

AI автомобиля в изменчивом мире на примере Ex Machina

Докладчик

Антон Савин, ведущий программист Targem Studio, anton.savin@nival.com
<http://www.targem.ru>



- Pathfinding
- Pathtracking
- Бой

- **Двумерная карта проходимости**
Практически везде, где есть ландшафт
- **Waypoints**
Геометрически сложные пространства
- **Ориентированная карта проходимости**
Казалось, может нам помочь
- **Учитывать скорость при поиске пути**
Слишком сложно по времени

- **Взаимодействие с физическим движком**
Положением и скоростью машины управляет физический движок. Управлять машиной можно лишь нажатием на газ, тормоз и крутя руль.
- **Объезд препятствий**
Не все препятствия можно объехать, просто найдя подходящий путь.

Общий алгоритм:

```
while( не доехали )
{
    ехать к текущей точке();          //(1)
    if( близко к текущей точке )
        притормозить();              //(2)

    if( доехали до текущей точки )    //(3)
    {
        if( текущая точка - последняя )
            break;
        else
            ++текущая точка;
    }
}
```

Замечание:

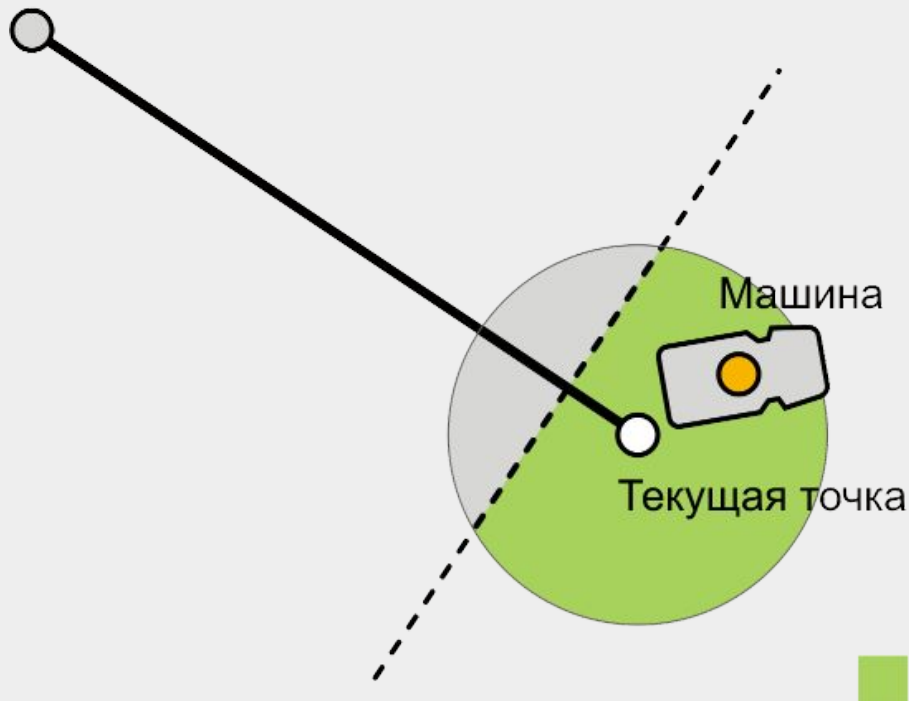
Точки пути должны быть как можно дальше друг от друга, т.е путь надо спрямлять.



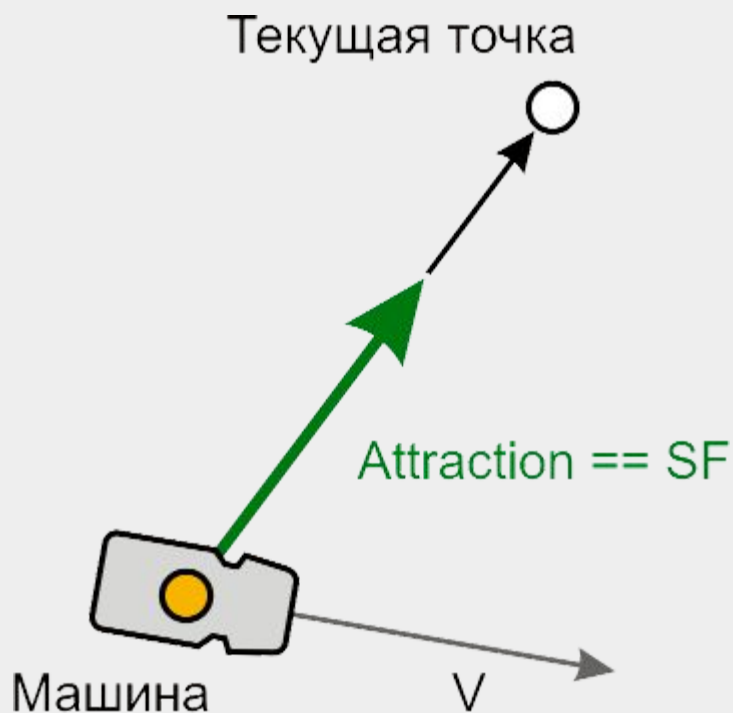
- Повернуть руль в сторону точки, удерживать машину на заданном курсе
- **Ex Machina:** простой подход – желаемый угол поворота колес пропорционален углу между направлением «вперед» для машины и направлением до точки пути. Может начаться «вихляние».
- Если машина развернута задом к точке и скорость мала, то можно развернуться «по трем точкам» со сдачей назад.
- **Пути для улучшения:** PID-регуляторы (учитывается не только само отклонение, но также интеграл отклонения за некоторый период времени и производная отклонения)

- Чем больше угол поворота к следующей точке и скорость машины, тем сильнее надо притормаживать и тем раньше надо начать торможение. Если угол поворота невелик, можно совсем не тормозить.
- Точное значение силы трения на всем пути неизвестно, так как оно зависит от типа ландшафта, от угла наклона и т.п.
- Считаем, что коэффициент трения не меньше некоторого минимума.

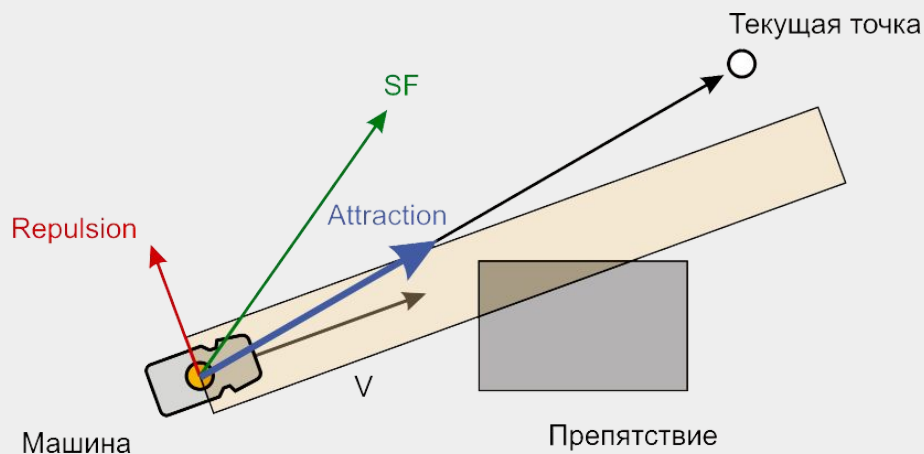
Предыдущая точка



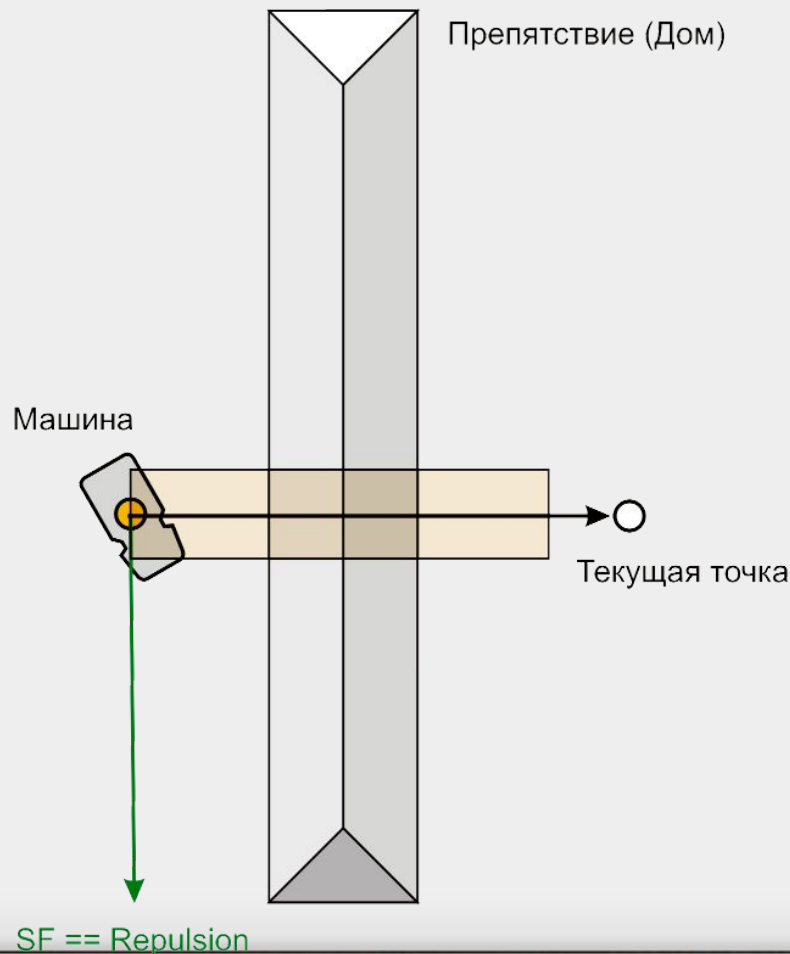
- Препятствия: статические, динамические
Статические: деревья, строения
Динамические: другие машины, обломки
- Нужен единый механизм объезда препятствий
- Почему бы не искать пути так, чтобы препятствия не надо было объезжать?
Можно ли препятствия врисовать в карту проходимости?
- Модификация алгоритма Flocking: считаем для машины вектор steering force (SF) и дальнейшее движение осуществляем, исходя из него.



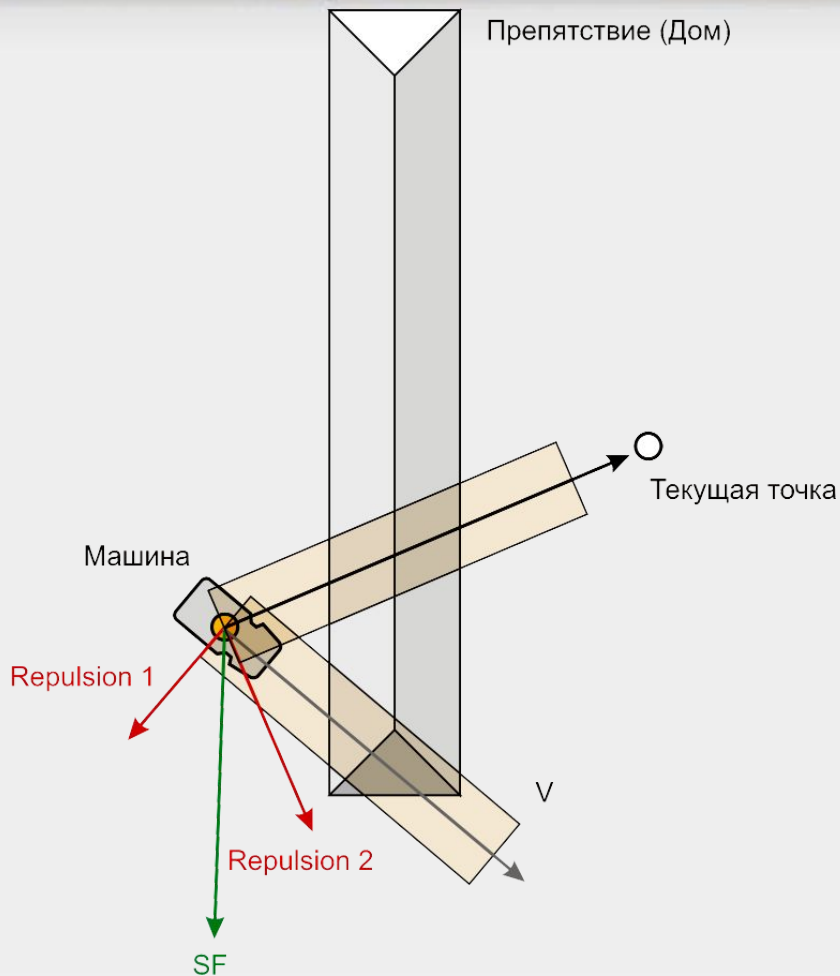
В простейшем случае (когда нет препятствий) SF равен силе притяжения к очередной точке пути.



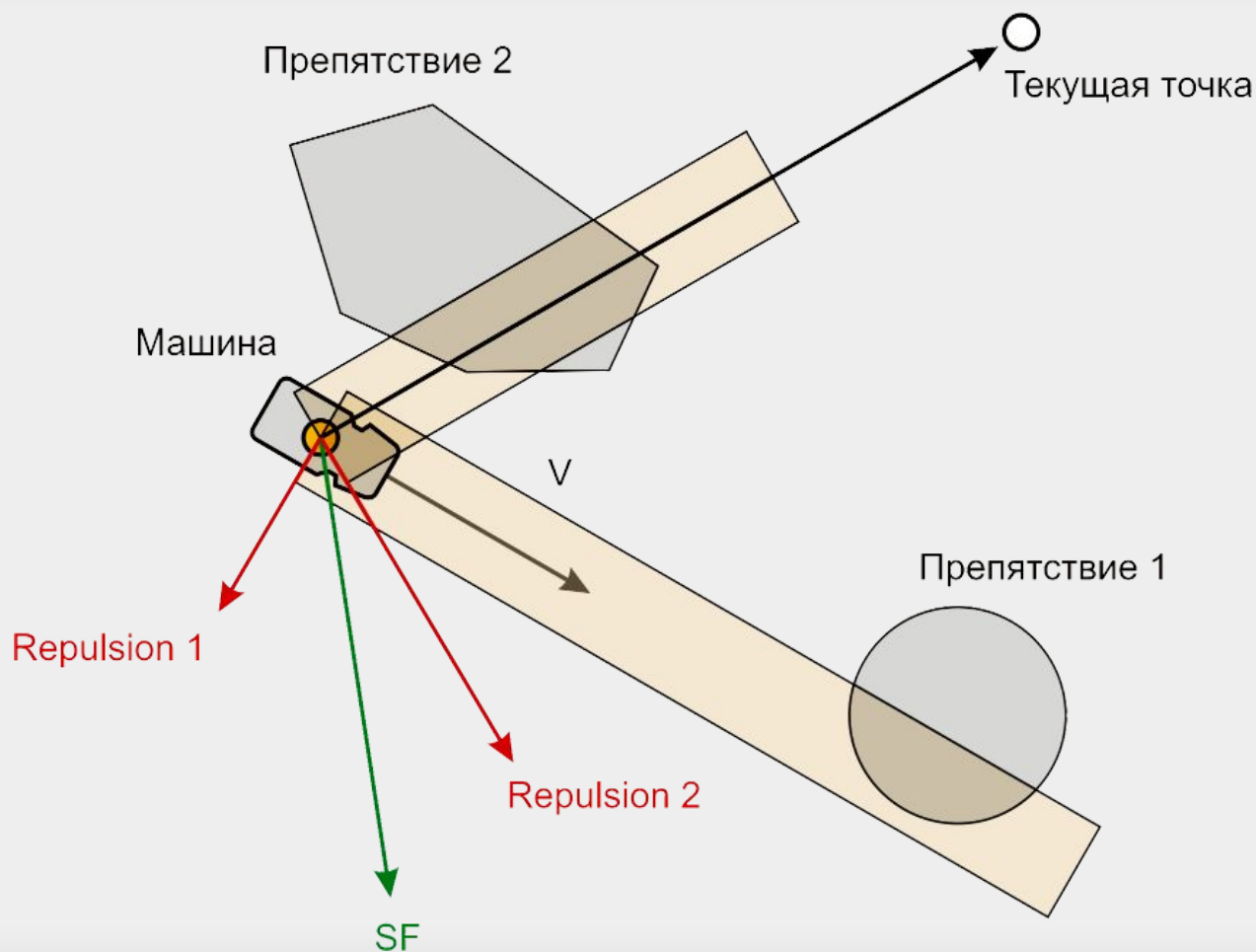
- Препятствия впереди по ходу движения ищем с помощью прямоугольника шириной чуть больше машины и длиной, пропорциональной скорости машины.
- Сила отталкивания от препятствия направлена под 90° к скорости машины, в сторону от центра препятствия.



- Существует еще один тип препятствий: стоящие между машиной и точкой пути. Ищем их с помощью соответствующего прямоугольника.
- Сила отталкивания от таких препятствий перпендикулярна к направлению до точки пути, в сторону от центра препятствия
- Если любое препятствие близко, то аннулируем силу притяжения к точке пути.
- Новый тип препятствий позволяет объезжать выпуклые препятствия любого размера.



- Одно и то же препятствие может выступать в обоих ролях одновременно.



- На самом деле все вышеперечисленные алгоритмы в Ex Machina работают в 3D. То есть вместо прямоугольников – боксы, препятствия – выпуклые многоугольники или сферы.
- Алгоритм работает для любых выпуклых объектов, однако на практике применяются только боксы и сферы, т.к. вычислительно сложно определять пересечения.
- Алгоритм работает не всегда. Например, для его работы требуется, чтобы расстояния между препятствиями были больше ширины машины, иначе машина может никогда их не объехать

Вопросы?

Антон Савин, anton.savin@nival.com