

- 1、在菜单左边加一个智能体的选项
- 2、建一个智能体蓝图：
- 3、CitizenCharacter.h控制人的行为，添加时间轴控制行人移动：
- 4、怎么一开始就调用行人里面的时间轴等其他函数：
- 5、添加控制智能体的UI界面：
- 6、怎么增加编辑组件的光标：
- 7、怎么保存数据，行人路径编辑过的节点：
- 8、实时编译快捷键：ctrl+alt+f11，全屏/退出：f11
- 9、初始化ue的蓝图，用编辑对象来初始化界面（挂载界面？）：
- 10、怎么保存智能体的配置信息：
- 11、模型和c++关联：
- 12、怎么在C++调用蓝图的函数：
- 13、初始化Character的Controller：
- 14、更新Actor的坐标轴：
- 15、UE引擎中细节面板(DetailsPanel)内容消失问题
- 16、角色不能移动
- 17、C++编译器把变量声明识别成函数声明

by a_hoi

1、在菜单左边加一个智能体的选项

在UE table加一行，要改对应智能体的类，等下建好再改也行，AgentDef.h文件EAgentType枚举要加一项

2、建一个智能体蓝图：

先在rider建好一个类，然后在ue新建蓝图，选择继承刚刚新建的类，蓝图里面的组件的名字要和类成员的名字一样

3、CitizenCharacter.h控制人的行为，添加时间轴控制行人移动：

```
void ACitizenCharacter::OnTimelineProcess(float v)
{
    //UE_LOG(LogTemp,Warning,TEXT("in NavigateToSpline 啦啦啦啦啦啦啦 %f"),DestLane->getUSplineComponent()->GetSplineLength());

    const float length=DestLane->getUSplineComponent()->GetSplineLength();

    const FVector currentLocation = DestLane->getUSplineComponent()-
```

```
>GetLocationAtDistanceAlongSpline(v*length, ESplineCoordinateSpace::World);

    FRotator currentFRotator=DestLane->getUSplineComponent()-
>GetRotationAtDistanceAlongSpline(v*length, ESplineCoordinateSpace::World);

    //SetActorLocationAndRotation(currentLocation, currentFRotator);
    //SetActorRotation(currentFRotator);

    //UE_LOG(LogTemp, Warning, TEXT("inooooooooooooooooooooooooooooooooooooooooooooo
OnTimelineProcess %f"), currentLocation.X);
    //NavigateTo(currentLocation, false);

    FVector loc = DestLane->getUSplineComponent()-
>GetLocationAtDistanceAlongSpline(v*length, ESplineCoordinateSpace::World);
    FVector tan = DestLane->getUSplineComponent()-
>GetTangentAtDistanceAlongSpline(v*length, ESplineCoordinateSpace::World);
    FRotator rot = tan.Rotation();
    SetActorLocationAndRotation(loc, rot, false, nullptr,
ETeleportType::ResetPhysics);

}
```

4、怎么一开始就调用行人里面的时间轴等其他函数：

在ProjectManagerComponent.h文件中的void StartSimulateCurrentProject()函数调用

5、添加控制智能体的UI界面:

根据原有的UI文件，例如carui.h，.cpp新建一个UI类，在ue新建蓝图，选择继承刚刚新建的UI类，可以在从其他蓝图UI复制下拉框等组件过来，组件的名字要和类成员的名字一样，在函数NativeConstruct初始化UI

6、怎么增加编辑组件的光标:

EditMeshComponent.cpp的很多个函数负责，在if else语句里，具体查找一下

AxisActor.cpp文件也有，在if else语句里，具体查找然后补充一下

7、怎么保存数据，行人路径编辑过的节点:

EditMeshComponent.cpp文件里面，保存：函数SaveActor:

```
//这里是保存行人路径的节点数据
        if(ACitizenLane* clane = Cast<ACitizenLane>(agent))
        {
```

```
data.CitizenLaneSectionPosArr=clane->GetSectionData();  
}
```

要先在SimProject.h文件的结构体FAgentData定义数据结构，就像：

```
/* -----行人路径属性----- */  
UPROPERTY(BlueprintReadWrite, Category=Lane)  
TArray<FLaneSectionData> CitizenLaneSectionPosArr;
```

加载：函数LoadProjectAgents：

```
//这里是根据保存的数据重新生成行人路径的节点  
else if(d.AgentType == EAgentType::CitizenLane)  
{  
    ACitizenLane* clane=Cast<ACitizenLane>(actor);  
    clane->CreateSection(d.CitizenLaneSectionPosArr);  
}
```

8、实时编译快捷键：ctrl+alt+f11，全屏/退出：f11

9、初始化ue的蓝图，用编辑对象来初始化界面（挂载界面？）：

在EditPropertyUI.h新建函数，

```
void UEditPropertyUI::ShowPersonLane(bool show, AAgentActor* actor)  
{  
    if(show)  
    {  
        if(!BP_EditCitizenLaneUI)  
        {  
            UClass* EditCitizenClass =  
StaticLoadClass(UEditCitizenLaneUI::StaticClass(),this,EDIT_CITIZEN_LANE_CLASS);  
            BP_EditCitizenLaneUI = CreateWidget<UEditCitizenLaneUI>  
(GetWorld(),EditCitizenClass);  
            SB_Property->AddChild(BP_EditCitizenLaneUI);  
        }  
        BP_EditCitizenLaneUI->Init(actor);  
        BP_EditCitizenLaneUI->  
>SetVisibility(ESlateVisibility::SelfHitTestInvisible);  
        SetItemPadding(BP_EditCitizenLaneUI);  
    }  
    else  
    {  
        if(BP_EditCitizenLaneUI)  
        {  
            BP_EditCitizenLaneUI->RemoveFromParent();  
        }  
    }  
}
```

```

    }
    BP_EditCitizenLaneUI = nullptr;
}
}

```

调用：

```

void UEditPropertyUI::Init(AInteractableActor* focusActor)
{
    BP_EditBasicUI->Init(focusActor);
    if(AAgentActor* agent = Cast<AAgentActor>(focusActor))
    {
        EAgentType type = agent->GetAgentType();
        ShowLane(type == EAgentType::Lane, agent);
        ShowCar(type == EAgentType::Car, agent);
        ShowPersonLane(type == EAgentType::Citizen, agent);
        // switch()
        // {
        // case EAgentType::Lane:
        //     ShowLane(true, agent);
        //     break;
        // }
    }
}

```

10、怎么保存智能体的配置信息：

比如说我给智能人**A**设置了路径1，怎么保存**A**的路径1，下次加载项目**A**默认就是路径1，不用重新点下拉框再选一遍

首先在保存数据结构的文件**SimProject.h**里面的结构体**FAgentData**定义一项数据，例如目的路径，好像用的是tag标记之类的：

```

// 行人目的路径
UPROPERTY(BlueprintReadWrite, Category=CitizenLane)
FName CitizenDestinationLaneTag;

```

然后在文件**EditMeshComponent.cpp**的函数**SaveActor**保存：

```

// 行人数据
else if(ACitizen* citizen = Cast<ACitizen>(agent))
{
    data.Location = citizen->GetCharacter()->GetActorLocation();
    data.Rotation = citizen->GetCharacter()->GetActorRotation();
    //保存行人的目的路径tag信息，以能够重新加载
}

```

```

        if(citizen->GetCharacter()->GetDestCitizenLane())
        {
            data.CitizenDestinationLaneTag=citizen->GetCharacter()-
>GetDestCitizenLane()->GetTag();
        }
        // data.Location = citizen->GetActorLocation();
        // data.Rotation = citizen->GetActorRotation();
    }

```

然后在文件ProjectManagerComponent.cpp的函数ResetAgents()重新加载数据，就像：

```

//从已保存的信息中来初始化每个智能体人的目的地路径
for(auto* it : agentArr)
{
    AAgentActor* actor = Cast<AAgentActor>(it);
    FName tag = actor->GetTag();
    if(tag == "") continue;
    const FAgentData& data = m_CurEditProject.GetAgentData()[tag];
    //开始给每个智能人赋值啦啦啦
    if(data.AgentType == EAgentType::Citizen)
    {
        ACitizen* citizen = Cast<ACitizen>(it);
        //citizen->ResetStatus();
        for(auto * it2 : agentArr)
        {
            if(ACitizenLane* clane = Cast<ACitizenLane>(it2))
            {
                if(clane->GetTag() == data.CitizenDestinationLaneTag)
                {
                    citizen->GetCharacter()->SetDestLane(clane);
                }
            }
        }
    }
}
}

```

11、模型和c++关联：

新建c++类，编译，在ue右键新建蓝图类，选择继承c++类，然后在蓝图类选择模型、材质

12、怎么在C++调用蓝图的函数：

BlueprintImplementableEvent

<https://www.bilibili.com/read/cv28532491/>

13、初始化Character的Controller:

在调用导航前调用函数来初始化Controller

```
void ABoatCharacter::NavigateTo(FVector location)
{
    //初始化Character的Controller
    SpawnDefaultController();

    if(GetController()==nullptr)
    {

UE_LOG(LogTemp,Warning,TEXT("inoooooooooooooooooooooooooooooooooooooooooooooooooooo
AIController null"));
    }
    AAIController* MyAIController = Cast<AAIController>(GetController());
    UAIBlueprintHelperLibrary::SimpleMoveToLocation( MyAIController , location);

}
```

14、更新Actor的坐标轴:

在文件AAxisActor.cpp的函数InitTransform中:

```
else if(ABoat* boat = Cast<ABoat>(m_operateActor))
{
    if(bUpdateActor) boat->GetCharacter()->SetActorLocation(location, false,
nullptr, ETeleportType::TeleportPhysics);
    location = boat->GetCharacter()->GetActorLocation();
    // if(bUpdateActor) citizen->UpdateLocation(location, false);
    if (axisType == EAxisType::RELATIVE || handleType == EHandleType::SCALE)
    {
        rotation = boat->GetCharacter()->GetActorRotation();
    }
}
```

15、UE引擎中细节面板(DetailsPanel)内容消失问题

打个比方 有一个Blueprint, 其中有一个Component叫做XXXComp, 点击XXXComp, 细节面板中没有内容

解决方法一 将XXXComp命名成XXXComp2(可以用, 但是我不喜欢, 一个合理的名字太难命名了)

解决方法二 将XXXComp冲Actor中去除. 就是将代码中, 所有有关的内容都注释掉, 并且编译通过

```
// UPROPERTY(VisibleAnywhere)
// class UXXXComponent* XXXComp;
```

然后再Blueprint中随便改个变量, 点击保存.(这个时候可以通过文本文件打开蓝图对应的.uasset文件, 搜索XXXComp和XXXComponent, 保证搜索不到XXXComphe和XXXComponent)

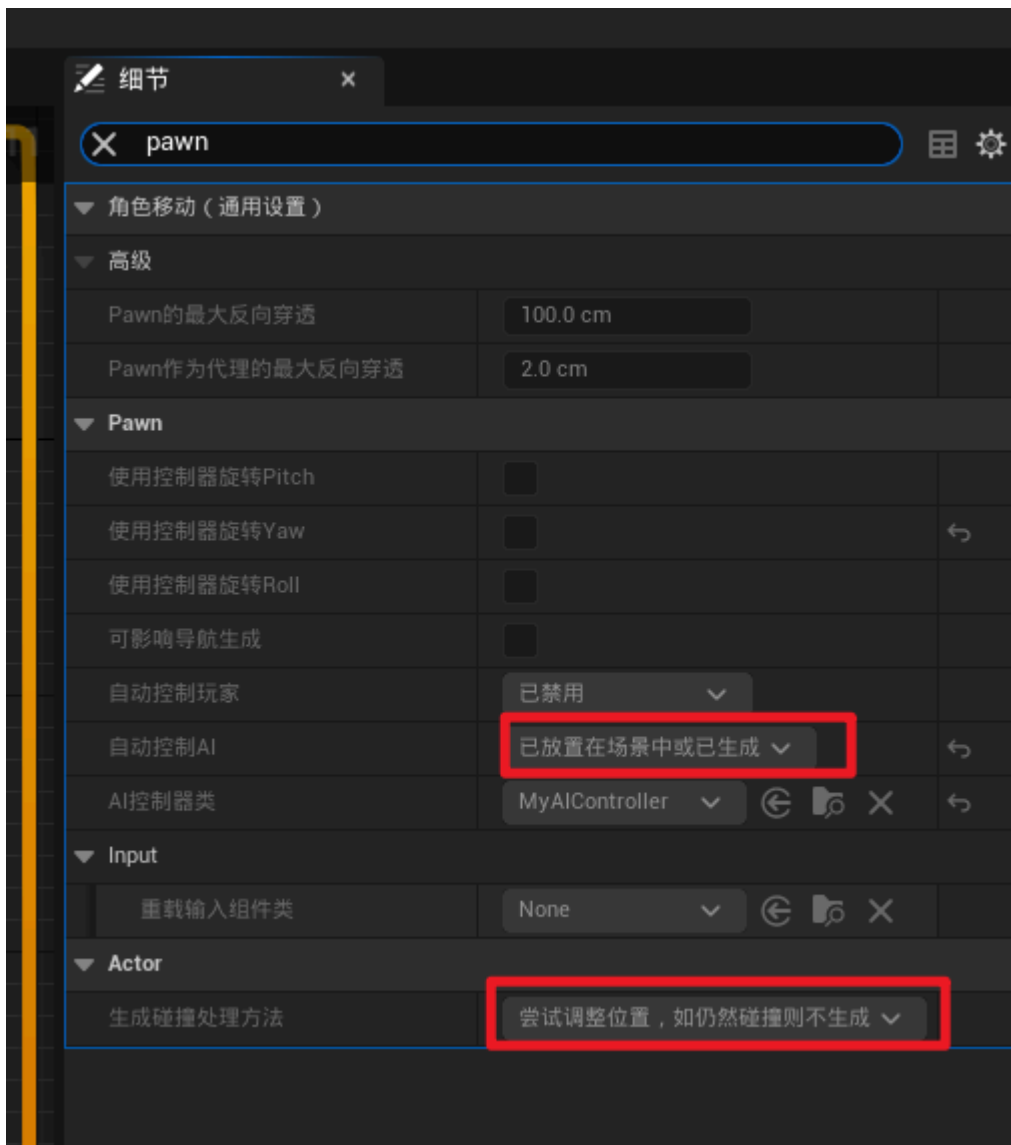
然后放开注释, 重新编译

```
UPROPERTY(VisibleAnywhere)
class UComponent* XXXComp;
```

然后再蓝图中随便改个变量, 点击保存, 就可以了. (可以通过文本文件打开蓝图对应的.uasset文件, 搜索XXXComp和XXXComponent, 能搜索到XXXComp和XXXComponent就可以了)

16、角色不能移动

(1) 调一下角色的pawn:



(2) 等导航网格全部生成完毕

17、C++编译器把变量声明识别成函数声明

```
template <typename T>
class FixedSizeQueue {
public:
    explicit FixedSizeQueue(std::size_t max_size) : max_size_(max_size) {}

    void push(const T& value) {
        if (queue_.size() == max_size_) {
            queue_.pop_front();
        }
        queue_.push_back(value);
    }

    // 其他需要的函数，例如front(), back(), size()等

private:
    std::size_t max_size_;
    std::deque<T> queue_;
};
```


//识别成函数声明，而不是代码声明

```
FixedSizeQueue<FString> DetailInfoList(10);
```

//解决办法：把括号改成花括号

```
FixedSizeQueue<FString> DetailInfoList{10};
```

//其他解决办法：

<https://blog.csdn.net/qfturayls/article/details/108556707>