1. Podczas obracania platformą za pomocą myszki widzimy, że platforma mimowolnie się przesuwa/odchyla i zostaje w takiej pozycji podczas obracania w osi Z. Spróbujemy teraz naprawić ten problem.
2. Node AddActorWorldRotation był przydatny do tego co dotychczas planowaliśmy, jednak teraz należy go usunąć i dodać nowe node'y które pozwolą na sprawowanie większej kontroli nad obrotem danego aktora.
3. W tym celu będzie potrzebny node o nazwie GetActorRotation (Utilities ->Transformation)
4. Przed dodaniem tego node'a posprzątaj cały graf z niepotrzebnej funkcjonalności - tzn. wyświetlania napisów, poruszania platformą za pomocą klawiatury itp.
5. Tak aby zostały tylko node'y potrzebne do:
   * pobierania pozycji myszki w osi X, Y.
   * do odwracania wartości osi Y.
   * do przetrzymywania referencji do danego obiektu.
   * wywoływania danego wydarzenia co klatkę.
6. Node GetActorRotation będzie zwracał wartość rotacji (stopnie) jakie zastosowano/podano dla danego aktora w osiach XYZ. A więc ten node powinien być połączony z node'm Floor.
7. Aby wartości były zwracane osobno dla danej osi możemy to zrobić na dwa sposoby:
   * Możemy kliknąć prawym przyciskiem myszy w pole Return Value i wybrać znowu opcje Split Struct Pin lub
   * Możemy przeciągnąć/dodać nowy node o nazwie Break Rotator (efekt w obu przydatkach będzie ten sam).
8. Po pobraniu aktualnych wartości od naszego aktora, warto byłoby również móc je ustawić/nadać.
9. Do tego celu posłuży nam node/funkcja SetActorRotation (Utilities -> Transformation)
10. Sam node wiele się nie różni od funkcji AddActorWorldRotation - mamy opcje Teleport Physics, którą również powinniśmy zaznaczyć. Jest również pole New Rotation, któremu można przekazywać/nadać wartości, aby obracać danym aktorem.
11. Tym razem jednak zamiast rozbijać opcje New Rotation to przeciągnijmy/dodajmy nowy node o nazwie Make Rotator.
12. Podsumowując na ten moment:
    * [GET] Node GetActorRotation podaje nam aktualną wartość rotacji dla danego aktora
    * Przekazana wartość zostaje rozbita na osie X(Roll)Y(Pitch)Z(Yaw) czyli otrzymujem aktualną wartość dla danej osi.
    * [SET] Node SetActorRotation pozwoli nadawać nową wartość obrotu/rotacji dla danego aktora.
    * Dodatkowo używamy node'a Make Rotator pozwalającego nadawać dowolną wartość dla danej osi.
    * **Pytanie czym to wszystko się różni od tego co zrobiliśmy wcześniej?** Tym, że:
      + pomiędzy pobraniem aktualnej wartości [GET] a nadaniem nowej [SET] możemy dodać node'y odpowiedzialne dotychczas za myszkę.
      + Wcześniej przekazywaliśmy tylko pozycję myszki w osiach XY bezpośrednio do node'a który obracał aktorem i w tym miejscu następował błąd, którego próbujemy się pozbyć.
      + Teraz będziemy pobierać aktualną pozycje aktora w danej osi i będziemy ją DODAWAĆ do aktualnej pozycji myszki w danej osi. Otrzymany wynik przekażemy do node'a Make Rotator i zlikwidujemy błąd związany z niechcianym obrotem (offsetem) platformy w osi Z,
13. A więc node Get Mouse X przesuńmy obok Break Rotator i połączmy za pomocą node'a służącego do dodawania (pamiętając, że wartości, które dodajemy są ciągle typu float, czyli zmiennoprzecinkowego).
14. To samo zróbmy w przypadku Get Mouse Y (pamiętając o tym, że nadal powinniśmy zamieniać/odwracać aktualnie pozyskaną wartość pozycji myszy w osi Y).
15. Nie zapomnijmy o tym, aby funkcja SetActorRotation wykonywała się (exec/execution) co klatkę - w końcu chcemy płynnie obracać naszą platformą za pomocą myszki.
16. Przetestujmy naszą scenę. W osi Z nie występuje już błąd, platforma się nie przesuwa - ta kombinacja node'ów gwarantuje, że wartość osi Z nie zostaje naruszona. Przekazywane są tylko dokładne aktualne wartości myszy i aktora w osiach XY.
17. Zanim przejdziemy dalej popatrzmy na przejrzystość naszego kodu/grafu. Node'y nie powinny być połączone ze sobą zbyt długimi kreskami - lepiej czasami zduplikować dany node, aby ulepszyć zapis i przejrzystość całego schematu (czy np. u Ciebie połączenie od node'a Floor do SetActorRotation ma 10 kilometrów? Jeśli tak to skróć je!).
18. Oprócz przejrzystości schematu i przesuwania się obiektów w osi Z to występuje też w naszym projekcie inny błąd związany z tym, że właściwie możemy w nieograniczony sposób obracać naszą platformą w osiach X i Y.
19. Powinniśmy zablokować obrót i ustawić odpowiedni przedział wartości/stopni w osiach XY np. przedział od -40 do 40 powinien być wystarczający.
20. Blokowanie/ustawianie/ograniczanie/wyznaczanie przedziałów jest zawsze w programowaniu związane z słowem Clamp i biblioteką Math.
21. Pamiętajmy również nadal, że obrót/rotacja, koordynaty myszki są typu float.
22. Zadanie:
    * Dodać odpowiednie node'y, które ograniczą ruchy platformy/podłogi w podanym przez nasz przedziale (min. -40 i max 40 stopni).
    * Wykonać zrzut ekranu, na którym widać schemat dotyczący ulepszonego sterowania myszką wraz z oknem głównym, w którym widać nazwę projektu znajdującą się w prawym górnym rogu okna (pamiętaj, aby w nazwie projektu było imię i nazwisko).
    * Zrzut ekranu prześlij na [adamzdzcichy@gmail.com](mailto:adamzdzcichy@gmail.com) (tytuł wiadomości: Zadanie 05 Blueprinty - Imię Nazwisko + grupa, rocznik) lub wrzuć na swojego githuba, gdzie powinien znajdować się cały projekt UE4 (pamiętaj o stworzeniu repozytorium na githubie dla całego projektu)