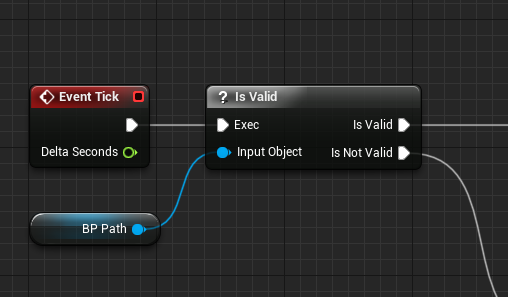
Описание работы кода

Тестовый проект состоит из нескольких взаимосвязанных Blueprints классов.

* **BP\_Car** - Сам тестовый автомобиль.
* **BP\_Path** - сплайн. Траектории которой автомобиль должен придерживаться
* **Static mesh** - преграды, столкновения которых автомобиль должен избегать
* **BP\_LiveCube** - движущаяся преграда.

В нужный Blueprints (BP\_Path) по которому автомобиль и поедет добавлен компонент *Spline*. Вытащив его в мир можно настроить его длину и траекторию.  
Зайдя в класс с автомобилем в окне Vievport располагаются 7 “**ArrowComponts**”. От них макросы выполняют проверку столкновения.

В функции (**PathAi)**, Автомобиль, находит путь до ближайшей точки Сплайна. В ней же установлена переменная “*Closensess*” которая отвечает за максимальную возможную отдалённость автомобиля от траектории.

В окне *EventGraph* на Event Tick стоит проверка на взаимодействие данного класса и класса со сплайном (*BP\_Path*).  
  
Дальше для большего удобства подключен *Sequence*. Он в порядке их обозначения вызовет сигналы на выходах данной функции.

На первом сигнале располагаются макросы “**TraceMacro**”, функцией которых заключается выполнять трассировку столкновения вдоль линии. Имеются переменные типа Float, задающие длину трассировки и компоненты **“ArrowComponts”** от которых происходит выстрел трейса.

На втором сигнале Sequence располагается код отвечающий за поиск пути для автомобиля и столкновение трейсов с преградами.

При столкновении трейсов от “**TraceLeft-Forward**" и “**TraceRight-Forward**" автомобиль заранее будет поворачивать колёса в противоположную сторону.

Трейсы “**TraceLeft**” ,” **TraceRight**” и “**TraceLeft-side**”,” **TraceRight-side**” определяют, сможет ли автомобиль проехать между преградами не задев себя.

Трейс “**TraceForwardLine**” определяет расстояние между собой и преградой. В случае если несколько трейсов и “**TraceForwardLine**” встречают препятствие, то автомобиль сдаёт назад.