

# РЕЗУЛЬТАТЫ РАБОТЫ

После внедрения новых функций в систему анимации стало возможным:

- ✓ Убрать ненужные постоянные ресурсы, использовавшиеся для отображения временных, ограничить количество точек принятия решений лишь теми, которые необходимы для прогона модели
- ✓ Значительно сократить код во вкладке анимации, убрав дублирование, сделать код более читаемым

Вкладка RSS

```
$Resources
Баржа      : Баржи trace Погрузка 0 0 0 0 0 0
Погрузчик1 : Погрузчики trace * 5 8 1 0
Погрузчик2 : Погрузчики trace * 3 5 2 0
Разгрузчик : Погрузчики trace * 1 2 0 0
$End
```

Вкладка FRM

```
array<Контейнеры>A=Select(Контейнеры : Контейнеры.состояние != Заказан).getArray();
for ( i = 0; i < A.Size; i++ )
{
    ----- список активных контейнеров -----
    R = A[i].cR; G = A[i].cG; B = A[i].cB;
    text(contX + A[i].длина*cSize + 4, curY + A[i].ширина*0.5*cSize - 6, 10, 12, transparent, black, > A[i]. ширина);
    text(contX + A[i].длина*cSize + 12, curY + A[i].ширина*0.5*cSize - 6, 10, 12, transparent, black, > 'x');
    text(contX + A[i].длина*cSize + 20, curY + A[i].ширина*0.5*cSize - 6, 10, 12, transparent, black, > A[i].длина);
    text(contX + 4, curY + A[i].ширина*0.5*cSize - 6, 153, 12, transparent, black, > A[i].состояние);

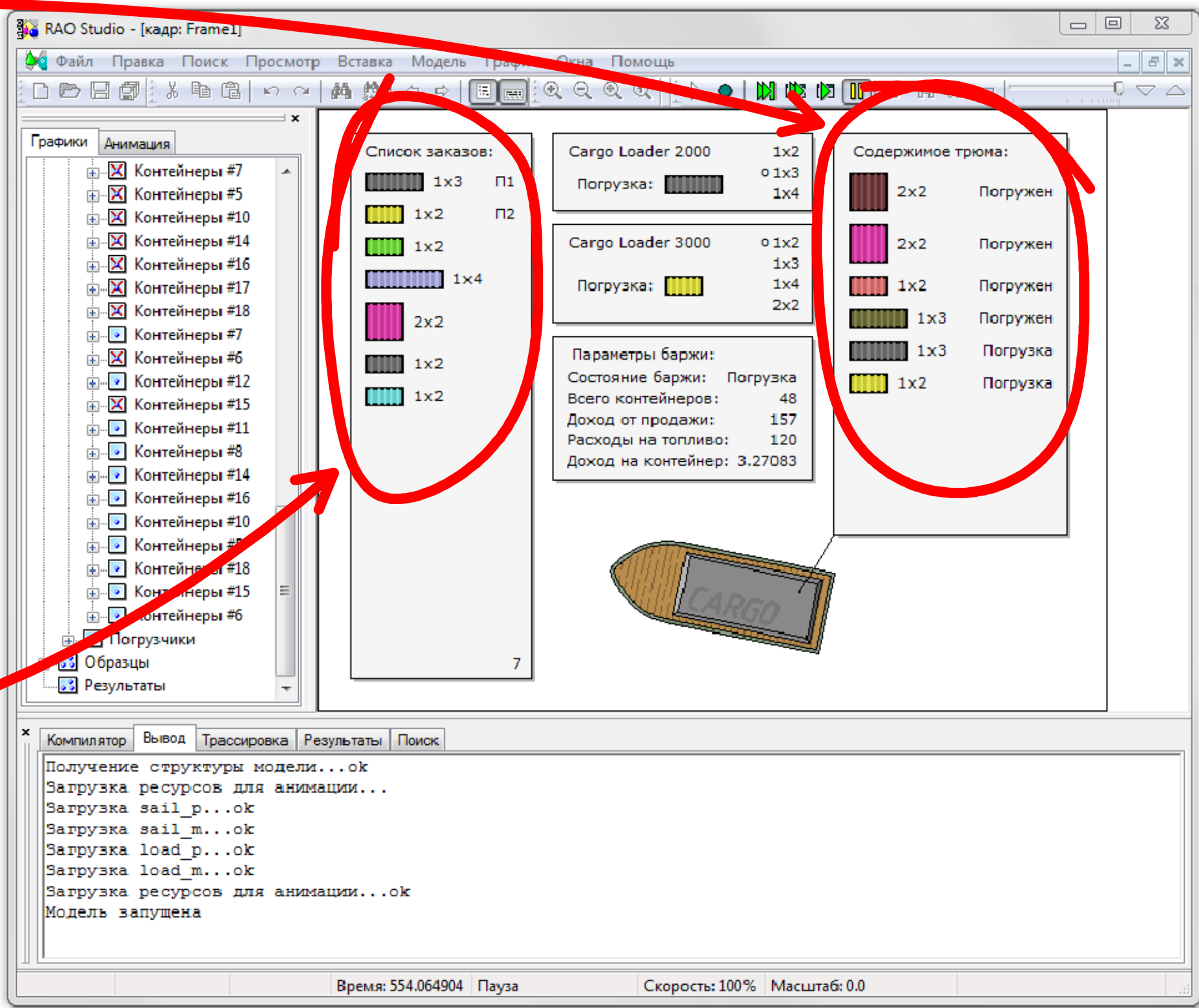
    rect (contX, curY, A[i].длина*cSize, A[i].ширина*cSize, <R, G, B>, black);
    rect (contX + 1, curY + 1, A[i].длина*cSize - 2, A[i].ширина*cSize - 2, <R, G, B>, black);
    for ( j = 0; j < A[i].длина*3 - 1; j++ )
        rect(contX + 5*(j + 1) - 1, curY + 1, 2, A[i]. ширина*cSize - 2, transparent ,<R + 30, G + 30, B + 30>);
    curY+= 10 + A[i].ширина*cSize ;
}
-----
```

```
array<Контейнеры> Z=Select(Контейнеры : Контейнеры.состояние == Заказан
                           or Контейнеры.состояние == Погрузка).getArray();
for ( i = 0; i < Z.Size; i++ )
{
    if (ordY + Z[i].ширина*cSize < by + 163)
    {
        ----- список заказов -----
        text(ordX + Z[i].длина*cSize + 4, ordY + Z[i].ширина*0.5*cSize - 6, 10, 12, transparent, black, > Z[i]. ширина);
        text(ordX + Z[i].длина*cSize + 12, ordY + Z[i].ширина*0.5*cSize - 6, 10, 12, transparent, black, > 'x');
        text(ordX + Z[i].длина*cSize + 20, ordY + Z[i].ширина*0.5*cSize - 6, 10, 12, transparent, black, > Z[i].длина);
        if (Z[i].погрузчик != 0)
        {
            text(ordX + 4, ordY + Z[i].ширина*0.5*cSize - 6, 104, 12, transparent, black, > 'П');
            text(ordX + 108, ordY + Z[i].ширина*0.5*cSize - 6, 10, 12, transparent, black, < Z[i]. погрузчик);
        }

        R = Z[i].cR; G = Z[i].cG; B = Z[i].cB;
        rect (ordX, ordY, Z[i].длина*cSize, Z[i].ширина*cSize, <R, G, B>, black);
        rect (ordX + 1, ordY + 1, Z[i].длина*cSize - 2, Z[i].ширина*cSize - 2, <R, G, B>, black);
        for ( j = 0; j < Z[i].длина*3 - 1; j++ )
            rect(ordX + 5*(j + 1) - 1, ordY + 1, 2, Z[i]. ширина*cSize - 2, transparent ,<R + 30, G + 30, B + 30>);
    }
}
```

...

Скриншот анимации модели



Глобальные переменные		
a только отображенные в список. Формат использования:		
Not_For_All ( <логическое выражение>   NoCheck )		
Empty	Возвращает значение ИСТИНА, если список отображенных ресурсов пуст, в противном случае - ЛОЖЬ. Формат использования:	Логическое
Empty ()		
Size	Возвращает размер списка ресурсов, целое число. Если список пуст, то возвращается ноль. Формат использования:	Арифметическое
Size ()		
getArray	Работает только в системе анимации. Возвращает массив с отображенными ресурсами, тип массива соответствует типу ресурса. Если список ресурсов пуст, возвращаемый массив также имеет нулевой размер. Формат использования:	Арифметическое
getArray ()   get_array ()		
Примеры		
Exist(Заявка : Заявки.Состояние = ожидает)		
Not_Exist(Тип_1 : NoCheck)		
Not_For_All(Станок : Станок.Количество > 3 and Станок.Состояние = свободен)		
Select (Заявка: Заявка.Время_прихода >= 480 and Заявка.Состояние = в_очереди).For_All(Заявка.Время_в_очереди > 20)		
array<Заявки> z = Select(Заявки: Заявки.Время_прихода >= 300).getArray();		

Также была расширена справочная документация по функции Select