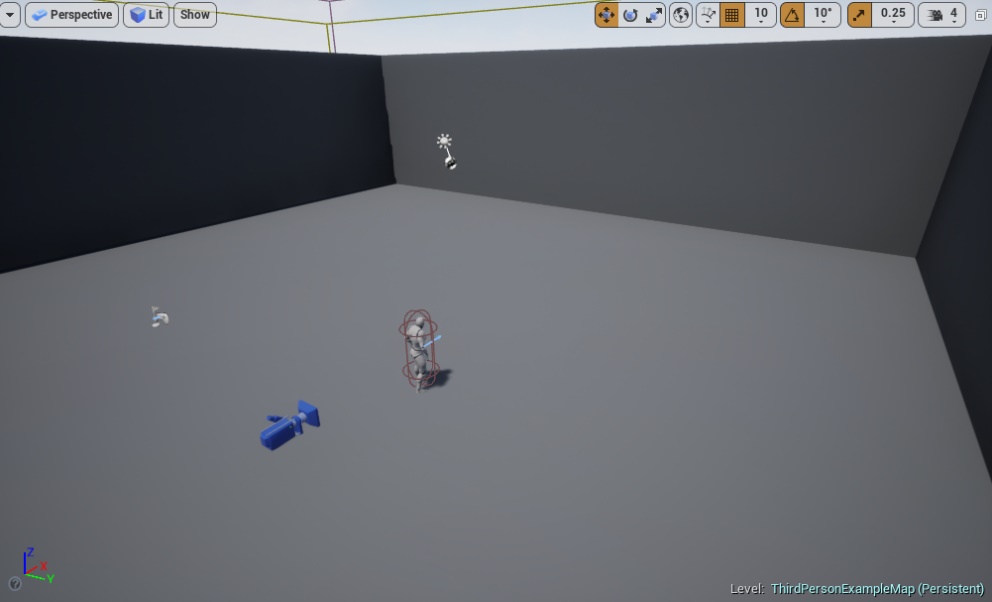
# Lecture 10: Blueprints in Action 1

# Exercise 2

In this exercise, you will create a Blueprint that moves toward a random point. When the Blueprint reaches its destination, a new random destination is set. The movement will be calculated step-by-step to show the different uses of vectors.

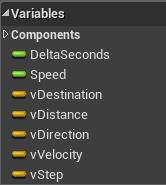
## Directions

1. Create a new project using the **Third Person** template. Remove the Static Mesh and Text Render Actors from the middle of the Level (see Figure 1).



*Figure 1: Clearing the Level*

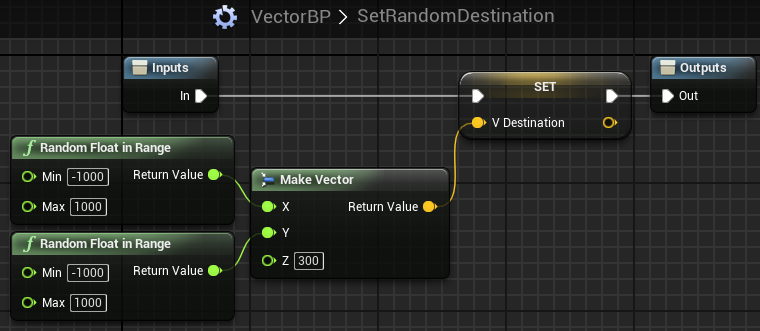
1. In the **Content Browser**, click the green **Add New** button and select “**Blueprint Class**” to create a new Blueprint class. In the **Pick Parent Class** window, choose “**Actor**”. Rename the Blueprint “**VectorBP**”.
2. Double-click **VectorBP** to open the **Blueprint Editor**.
3. In the **Components** panel, click the **Add Component** button and choose the **Static Mesh** component. In the **Details** panel, assign a Static Mesh asset to the componentto represent the Blueprint in the Level.
4. In the **My Blueprint** panel, create the variables shown in Figure 2.



*Figure 2: VectorBP Variables*

* **DeltaSeconds (float)**: Stores the value passed by the **Tick** event, representing the time that has passed since the last tick.
* **Speed (float)**: Stores the speed value of the Blueprint. Set the default value to 100 (cm/s).
* **vDestination (vector)**: Vector that stores the position where the Blueprint needs to move.
* **vDistance (vector)**: Vector indicating the distance from the current position of the Blueprint to its destination.
* **vDirection (vector)**: Unit vector that indicates the direction that the Blueprint should follow to reach its destination.
* **vVelocity (vector)**: Vector representing the speed and direction of the Blueprint in cm/s.
* **vStep (vector)**: Vector with the final result of the calculation, indicating the movement that the Blueprint should make at the current tick.

1. Create a macro named “**SetRandomDestination**” to set a random destination. The nodes of the macro are shown in Figure 3. The value of the **Z** axis is fixed at “**300**”, and the values of the **X** and **Y** axes can range from “**–1000**” to “**1000**”. The vector created is stored in the **vDestination** vector.



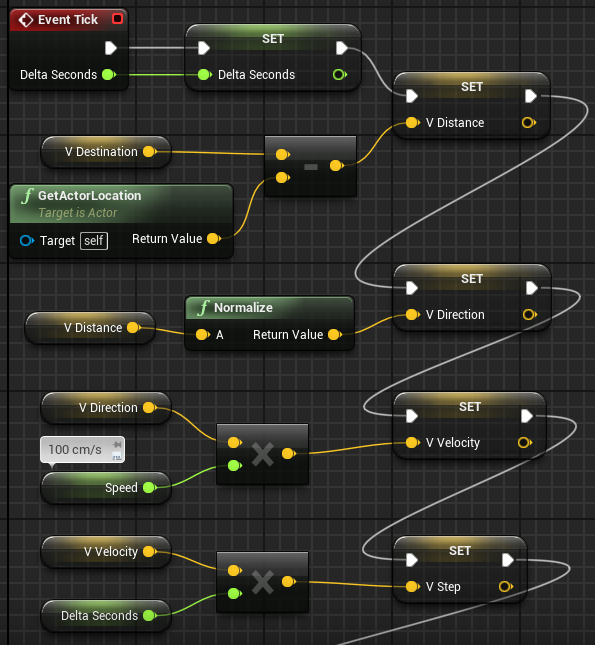
*Figure 3: SetRandomDestination Macro*

1. The **SetRandomDestination** macro must be called from the **BeginPlay** event to set the starting destination (see Figure 4).



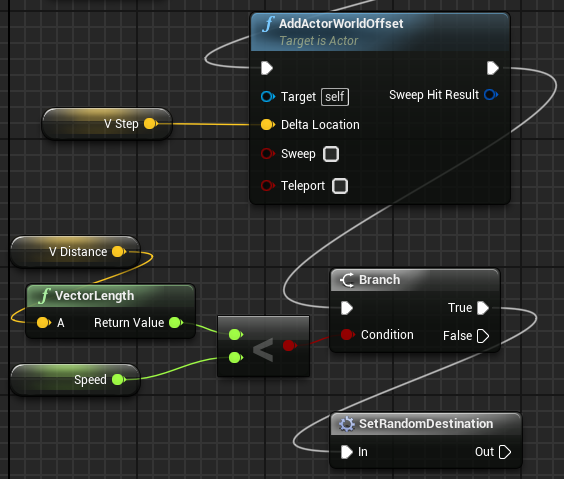
*Figure 4: Begin Play Event*

1. Figure 5 shows the first part of the **Tick** event, where the **vDestination**, **vDistance**, **vDirection**, and **vVelocity** vectors are calculated until the value of the **vStep** vector is found. All of this calculation could be unified in a single mathematical expression, but each part of the calculation has been separated out to facilitate understanding.



*Figure 5: Vector Operations*

1. Figure 6 shows the second part of the **Tick** event, which moves the Blueprint using the **AddActorWorldOffset** function and tests if the Blueprint has arrived at its destination. This test is done by doing a comparison to see if the current distance is smaller than the value of the **Speed** variable. If “**true**”, a new destination is set.



*Figure 6: Vector Operations*

1. Add an instance of the **VectorBP** Blueprint to the Level and then play the Level.

## Outcome

After pressing **Play**, you will see an instance of the **VectorBP** Blueprint moving toward a random point. After reaching the point, it will move to another random point.