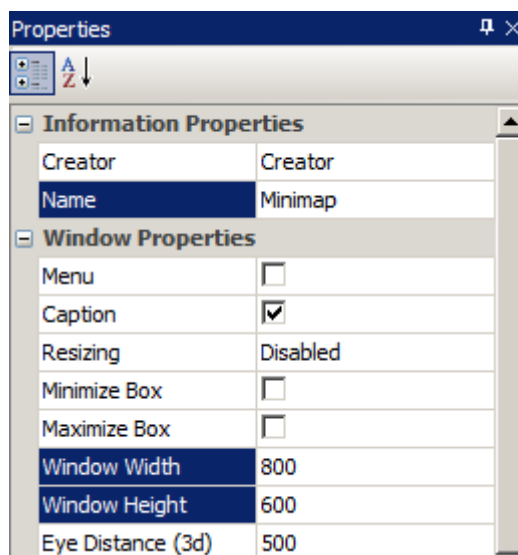
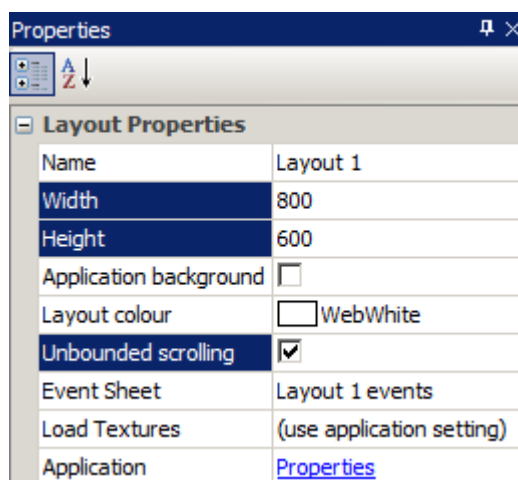
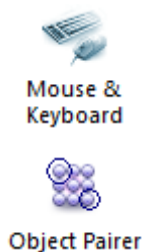


Всем приветы! Сегодня я расскажу вам как сделать качественную миникарту на конструкторе игр и программ Scirra Construct Classic. Миникарта – это такой объект, который показывает важные окружающие объекты на карте. В программе уже есть стандартный объект Minimap, но мы сделаем такую миникарту, иконки которых можно легко менять. Так же миникарта будет отображать противников и угол их направления не относительно уровня, а относительно главного персонажа, который будет всегда стоять в центре миникарты. Так же уровень будет бесконечный, а жанр игры Top Down Shooter.

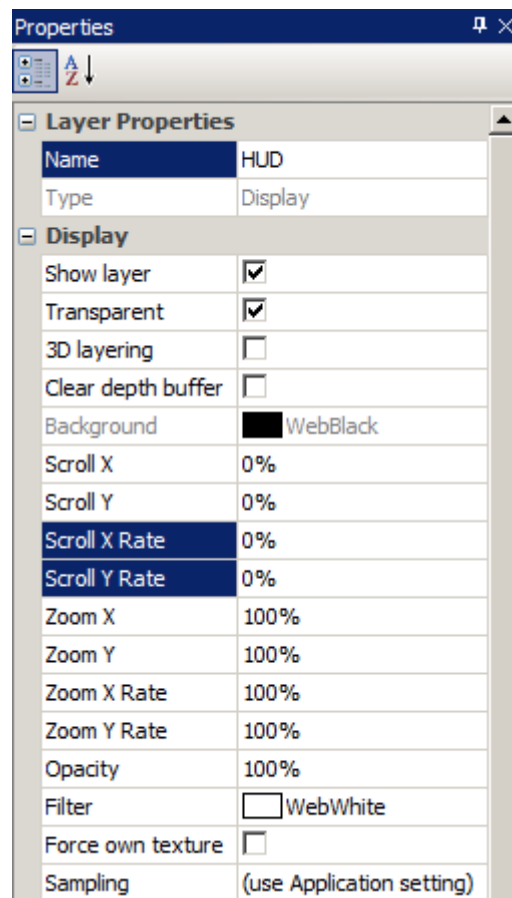
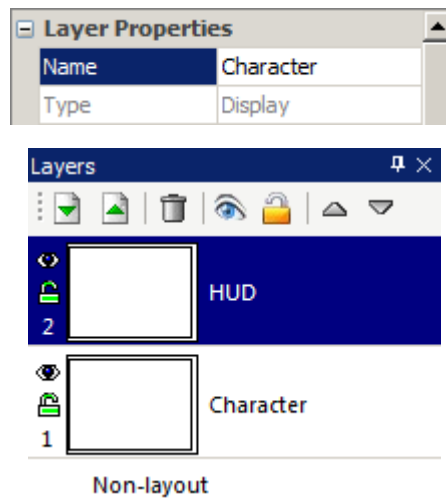
Создайте New Direct-X game и настройте проект. Размер уровня и окна можете поставить любыми. Приложение называйте тоже как хотите. Так же поставьте галочку на Unbounded scrolling, чтобы камера не прерывалась на границах уровня.



Добавьте 2 объекта: Mouse & Keyboard и Object Pairer.



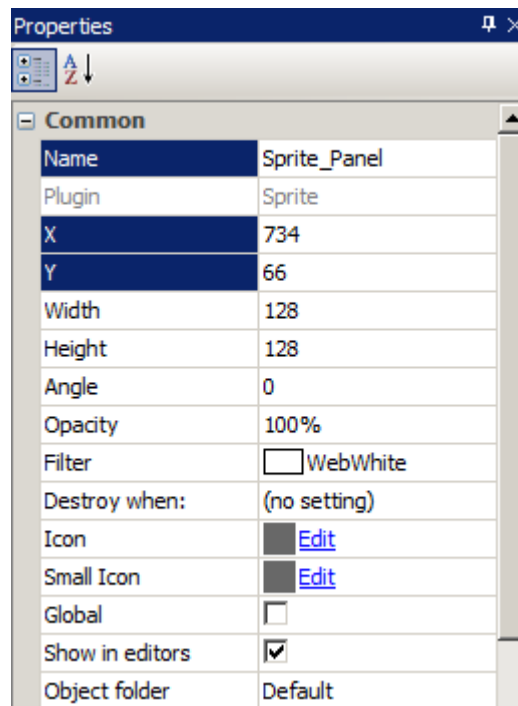
Для начала поработаем со слоями. Назовите первый слой Character и создайте второй слой, который назовите HUD. Установите параметр прокрутки второго слоя на 0. На нём и будет находиться наша миникарта.



Для начала давайте добавим панель миникарты. Она будет играть роль объекта, на котором вся миникарта и будет базироваться. Создайте объект Sprite на слое HUD и назовите его Sprite_Panel. Установите его там, где хотите видеть вашу миникарту.



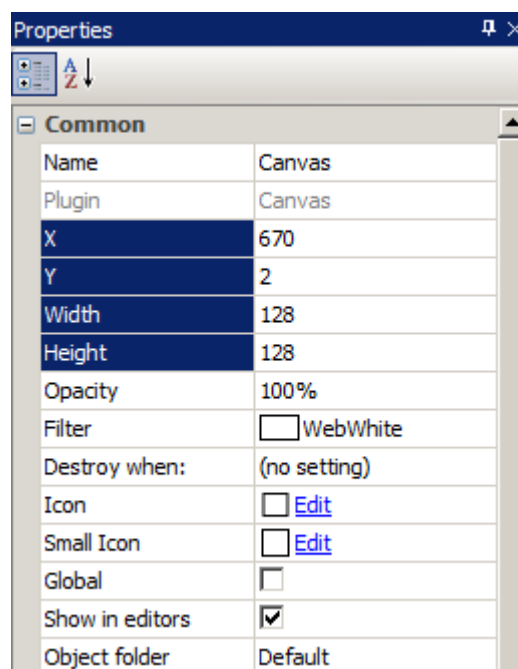
Sprite



Добавьте объект Canvas на слое HUD, который будет отображать иконки объектов на уровне. Настройте его размер такой же, какой стоит на панели миникарты. Можете поставить его за уровень или наложить на панель – это не имеет разницы, т. к. он всё равно будет стоять на панели.



Canvas



Теперь можем добавить игровые объекты. Ими будут Главный Герой и второстепенные персонажи.



Создайте объект Sprite на слое Character и назовите его Sprite_Character. Переместите его в центр уровня. Т. к. в нашем случае вид сверху, добавьте ему поведение 8 Direction и отключите вращение, т. к. персонаж будет всегда смотреть на курсор. Так же поставьте ему атрибут Center view on me.




Sprite

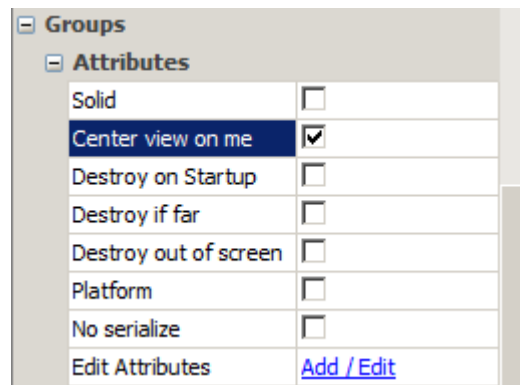
Properties

Common

Name	Sprite_Character
Plugin	Sprite
X	400
Y	300
Width	32
Height	32
Angle	0
Opacity	100%
Filter	<input type="checkbox"/> WebWhite
Destroy when:	(no setting)
Icon	 Edit
Small Icon	 Edit
Global	<input type="checkbox"/>
Show in editors	<input checked="" type="checkbox"/>
Object folder	Default

8Direction Behavior

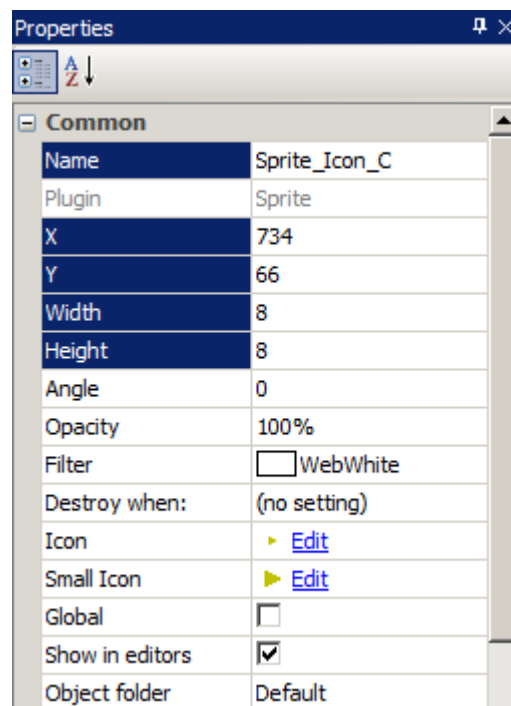
This behavior	 Remove
Name	8Direction
Control	Player 1
Max Speed	200
Acceleration	600
Deceleration	500
Rotation	None
Directions	8 directions
New Behavior	Add



Теперь нужно создать иконку персонажа, которую будет отображать миникарта. Создайте объект Sprite на слое HUD, переименуйте его в Sprite_Icon_C и оттяните за уровень или поставьте на панель.



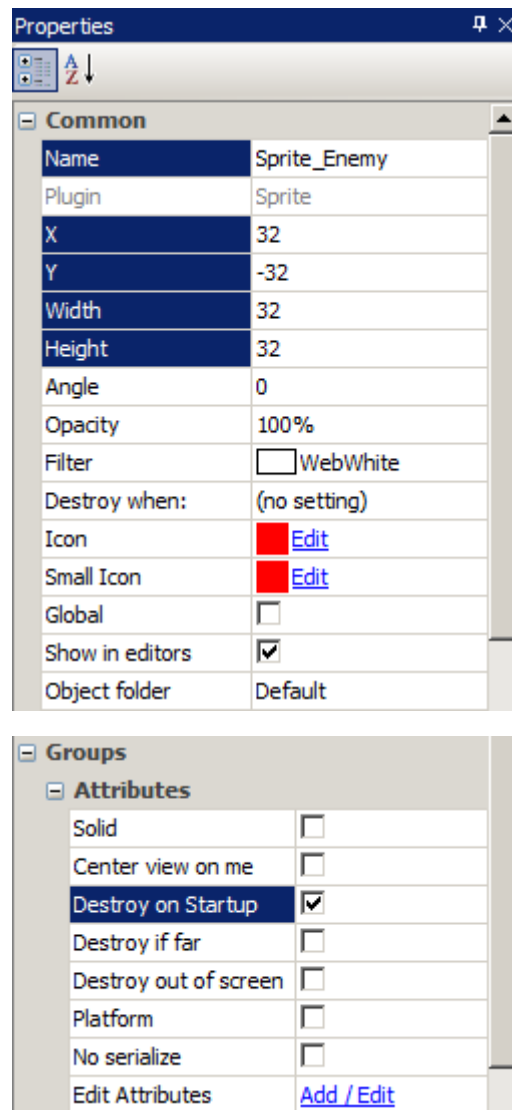
Sprite



Теперь нужно создать второстепенного персонажа. Создайте объект Sprite на слое Character. Назовите его Sprite_Enemy и вынесите его за пределы уровня. Так же поставьте ему атрибут Уничтожение при запуске.

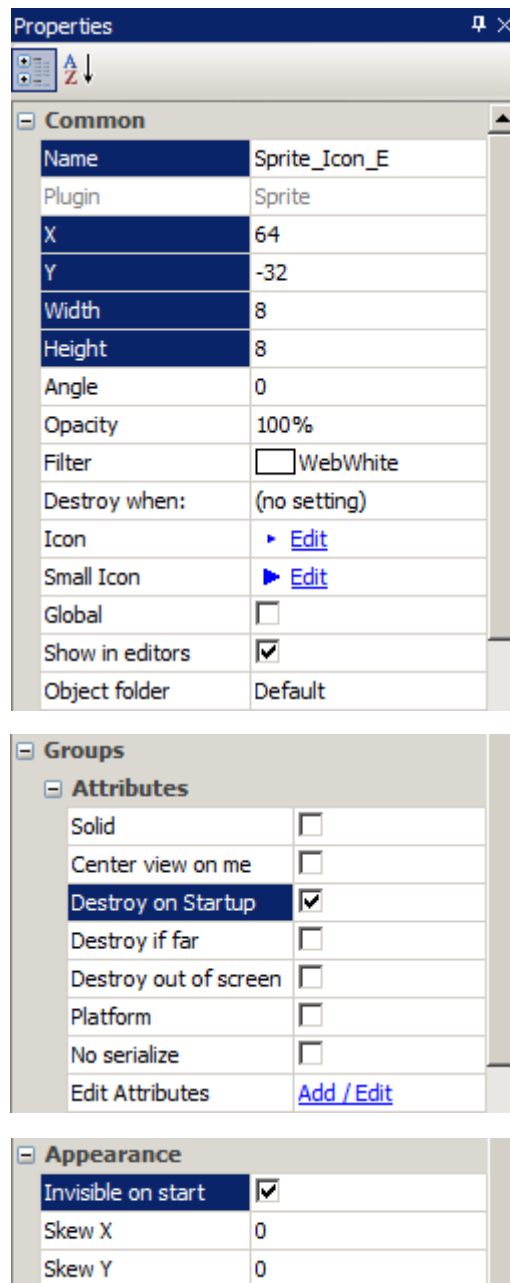


Sprite



Осталось добавить иконку второстепенного персонажа для миникарты. Создайте объект Sprite на слое HUD и назовите его Sprite_Icon_E. Вынесите его за пределы уровня и поставьте атрибут Уничтожение при запуске. Так же не забудьте поставить видимость Спрайта – Невидимый при старте.



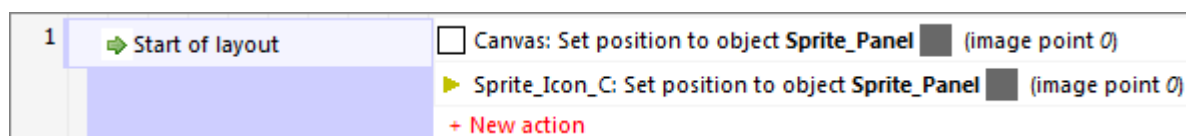


На этом создавать объекты мы закончили. Перейдём к событиям. Сначала поставьте Canvas и иконку Главного Героя на панель.

+ System: Start of layout

-> Canvas: Set position to object Sprite_Panel (image point 0)

-> Sprite_Icon_C: Set position to object Sprite_Panel (image point 0)

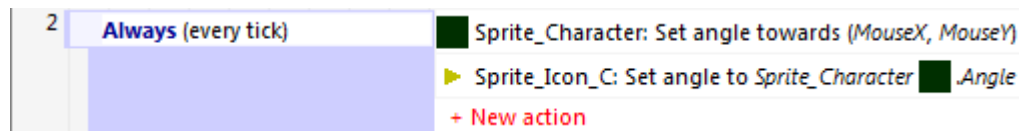


Теперь сделайте так, чтобы Главный Герой всегда смотрел в сторону Курсора. Сделайте так, чтобы угол иконки Главного Героя был такой же, как и у него самого.

+ *System: Always (every tick)*

-> *Sprite_Character: Set angle towards (MouseX, MouseY)*

-> *Sprite_Icon_C: Set angle to Sprite_Character.Angle*



Дальше нам нужно создать второстепенных персонажей вместе с принадлежащими ими иконками. Создайте событие – условие при клике мыши – создайте Sprite_Enemy, установите ему случайный угол и создайте Sprite_Icon_E, который прикрепите в пару к персонажу через Object Pairer.

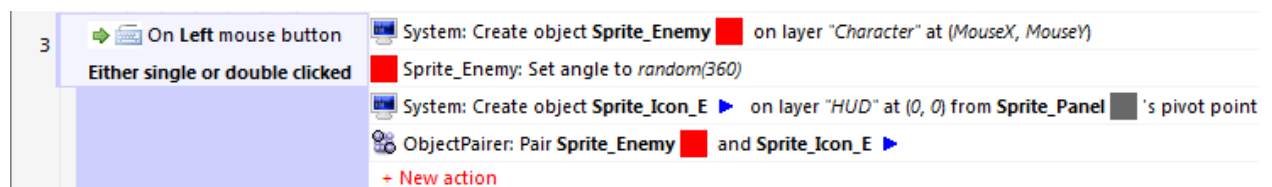
+ *MouseKeyboard: On Left mouse button Either single or double clicked*

-> *System: Create object Sprite_Enemy on layer "Character" at (MouseX, MouseY)*

-> *Sprite_Enemy: Set angle to random(360)*

-> *System: Create object Sprite_Icon_E on layer "HUD" at (0, 0) from Sprite_Panel's pivot point*

-> *ObjectPairer: Pair Sprite_Enemy and Sprite_Icon_E*



Теперь самая сложная часть. Нам нужно сделать формулу, по которой будут перемещаться иконки. Т. к. иконки, принадлежащие второстепенным персонажам, будут двигаться относительно центра – нам нужно вычислять дистанцию от Главного Героя к персонажу. $(Sprite_Enemy.XY - Sprite_Character.XY) / 10$ – это и будет наша формула. Число 10 играет роль радиуса. Чем оно меньше, тем быстрее второстепенные персонажи будут отдаляться от Главного Героя. Здесь нужно не забыть прибавить в конце позицию панели, чтобы иконки брали центр именно с неё, а не с начала координат.

Давайте разберём формулу и разберёмся, почему же я написал Enemy - Character, а не наоборот и почему я взял именно эти значения.

Как вы поняли, нам нужно найти то положение для иконки, которое соответствует положению Главного Героя относительно второстепенных персонажей. То есть, значения, которые нам нужны – это их позиции.

Представьте себе координатную прямую. На ней расположены Главный Герой и персонаж так, что персонаж стоит правее, то есть дальше Главного Героя. Если бы мы написали `Character - Enemy`, то мы бы получили отрицательное значение, потому что мы бы от меньшего отнимали большее. Но если центр это и есть Главный Герой, а `Enemy` на нашей прямой стоит правее Главного Героя, то при получении отрицательного значения на миникарте `Enemy` ушел бы влево, потому что отрицательные значения находятся по левую сторону.

Но если поставить в уравнении `Enemy - Character`, то всё выходит, как надо: от большего отнимаем меньшее, получается положительное число, и оно находится по правую сторону, там, где и должен находиться `Enemy` на миникарте.

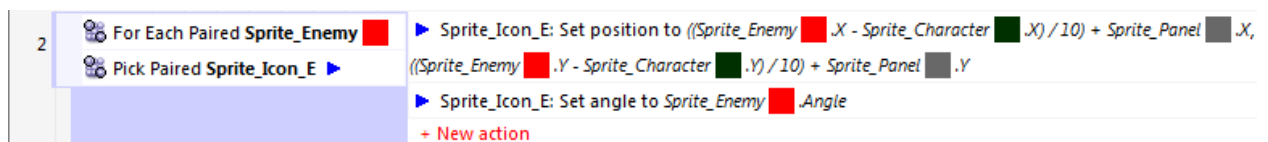
Создайте новое событие – 2 условия, для каждого в паре `Sprite_Enemy`, каждая пара `Sprite_Icon_E`. Установите позицию иконки второстепенного персонажа на позицию из выше сказанной формулы и установите её угол такой же, как и у Главного Героя. Так же это событие следует перетянуть и поставить перед событием `Always`, чтобы это событие выполнялось до того, как `Canvas` будет отрисовывать иконки второстепенных персонажей.

+ ObjectPairer: For Each Paired Sprite_Enemy

+ ObjectPairer: Pick Paired Sprite_Icon_E

-> Sprite_Icon_E: Set position to $((\text{Sprite_Enemy.X} - \text{Sprite_Character.X}) / 10) + \text{Sprite_Panel.X}$, $((\text{Sprite_Enemy.Y} - \text{Sprite_Character.Y}) / 10) + \text{Sprite_Panel.Y}$

-> Sprite_Icon_E: Set angle to `Sprite_Enemy.Angle`

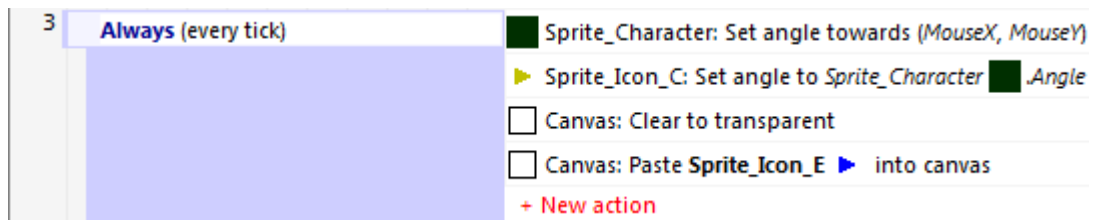


Теперь нам нужен `Canvas`. В событии `Always` добавьте действие, чтобы `Canvas` рисовал объект `Sprite_Icon_E`. И чтобы это дело не смазывалось, добавьте очищение Канваса, которое следует поставить перед рисованием.

+ System: Always (every tick)

-> Canvas: Clear to transparent

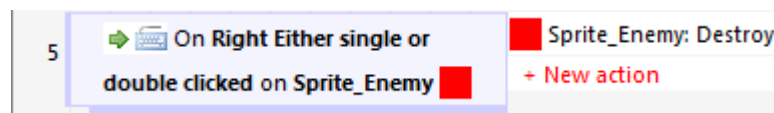
-> Canvas: Paste Sprite_Icon_E into canvas



Ну, и чтобы не было неудобств при удалении второстепенных персонажей и удалении их иконок – создайте еще одно событие. При клике правой кнопкой мыши по второстепенному персонажу – уничтожить персонажа.

+ MouseKeyboard: On Right Either single or double clicked on Sprite_Energy

-> Sprite_Energy: Destroy

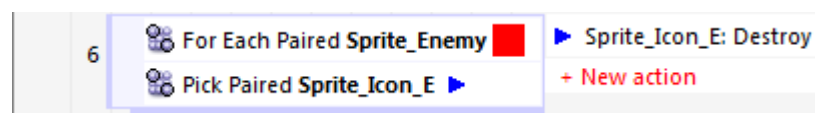


Но что же делать для удаления иконок? Создайте под-событие под Right click – 2 условия, для каждого в паре Sprite_Energy, каждая пара Sprite_Icon_E – уничтожить иконку персонажа.

+ ObjectPairer: For Each Paired Sprite_Energy

+ ObjectPairer: Pick Paired Sprite_Icon_E

-> Sprite_Icon_E: Destroy



Как вы поняли, Object Pairer работает почти как Container.

В итоге, по событиям:

1	Start of layout	<div>Canvas: Set position to object Sprite_Panel (image point 0)</div> <div>Sprite_Icon_C: Set position to object Sprite_Panel (image point 0)</div> <div>+ New action</div>
2	<div>For Each Paired Sprite_Enemy</div> <div>Pick Paired Sprite_Icon_E</div>	<div>Sprite_Icon_E: Set position to $((\text{Sprite_Enemy}.X - \text{Sprite_Character}.X) / 10) + \text{Sprite_Panel}.X, ((\text{Sprite_Enemy}.Y - \text{Sprite_Character}.Y) / 10) + \text{Sprite_Panel}.Y$</div> <div>Sprite_Icon_E: Set angle to Sprite_Enemy.Angle</div> <div>+ New action</div>
3	Always (every tick)	<div>Sprite_Character: Set angle towards (MouseX, MouseY)</div> <div>Sprite_Icon_C: Set angle to Sprite_Character.Angle</div> <div>Canvas: Clear to transparent</div> <div>Canvas: Paste Sprite_Icon_E into canvas</div> <div>+ New action</div>
4	On Left mouse button Either single or double clicked	<div>System: Create object Sprite_Enemy on layer "Character" at (MouseX, MouseY)</div> <div>Sprite_Enemy: Set angle to random(360)</div> <div>System: Create object Sprite_Icon_E on layer "HUD" at (0, 0) from Sprite_Panel's pivot point</div> <div>ObjectPairer: Pair Sprite_Enemy and Sprite_Icon_E</div> <div>+ New action</div>
5	On Right Either single or double clicked on Sprite_Enemy	<div>Sprite_Enemy: Destroy</div> <div>+ New action</div>
6	<div>For Each Paired Sprite_Enemy</div> <div>Pick Paired Sprite_Icon_E</div>	<div>Sprite_Icon_E: Destroy</div> <div>+ New action</div>

Вы можете настраивать свою миникарту как хотите, давать различные иконки окружающим объектам и указывать радиус обзора персонажа.

На этом я с вами прощаюсь, спасибо, что дочитали до конца и до следующих встреч!

Автор урока Navuhodonosor.