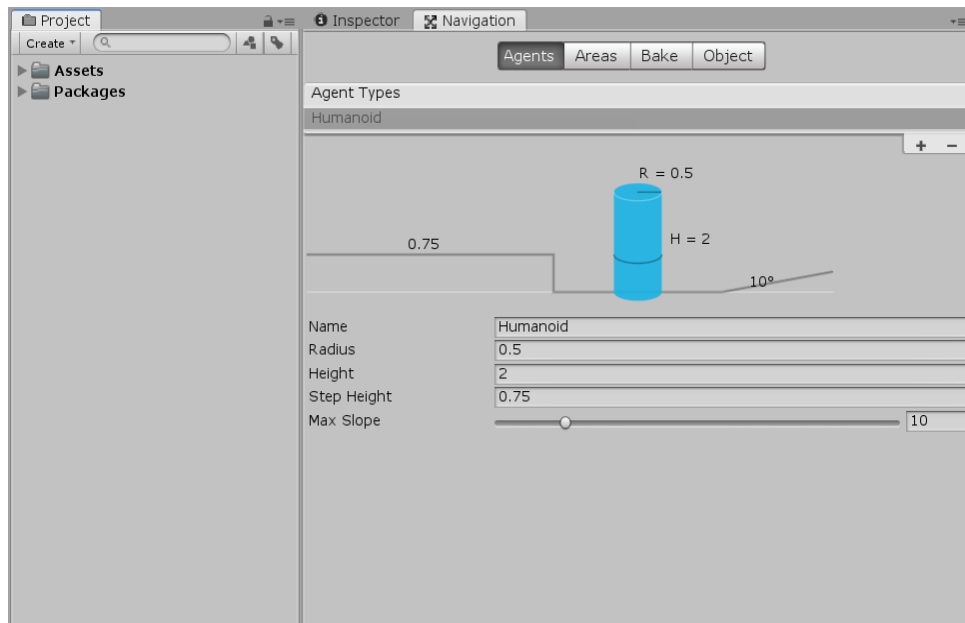
У Unity агенти описуються як циліндри, а поверхні, якими вони можуть рухатися, як опуклих багатокутників. Для пошуку шляху між двома точками в сцені, спочатку зіставляється початкова та кінцева точки з найближчими до них полігонами. Далі, відвідуючи всі сусідні полігони, шукається шлях, що дозволяє досягти пункту призначення. У Unity реалізації даних дій використовується поширений алгоритм пошуку шляху A^* .

Послідовність полігонів, яка описує шлях до кінцевої точки, називається коридором.

Принцип роботи NavMesh

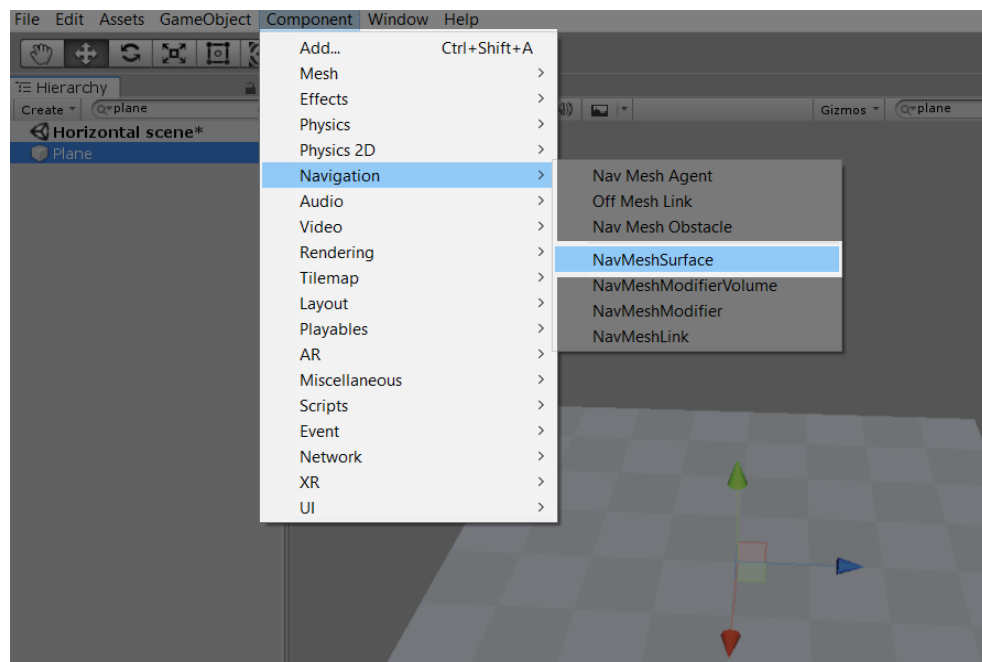
Насамперед, для пошуку шляхів і створення карт "навігацій", сцена має бути спеціально підготовлена. У 2D-іграх використовується двовимірна сітка, простіше кажучи, масив точок, по яких персонаж може переміщатися. Для 3D-проектів необхідно створювати спеціальну окрему сітку, яка повинна враховувати, крім довжини і ширини сцени, ще й її висоту.

Unity має стандартний набір необхідних інструментів для створення карт "навігацій", всі вони знаходяться в розділі Navigation.



Ці стандартні інструменти дають змогу створювати найпростіші карти на будь-якій горизонтальній поверхні.

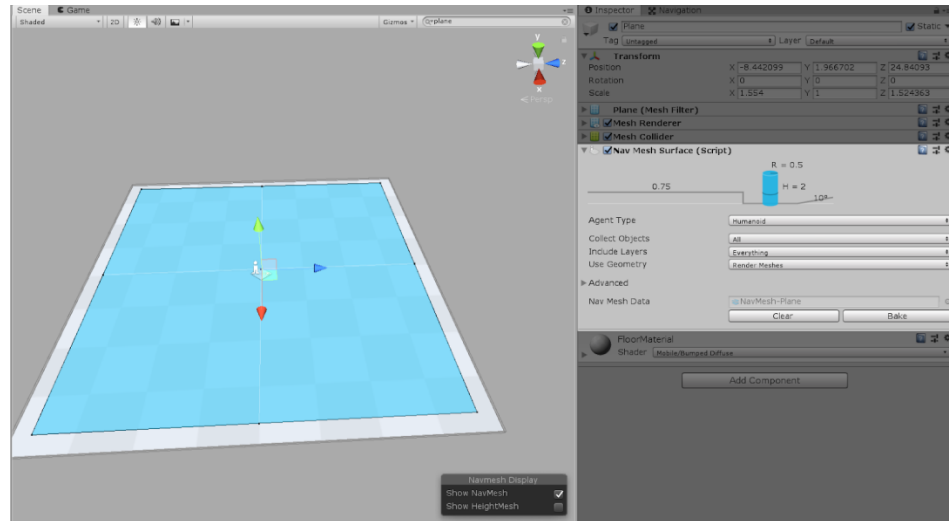
Після того як створили поверхню на сцені, необхідно додати їй компонент NavMeshSurface.



З усього набору доступних параметрів компонента NavMeshSurface скористаємося Clear і Bake, які знаходяться в самому низу переліку налаштувань.

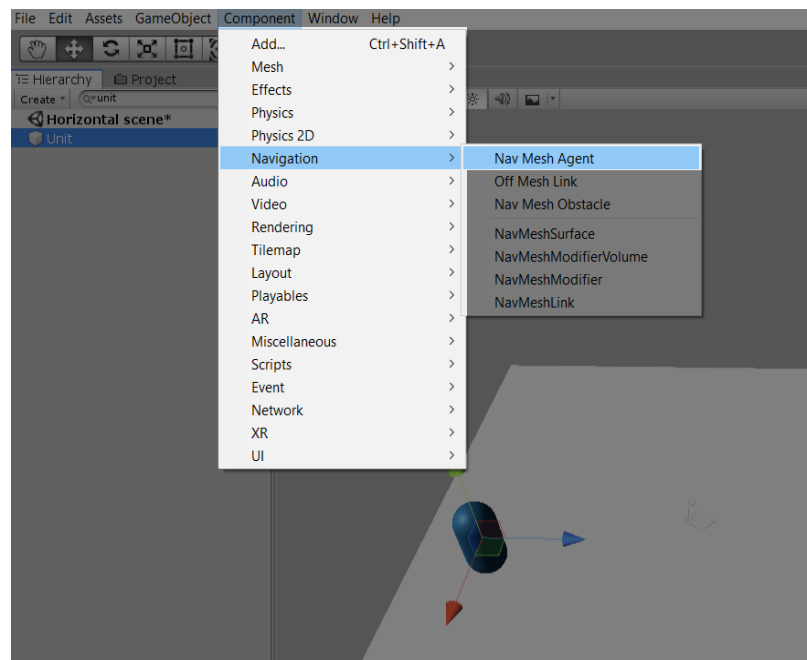
Wake дає змогу підготувати або "запекти" карту на поверхнях, з урахуванням розміру самої поверхні та характеристик об'єкта, який буде переміщатися по цій поверхні.

Натискаємо кнопку Bake і дивимося на результат, який вийшов після "запекання".

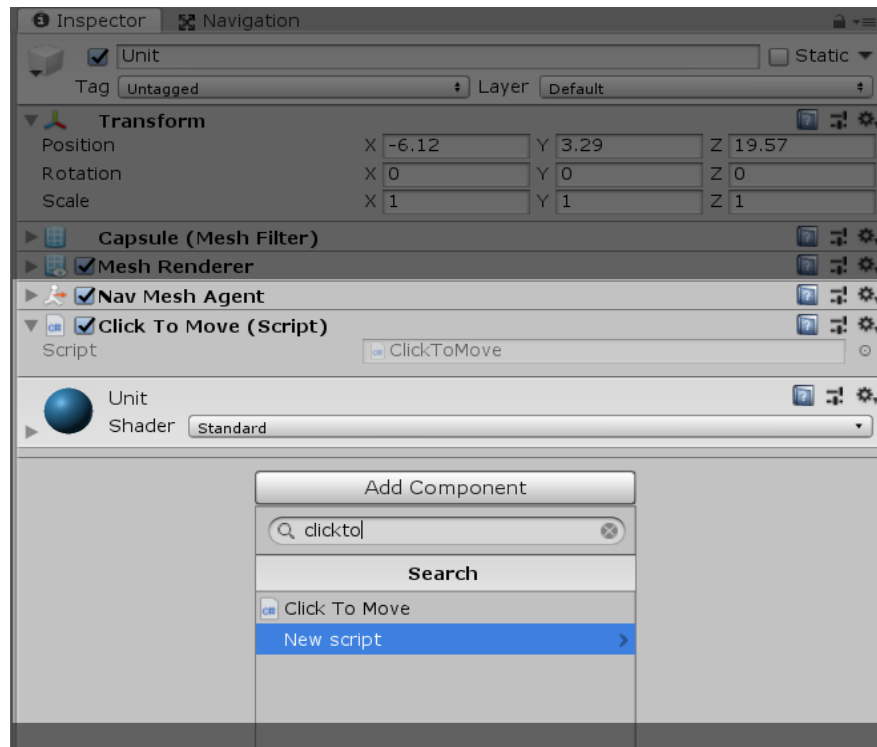


Готова карта відображається як тонка напівпрозора синя поверхня поверх об'єкту.

Поверхня готова, щоб по ній могли переміщатися юніти. Для цього можна додати будь-який об'єкт, наприклад, скористаємося звичайним 3D Capsule після чого відразу додаємо йому компонент NavMeshAgent.



Компонент NavMeshAgent необхідний, щоб знаходити шляхи за готовими картами "навігації". Також компонент має стандартні налаштування, щоб переміщати юніт знайденими шляхами на карті. Тому налаштування компонента NavMeshAgent можна залишити без зміни. Далі, щоб можна було вказувати точку, куди необхідно відправити юніт, додамо йому ще один компонент ClickToMove.



Компонент ClickToMove дасть змогу гравцеві мишкою вказувати точку, куди слід рухатися юніту. Після всіх цих дій можна запустити проєкт і перевірити роботу компонентів.

Після додавання перешкоди на поверхню, потрібно її знову "запекти" за допомогою компонента NavMeshSurface.

Сам процес створення NavMesh з геометрії рівня Unity називається запіканням. Протягом нього всі Render Meshes та Terrains позначені на сцені як Navigation Static збираються разом та обробляються для створення навігаційної сітки. Для того, щоб запекти рівень, потрібно відкрити вікно навігації.

Контрольні питання:

1. Що таке NavMesh?
2. Для чого створюється навігаційна сітка?
3. Для чого використовується NavMeshAgen?
4. Для чого використовується Off-mesh Link?

5. Для чого використовується NavMesh Obstacle?
6. Що таке запікання? Коли і як воно здійснюється?
7. Які об'єкти називаються Navigation Static?

Самостійна робота:

Доопрацювати лабораторну роботу.