

Министерство образования Российской Федерации

Томский государственный университет

Факультет информатики

Кафедра теоретических основ информатики

ДОПУСТИТЬ К ЗАЩИТЕ В ГАК

зав. кафедрой ТОИ, к. т. н.,

доцент _____ Ю. Л. Костюк

« » _____ 2002 г.

Кириллов Анатолий Борисович

ВИЗУАЛИЗАЦИЯ ТРЕХМЕРНЫХ МОДЕЛЕЙ РЕАЛЬНЫХ ОБЪЕКТОВ
ДОРОЖНОЙ ОТРАСЛИ: АВТОМОБИЛЬНЫХ ДОРОГ, СООРУЖЕНИЙ НА
НИХ И ДОРОЖНЫХ ЗНАКОВ

Дипломная работа

Научный руководитель:

Директор ИДЦ «Индор», к.т.н.

_____ В. Н. Бойков

Исполнитель:

студент гр. 1471

_____ А. Б. Кириллов

Электронная версия дипломной работы помещена в электронную
библиотеку: файл _____ Администратор _____ Дата _____

Томск – 2002

РЕФЕРАТ

Дипломная работа 50 с., 18 рис., 2 таблицы, 5 источников, 3 приложения.

СИСТЕМА ПРОЕКТИРОВАНИЯ АВТОМОБИЛЬНЫХ ДОРОГ ReCAD, ВИЗУАЛИЗАЦИЯ, ТРЕХМЕРНАЯ МОДЕЛЬ, МОДУЛЬ РАСШИРЕНИЯ, ДОРОЖНЫЕ ЗНАКИ, ПРОЕКТИРОВАНИЕ, ГОСТ 10807-78, DIRECT 3D.

- (1) Цель работы – разработать модуль визуализации трехмерного вида проекта для системы проектирования автомобильных дорог, а также модуль для создания и редактирования дорожных знаков и знаков индивидуального проектирования.
- (2) Область применения – система проектирования автомобильных дорог ReCAD.
- (3) Метод исследования – экспериментальный на ЭВМ.
- (4) Результаты – автором данной работы были изучены стандарты, касающиеся дорожных знаков, подготовлены текстуры с их изображениями, разработаны основные структуры данных для хранения знаков и их описаний в библиотеке, а также разработан модуль визуализации трехмерного вида проекта и редактор дорожных знаков в соответствии со всеми требованиями, указанными в техническом задании.

ОГЛАВЛЕНИЕ

РЕФЕРАТ	2
ВВЕДЕНИЕ	4
1. ЗНАКИ ДОРОЖНЫЕ. ОБЩИЕ СВЕДЕНИЯ, СТАНДАРТЫ	5
1.1. Группы, наименования, изображения	5
1.2. Размеры дорожных знаков	5
1.3. Параметры, наносимые на знаки	9
1.4. Знаки индивидуального проектирования.....	10
2. ТРЕХМЕРНЫЙ ВИД.....	15
2.1. Основные понятия и концепция Direct3D.....	15
3. РЕАЛИЗАЦИЯ	17
3.1. Библиотека дорожных знаков	17
3.2. Текстовая информация на знаках	18
ЗАКЛЮЧЕНИЕ.....	20
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	21
ПРИЛОЖЕНИЕ 1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	22
1. ТРЕХМЕРНЫЙ ВИД.....	22
1.1. Основные сведения	22
1.2. Запись видеоролика	23
1.3. Окно настроек 3D-вида	24
2. ДОРОЖНЫЕ ЗНАКИ	26
2.1. Окно редактора дорожных знаков.....	26
2.2. Окно редактора знаков индивидуального проектирования.....	27
ПРИЛОЖЕНИЕ 2. РУКОВОДСТВО ПРОГРАММИСТА	33
1. МОДУЛЬ ВИЗУАЛИЗАЦИИ ТРЕХМЕРНОГО ВИДА.....	33
1.1. View3DMainUnit.pas - основной модуль	33
1.2. View3DSetupUnit.pas – модуль настроек	38
1.3. View3DCodecUnit.pas – модуль настроек программы видеосжатия.....	39
2. МОДУЛЬ РЕДАКТОРА ДОРОЖНЫХ ЗНАКОВ.....	40
2.1. Znaks3D.pas – модуль расширения дорожных знаков (плагин)	40
2.2. View3DZnakUnit.pas – редактор дорожных знаков	43
2.3. IndividualSignUnit.pas – редактор знаков индивидуального проектирования	46
ПРИЛОЖЕНИЕ 3. АКТ ВНЕДРЕНИЯ	49

ВВЕДЕНИЕ

Существует немало специализированных компьютерных программ проектирования, призванных помочь инженерам в проектировании и автоматизации некоторых вычислительных процессов. К их числу относится система проектирования автомобильных дорог ReCAD, работу над которой ведет отдел программного обеспечения Инженерного дорожного центра «Индор». Немаловажную роль в этой системе играет компонент, который отвечает за визуализацию проекта в трехмерном виде. Он призван помочь разработчику увидеть и визуальнo оценить проект в целом: отснятую и оцифрованную модель местности, пути прохождения проектных трасс, развязки дорог, дорожное полотно с разметкой и другие дорожные элементы.

Перед автором данной работы были поставлены следующие цели: разработать модуль для визуализации трехмерного вида проекта, а также модуль редактора дорожных знаков и знаков индивидуального проектирования для их последующей трехмерной визуализации в системе проектирования автомобильных дорог ReCAD. Модуль редактора дорожных знаков должен предоставлять возможность выбора знаков, различных схем их размещения на одной стойке, а также возможность редактирования параметров тех знаков, у которых они присутствуют. Модуль редактора знаков индивидуального проектирования должен производить автоматическое размещение указанных элементов на щите дорожного знака в соответствии с нормами и стандартами, описанными в ГОСТ 10807-78 «Знаки дорожные: общие технические условия».

1. ЗНАКИ ДОРОЖНЫЕ. ОБЩИЕ СВЕДЕНИЯ, СТАНДАРТЫ

Стандарт ГОСТ 10807-78 распространяется на дорожные знаки, предназначенных для информирования участников дорожного движения об условиях и режимах движения на дорогах и улицах.

1.1. Группы, наименования, изображения

Стандарт устанавливает семь групп дорожных знаков:

- 1 – предупреждающие;
- 2 – приоритета;
- 3 – запрещающие;
- 4 – предписывающие;
- 5 – информационно-указательные;
- 6 – сервиса;
- 7 – дополнительной информации (таблички).

Номер знака состоит из номера группы, порядкового номера знака в группе, порядкового номера разновидности (при наличии), разделенных между собой точками.

Изображения дорожных знаков (кроме знаков индивидуального проектирования) 5.20.1, 5.20.2, 5.21.1-5.27, 5.31 приведены в приложении 1.

Надписи на знаках (буквы, цифры, знаки препинания) должны выполняться специальным шрифтом. Начертания символов можно посмотреть в настоящем ГОСТе, здесь лишь стоит отметить, что при программной реализации использовался TrueType шрифт RoadSign.ttf, который по начертанию полностью соответствует приведенному в ГОСТ.

1.2. Размеры дорожных знаков

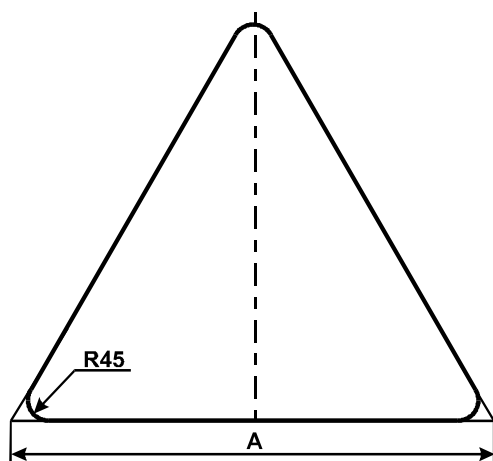
Дорожные знаки должны изготавливаться четырех типоразмеров: I – малого, II – нормального, III – большого, IV – очень большого.

Типоразмеры знаков в зависимости от условий применения должны выбираться в соответствии с таблицей 1.

Таблица 1.

Типоразмер знака	Применение знака	
	Вне населенных пунктов	в населенных пунктах
I	Дороги с одной полосой	Дороги местного назначения
II	Дороги с двумя и тремя полосами	Магистральные дороги
III	Дороги с четырьмя и более полосами и автомагистрали	Скоростные дороги
IV	Ремонтные работы на автомагистралях, опасные участки на других дорогах при обосновании целесообразности применения	

Размеры изображения знаков по типоразмерам должны соответствовать приведенным на рис. 1 – 8 (все размеры указаны в миллиметрах).



Номер знака	Типоразмер	A
1.1, 1.2, 1.5-1.7, 1.9-1.14, 1.16, 1.20-1.22, 1.24-1.30, 2.3, 2.4	I	700
	II	900
	III	1200
1.8, 1.15, 1.17-1.19, 1.23	I	700
	II	900
	III	1200
	IV	1500

Рис. 1. Треугольные знаки

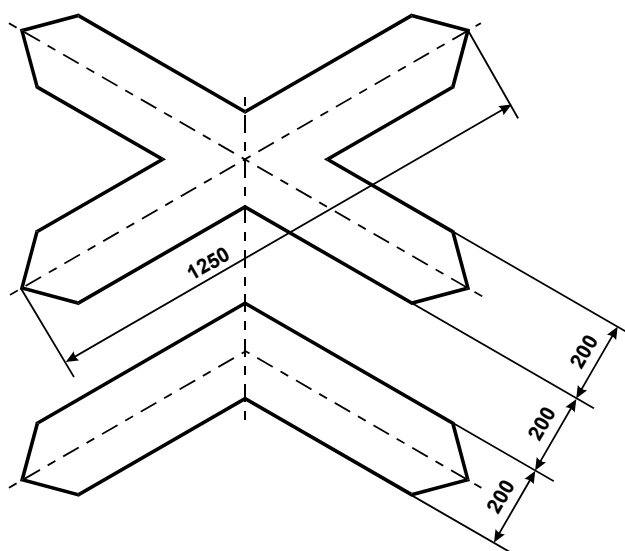
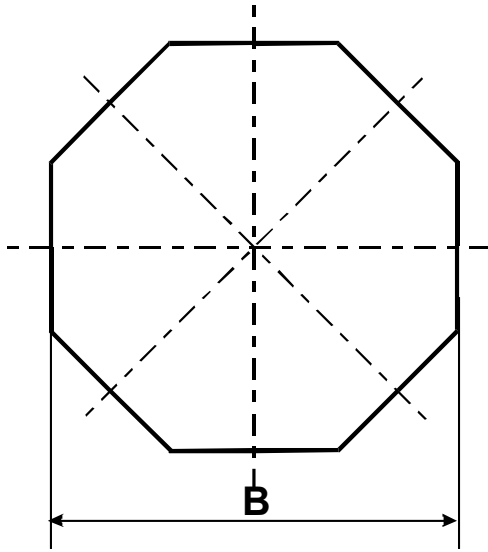
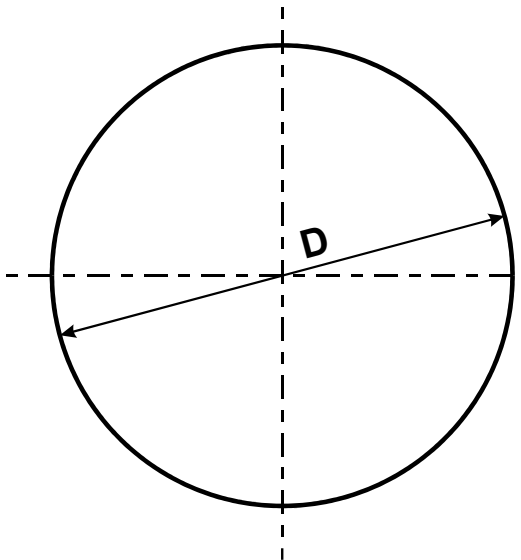


Рис. 2. Знаки 1.3.1, 1.3.2



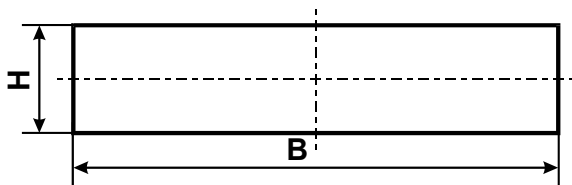
Номер знака	Типоразмер	В
2.5	II	700
	III	900
	IV	1200

Рис. 3. Знак 2.5



Номер знака	Типоразмер	В
2.6, 3.1-3.9, 3.11-3.16, 3.18.1-3.19, 3.21-3.23, 3.25-4.4	I	600
	II	700
	III	900
3.10, 4.5, 4.6	II	700
3.17.1, 3.17.2, 4.7, 4.8	II	700
	III	900
3.20, 3.24	I	600
	II	700
	III	900
	IV	1200

Рис. 4. Круглые знаки



Номер знака	Типоразмер	Н	В
1.31.1-1.31.3	II	500	2250
	III	700	3150

Рис. 5. Знаки 1.31.1-1.31.3

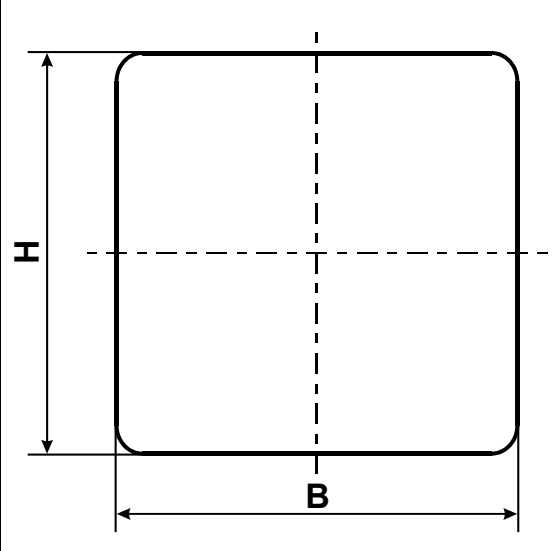
	Номер знака	Типоразмер	В	R
	2.1, 2.2, 2.7, 5.5, 5.6, 5.15-5.16.2, 5.18- 5.19.3, 7.13	I	600	45
		II	700	
		III	900	
	5.8.2-5.8.6, 5.9-5.11.2, 5.35-5.37, 7.1.2	II	700	
		III	900	
	5.17.1-5.17.4	I	600	
		II	700	
5.20.3	-	>1200		
5.29.1	II	350		

Рис. 6. Квадратные знаки

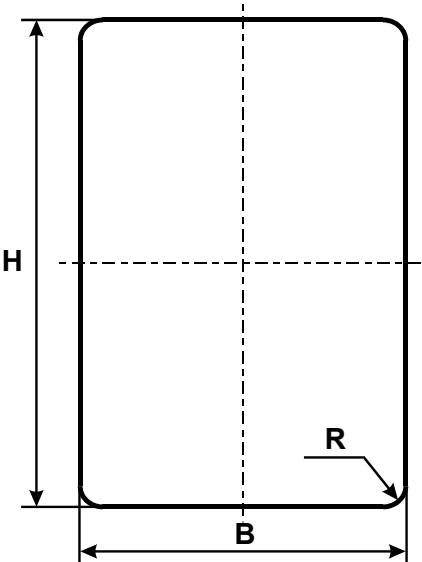
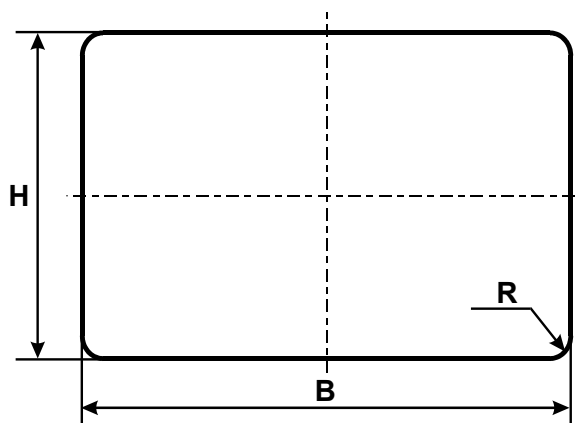
	Номер знака	Типоразмер	Н	В	R
	5.1-5.4	III	1350	900	45
	5.12-5.14, 5.38, 5.39	I	900	600	
	6.1-6.12	I	900	600	
		II	1050	700	
		III	1350	900	
5.34.1, 5.34.2	IV	2000	1500		

Рис. 7. Вертикальные прямоугольные знаки



Номер знака	Типоразмер	Н	В	Номер знака	Типоразмер	Н	В
1.4.1-1.4.6, 7.1.3, 7.1.4, 7.2.2-7.11, 7.14-7.18	I	300	600	5.7.1, 5.7.2, 5.33 5.30.1-5.30.3	II	350	1050
	II	350	700		III	450	1350
	III	450	900	5.29.2, 5.32.1-5.32.3	II	350	1050
7.1.1, 7.2.1, 7.12	I	300	600	5.8.1, 5.8.7, 5.8.8	II	700	1400
	II	350	700		III	900	1800
	III	400	900	5.28	II	200	300
	IV	600	1200	5.29.1, 5.29.2	II	350	700

Рис. 8. Горизонтальные прямоугольные знаки

На знаках 1.1-2.4, 2.6, 3.2-3.31, 5.28, 5.30.1-5.30.3, 5.31.1-5.34.2 ширина наружной каймы должна быть 10 мм, а на знаках 2.5, 2.7, 3.1, 4.1-5.19.3, 5.20.3, 5.35-5.39, 6.1-6.12 – 22 мм. Внутренний радиус закругления красной каймы на знаках 1.1, 1.2, 1.5-1.30, 2.3.1-2.3.3, 2.4 должен составлять 10 мм.

1.3. Параметры, наносимые на знаки

Значения параметров, наносимых на знаки 1.13, 1.14, 3.4, 3.11-3.16, 3.24, 3.25, 4.7, 4.8, 5.8.3, 5.11.2, 5.18, 5.20.1, 5.21.1, 5.21.2, 5.27, 7.1.1-7.2.2, 7.2.5, 7.2.6, 7.5.3-7.5.7, 7.9, 7.11, должны выбираться из приведенных в таблице 2.

Таблица 2.

Номер знака	Наименование параметра	Значение параметра
1.13, 3.14	Линейные величины	Кратное 0,1 м
3.15		Кратное 1,0 м
3.16, 5.11.2, 7.1.2		10, 15, 20, 35, 30, 40, 50, 60, 70, 80, 100, 120, 150, 200, 250, 300 м
5.20.1, 7.1.1, 7.1.3, 7.1.4, 7.2.1, 7.2.2, 7.2.5, 7.2.6		10, 15, 20, 35, 30, 40, 50, 60, 70, 80, 100, 120, 150, 200, 250, 300, 350, 400, 500, 600, 700, 800, 900 м; в диапазоне от 1 до 10 км – кратное 0,1 км, 10 км и более – кратное 1,0 км
5.21.1, 5.21.2, 5.27		В диапазоне от 0 до 10 км – кратное 0,1 км, 10 км и более – кратное 1,0 км (размерность км не указывается)

3.24, 3.25, 4.7, 4.8, 5.8.3, 5.18	Скорость	Кратное 10 км/ч (разметность км/ч не указывается)
3.4, 7.11	Масса	2,4; 8; 10; 15; 20; 25 т
3.11		1,5; 2 ; 2,5; 3; 3,5; 4; 5; 6; 7; 8; 9; 10; 12; 15; 8; 20 т и далее кратные 5 тонн
3.12	Сила тяжести	1,5; 2; 2,5; 3; 3,5; 4; 5; 6; 7; 8; 9; 10 тс
7.5.3, 7.5.7	Время	Дни недели
7.5.4 – 7.5.7		Время суток, кратное 30 мин (23.00; 23.30; 24.00; 24.30; 1.00 и т.д.)
7.9		30 мин; 1 ч; 2 ч
1.13, 1.13	Уклон	Кратное 1%

1.4. Знаки индивидуального проектирования

На знаках 5.20.1, 5.20.2, 5.21.1, 5.21.2, 5.27, 5.29.1 и 5.29.2 для каждого из направлений должно быть указано не более трех названий населенных пунктов, других объектов или номеров маршрута.

Фон знаков 5.20.1 и 5.20.2 должен быть: зеленого цвета на знаках, предназначенных для установки на автомагистралях, синего цвета - на других дорогах вне населенных пунктов, белого цвета - в населенных пунктах. При этом, если на знаках, предназначенных для установки в населенных пунктах, указывают только названия других населенных пунктов или объектов, движение к которым осуществляется по автомагистрали или другой дороге, фон знаков должен быть соответственно зеленого или синего цвета.

На знаках, предназначенных для установки на автомагистралях, надпись, содержащая названия населенных пунктов или объектов, движение к которым осуществляется не по автомагистрали, должна быть выполнена на синем фоне, а предназначенных для установки на участке автомагистрали в пределах населенного пункта наименования объектов этого населенного пункта - на белом фоне.

На знаках, предназначенных для установки на дорогах, кроме автомагистралей, надпись, содержащая названия населенных пунктов или объектов, движение к которым осуществляется по автомагистрали, должна быть выполнена на зеленом фоне. При указании только объектов, находящихся в граничащем с дорогой населенном пункте, надпись должна быть выполнена на белом фоне.

На знаках с белым фоном надпись, содержащая названия других населенных пунктов или объектов, движение к которым должно осуществляться по автомагистрали или другой

дороге (не автомагистрали), должна быть выполнена на вставке соответственно зеленого или синего цвета.

На знаках (с зеленым фоном), предназначенных для установки на автомагистралях, надпись, содержащая названия населенных пунктов или объектов, движение к которым осуществляется по другой дороге (не автомагистрали) или находящихся в граничащем населенном пункте, должна быть выполнена на вставке с соответственно синим или белым фоном.

На знаках (с синим фоном), предназначенных для установки на других дорогах, надпись, содержащая названия населенных пунктов или объектов, движение к которым осуществляется по автомагистрали или находящихся в граничащем населенном пункте, должна быть выполнена на вставке с соответственно зеленым или белым фоном.

Вставки должны выполняться без каймы, за исключением синих или зеленых вставок соответственно на зеленом или синем фоне.

При указании нескольких направлений они должны даваться в последовательности (сверху - вниз): прямо, налево, направо. При указании одного направления знаки (части знаков), выполненные на фоне разного цвета, должны даваться в последовательности (сверху - вниз): зеленый, синий, белый.

Знаки 5.21.1 и части знаков 5.21.2 должны иметь зеленый фон, если движение к указанным на них населенным пунктам или объектам осуществляется по автомагистрали, синий, если движение осуществляется по другим дорогам, и белый, если указанные объекты расположены в населенном пункте.

Знаки 5.24-5.29.2, предназначенные для установки на автомагистралях, должны иметь зеленый фон, а предназначенные для установки на всех остальных дорогах, включая дороги в населенных пунктах, - синий. Знаки 5.29.1 и 5.29.2 с буквой «Е» должны иметь только зеленый фон.

Компоновочные размеры изображений знаков и надписей на них должны определяться высотой прописной буквы h_p , которую в зависимости от места установки знака выбирают из ряда: 75, 100, 150, 200, 300, 400, 500 мм.

Имена собственные в названиях объектов должны выполняться прописными буквами, а служебные (поясняющие) слова при них - строчными (например, площадь МИРА, музей А.С.ПУШКИНА, аэропорт ВНУКОВО). При самостоятельном употреблении служебные слова следует выполнять прописными буквами (например, МУЗЕЙ, АЭРОПОРТ).

Высота прописной буквы h_p на знаках 5.20.1, 5.20.2, 5.21.1-5.27, предназначенных для установки вне населенных пунктов, должна быть равна: 400 или 500 мм - на автомагистралях, 300 или 400 мм - на дорогах с четырьмя и более полосами (кроме

автомагистралей), 200 или 300 мм – на дорогах с двумя и тремя полосами, 100 или 150 мм – на дорогах с одной полосой, а предназначенных для установки в населенных пунктах должна быть равна: 200 или 300 мм - на скоростных дорогах, 150 или 200 мм - на магистральных дорогах, 100 или 150 мм - на других дорогах, 75 или 100 мм - для построения городской подсистемы ориентирования на местной улично-дорожной сети.

Высота прописной буквы h_p на знаках 5.29.1 и 5.29.2 должна составлять 150 мм, а на знаке 5.31 должна быть равна: 200 мм - для знаков, предназначенных для установки на дорогах вне населенных пунктов, и 100 мм - в населенных пунктах.

Предпочтительно использовать больший шрифт из двух, установленных для данной дороги. Надписи, относящиеся ко второстепенным объектам, допускается выполнять меньшим шрифтом.

Для надписей, содержащих более 10 элементов в строке (за элемент принимают букву, цифру, стрелку, символ, изображение какого-либо знака), допускается применять меньший размер шрифта, расположение надписи в две строки или перенос слов, сокращение часто употребляемых отдельных слов в именах собственных.

Ширина каймы на знаках должна быть равна $0,12 h_p$, внутренний радиус закругления каймы - $0,3 h_p$. Ширина наружной каймы на знаках 5.20.1, 5.20.2, 5.21.1 и 5.21.2 с белым фоном, а также на знаках 5.22, 5.23 и 5.31 должна быть равна $0,06 h_p$.

На знаке 5.21.2 ширина линии, разделяющей надписи, относящиеся к разным направлениям, должна быть равна $0,1 h_p$. Надписи на белом и синем (или зеленом) фоне разделяться линией не должны.

Ширина каймы на знаках 5.29.1 и 5.29.2 должна быть равна 18 мм, внутренний радиус закругления каймы - 45 мм.

Расстояние по горизонтали и вертикали между словами, числами, стрелками, цветными вставками, каймой знака или вставки, линией, которая разделяет надписи, относящиеся к разным направлениям, символами, изображениями каких-либо знаков должно быть не менее $0,3 h_p$. Предпочтительное расстояние между строками разных надписей, относящихся к одному направлению, должно составлять от $0,4$ до $0,8 h_p$, а для двустрочной надписи одного наименования - $0,4 h_p$.

Для знака 5.20.1 расстояние между надписями, относящимися к разным направлениям, не должно быть менее $2,0 h_p$. Допускается уменьшение этого расстояния до $1,0 h_p$, если границы надписей, расположенных одна под другой, не совпадают.

Если на знаке используют два размера шрифта, то для расчета размеров каймы знака и элементов изображения, относящихся к главным объектам, а также расстояния между ними и надписями, соответствующими второстепенным объектам, применяют больший размер

шрифта. Размеры элементов изображения, относящихся ко второстепенным объектам, определяют в этом случае по меньшему размеру шрифта.

Ширина каймы вставок на знаках 5.20.1 должна быть равна $0,1 h_{п}$.

Высота цифр знака 5.29.1, изображение которого используют на других знаках или вставках, должна быть равна $h_{п}$, принятой для надписей на этих знаках. При этом ширина обрамляющей каймы должна быть равна $0,1 h_{п}$, внутренний радиус закругления каймы $-0,15 h_{п}$, внешний вертикальный размер изображения знака $-1,5 h_{п}$.

Высота символа автомагистрали или аэропорта должна составлять от $1,0$ до $1,5 h_{п}$ для однострочной надписи и от $2,0$ до $2,5 h_{п}$ - для двустрочной надписи названия одного населенного пункта или объекта. Изображения символов должны соответствовать символам знаков 1.28 и 5.1.

Символы автомагистрали или аэропорта на знаках 5.20.1, 5.20.2 и 5.21 должны располагаться слева от названия населенного пункта или объекта. На знаках 5.20.1 и 5.20.2 при наличии изображения знака 5.29.1, относящегося к данному населенному пункту или объекту, символы относительно названия населенного пункта или объекта должны располагаться в стороне, противоположной изображению знака 5.29.1.

На знаках 5.20.1, 5.20.2 и 5.21 для обозначения объектов допускается наносить пиктограммы, которые должны размещаться в месте, предназначенном для символа автомагистрали, аэропорта или изображения знака 5.29.1. Высота пиктограммы должна быть от $1,0 h_{п}$ до $1,5 h_{п}$.

Стрелки на знаках должны выполняться в соответствии с рис. 9.

На знаках 5.20.2 и 5.21.2 стрелки должны быть длиной $L = 2,3 h_{п}$ и располагаться симметрично относительно верхней и нижней каймы (линии, разделяющей надписи). При вертикальном расположении стрелки допускается уменьшение ее длины за счет стержня до $2,0 h_{п}$.

На знаках 5.20.1 и 5.31 длина стрелок должна выбираться из компоновочных соображений, ширину стрелок для второстепенных направлений допускается уменьшать на 30 % по отношению к стрелке основного направления.

На знаках 5.20.2 при указании наименований нескольких пунктов маршрута допускается увеличение размера стрелки при сохранении пропорций.

На знаке 5.29.2 стрелка должна быть длиной $L = 240$ мм (высота оголовка стрелки 138 мм, размер стороны оголовка стрелки 160 мм, толщина ножки стрелки 60 мм) и располагаться симметрично на поле.

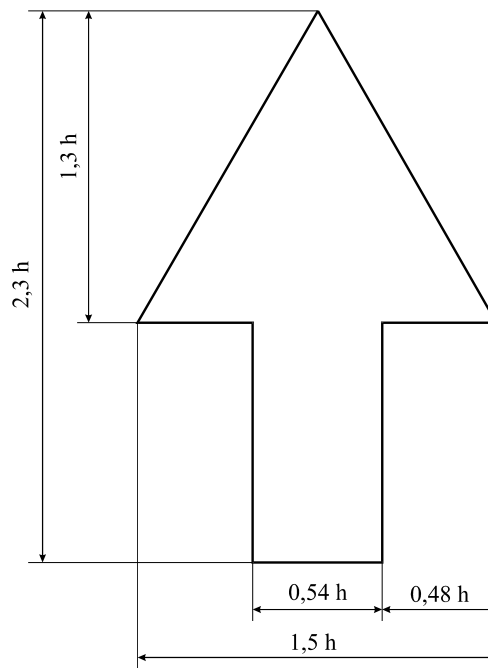


Рис. 9. Чертеж стрелки

Расстояние по горизонтали между началом внутренней границы оголовка знака и надписью, числом, символом автомагистрали или аэропорта на знаке 5.21.1 должно составлять от $0,15$ до $0,5 h_{п.}$

На знаке 5.20.1, в нижней части, должно указываться расстояние от места установки знака до пересечения. Высота цифр при этом должна соответствовать ближайшему меньшему значению $h_{п.}$ из приведенного выше ряда.

При нанесении на знаках 5.20.1, 5.20.2, 5.21, 5.27 наименований нескольких пунктов маршрута, соответствующих одному направлению движения, первым сверху указывают пункт, ближайший к месту установки знака.

Числа на знаках 5.21.1, 5.27 и отдельных частях (полях) знака 5.21.2, указывающие расстояние от места установки знака до названного пункта, должны размещаться справа от надписи, при этом цифры, выражающие одинаковые разряды чисел, должны находиться друг под другом.

На знаках 5.20.1 и 5.31 допускается наносить изображения других знаков, информирующих участников движения об особенностях маршрута или режима движения. При этом наибольший габаритный размер наносимых изображений должен составлять от $3,0$ до $5,0 h_{п.}$

Ширина красной полосы на знаках 5.23 и 5.25 должна быть равна $0,4 h_{п.}$ Расстояние по горизонтали между началом или концом полосы и каймой должно быть равно $h_{п.}$

2. ТРЕХМЕРНЫЙ ВИД

2.1. Основные понятия и концепция Direct3D

Microsoft Direct3D разработан для создания игр и трехмерной графики для операционной системы Windows. Это интерфейс, который предоставляет доступ к аппаратно-зависимой видеоаппаратуре, реализующей функции работы с трехмерными изображениями, как будто вы работаете с аппаратно-независимой системой.

Direct3D позволяет приложениям аппаратно-независимый способ доступа для взаимодействия с аппаратным видеоускорителем на низком уровне.

Direct3D обладает следующими особенностями:

- переключаемые z-буферы;
- сплошная закрашка закрашка Гуро;
- множество источников освещения нескольких типов;
- поддержка текстур;
- трансформация и отсечение;
- аппаратная независимость;
- полная поддержка для Windows 95, Windows 98/ME, Windows 2000/XP;
- поддержка архитектуры MMX, SSE и 3DNow!;
- работа в оконном и полноэкранным режимах;
- одновременный доступ к стандартной видеопамяти.

Direct3D может работать как с левосторонней, так и с правосторонней системой координат.

Естественными операциями при работе с трехмерными объектами являются операции переноса, вращения и масштабирования. В Direct3D это делается при помощи соответствующих матриц.

Рисование трехмерных объектов происходит при использовании различных примитивов. Самым простым примитивом является список точек с трехмерными координатами. Часто используются такие примитивы как полигоны. Наиболее простым полигоном является треугольник. Именно треугольники быстро отрисовываются аппаратным ускорителем, поэтому фигура любой сложности может быть представлена в виде множества связанных треугольников.

Для придания реалистичности необходимым условием является использование источников света. Их количество и тип выбираются в зависимости от сложности поставленной задачи и требований, предъявляемых к быстродействию.

Direct3D предоставляет функции для выделения буферов как в обычной оперативной памяти, так и в видеопамяти графического ускорителя.

В Direct3D существует такое понятие как «меш». Меш – это набор треугольников, сгруппированных по тем или иным признакам. Автор данной работы все трехмерные объекты представлял именно при помощи мешей. Меш задается при помощи двух буферов: вертексного (буфер вершин) и индексного (буфер индексов). В вертексном буфере перечисляются все точки со своими свойствами. Далее в индексном буфере тройками перечисляются индексы тех точек, которые образуют грань треугольника. Над мешем можно произвести функцию оптимизации (упорядочивание точек и индексов) с целью увеличения быстродействия отрисовки.

3. РЕАЛИЗАЦИЯ

Реализация выполнялась в среде программирования Borland Delphi 5 с использованием Direct3D v.8.

3.1. Библиотека дорожных знаков

Для начала были подготовлены текстуры с изображениями дорожных знаков в векторном формате (WMF) в Corel Draw 7.0, которые далее можно было масштабировать до любого размера. Далее был создан текстовый файл описания знаков. Каждая строка этого файла содержит описание одного знака и имеет следующую структуру:

- 1) Номер знака по ГОСТ 10807-78;
- 2) Тип знака (квадратный, круглый, треугольный и т.д.);
- 3) Наименование знака – текстовое описание знака;
- 4) Флаг присутствия параметров. Если здесь стоит true, то далее будет идти описание параметров текстовой информации, наносимой на знак, иначе конец описания знака;
- 5) Угол под которым будет выводиться надпись;
- 6) Ориентация надписи (Horiz – горизонтально, Vert - вертикально);
- 7) Размер шрифта для вывода надписей в пикселях;
- 8) Коэффициент уменьшения шрифта для вывода маленьких букв;
- 9) Цвет выводимой надписи;
- 10) X - координата для вывода надписи;
- 11) Y - координата для вывода надписи;
- 12) Текст по умолчанию.

Подробное описание параметров текстовой информации, наносимой на знак, будет приведено ниже. Здесь лишь можно отметить, что автором работы были выделены следующие типы знаков:

- TYPE_TRIANGLE – треугольный знак
- TYPE_TRIANGLE2 – перевернутый треугольный знак (знак «Уступите дорогу»);
- TYPE_CIRCLE – круглый знак;
- TYPE_QUAD – квадратный знак;
- TYPE_STOP – знак “Движение без остановки запрещено”;
- TYPE_MAINROAD – знак в форме ромба;
- TYPE_TURN – знаки “Направление поворота”;

- TYPE_RAILWAY – знак “Однопутная железная дорога”;
- TYPE_RAILWAY2 - знак “Многопутная железная дорога”;
- TYPE_VERTRECT, TYPE_RECT 1 - TYPE_RECT7
– различные по размерам знаки прямоугольной формы со скруглением углов;

За каждым из приведенных выше типов закреплена структура, содержащая высоту, ширину, длину ребра (для треугольных знаков) и радиус (для круглых знаков) для всех категорий знаков от I до III. Отсюда, при указании имени знака мы получаем его тип, а в соответствии с типом и категорией получаем необходимые размеры.

Графические файлы с изображениями знаков и файл описания затем компилируются в готовую библиотеку дорожных знаков (содержащую более 200 элементов), которая оформлена в виде DLL файла.

3.2. Текстовая информация на знаках

Как было отмечено в параграфе 1.3, некоторые знаки содержат параметры. Если в описании знака установлен признак присутствия параметров, то при создании текстуры с изображением дорожного знака на нее еще также будет нанесен текст. Поясним подробнее значение некоторых параметров.

“Ориентация надписи”. Может быть горизонтальная (см. рис. 10а) или вертикальная (см. рис. 10б).

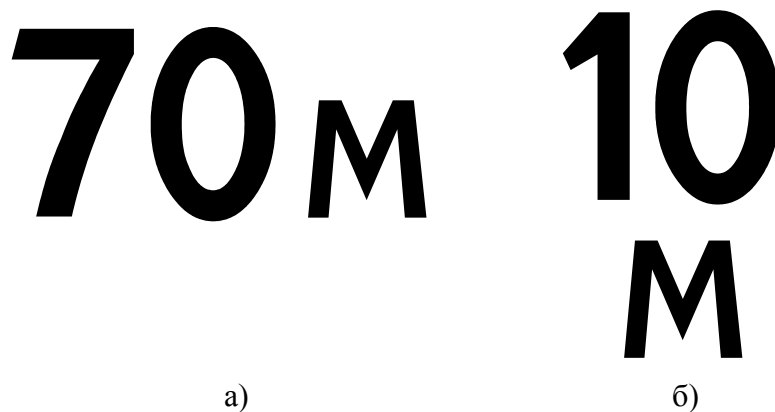


Рис. 10. Ориентация надписей

“Коэффициент уменьшения шрифта для вывода маленьких букв”. Проблема здесь обстоит в следующем: отношение высоты символа размерности и высоты символа номинала (см. рис. 11) для разных знаков различны. Так, например, на рисунке 11а это соотношение составляет 0.6, а на рисунке 11б – 0.5.



Рис. 11. Размеры надписей

Поэтому для вывода надписей используется два шрифта: один обычный (для вывода номинала), другой уменьшенный с учетом коэффициента (для вывода размерности). Используется посимвольный вывод с учетом прозрачности символьной матрицы, т.е. выводится сам символ без обрамляющего его прямоугольника. Способ вывода и форматирования надписи зависит от номера знака и осуществляется автоматически. Таким образом, пользователь вводит однострочный текст, который затем автоматически разбивается на необходимое число строк, в зависимости от содержания выбираются необходимые размеры шрифтов, производится центрирование и вывод на изображение знака. Так мы получаем законченную текстуру, готовую к применению.

ЗАКЛЮЧЕНИЕ

В ходе работы автором была изучена предметная область и стандарты, касающиеся дорожных знаков (ГОСТ 10807-78), подготовлены текстуры с их изображениями в векторном формате, разработаны основные структуры данных для хранения знаков и их описаний, создана библиотека дорожных знаков в виде DLL файла. В результате был разработан модуль для визуализации трехмерного вида проекта в системе проектирования автомобильных дорог ReCAD. Также был разработан модуль для создания и редактирования дорожных знаков, оформленный в виде модуля расширения (плагина), в соответствии со всеми требованиями, указанными в техническом задании. Обеспечена возможность установки созданных знаков в указанные координаты, как абсолютные, так и с привязкой к указанной трассе. По просьбе заказчика также добавлена возможность редактирования текстовой информации, которую могут содержать некоторые знаки. Функциональность редактора расширена на случай использования знаков индивидуального проектирования. Была обеспечена возможность отображения созданных знаков как некоторого подмножества трехмерных объектов для 3D-вида системы.

Реализация производилась в среде программирования Borland Delphi 5, с использованием графических возможностей Direct 3D v.8.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Н. Томпсон. Секреты программирования трехмерной графики для Windows 95 – Санкт-Петербург: Питер, 1997. – 340с.
2. DirectX 8.0 Programmer's Reference by Microsoft.
3. Microsoft Software Development Kit - January 2001.
4. ГОСТ 10807-78. Знаки дорожные. Общие технические условия – Москва, 1999. – 136с.
5. Указания по применению дорожных знаков – Москва: Транспорт, 1984. – 112с.

ПРИЛОЖЕНИЕ 1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

1. ТРЕХМЕРНЫЙ ВИД

1.1. Основные сведения

После открытия файла проекта системы ReCAD, нажмите «Ctrl+4» или выберите пункт меню «Окно – 3D-вид». При этом появится окно 3D-вида, которое позволяет увидеть объемное изображение существующей поверхности (см. рис. 12).

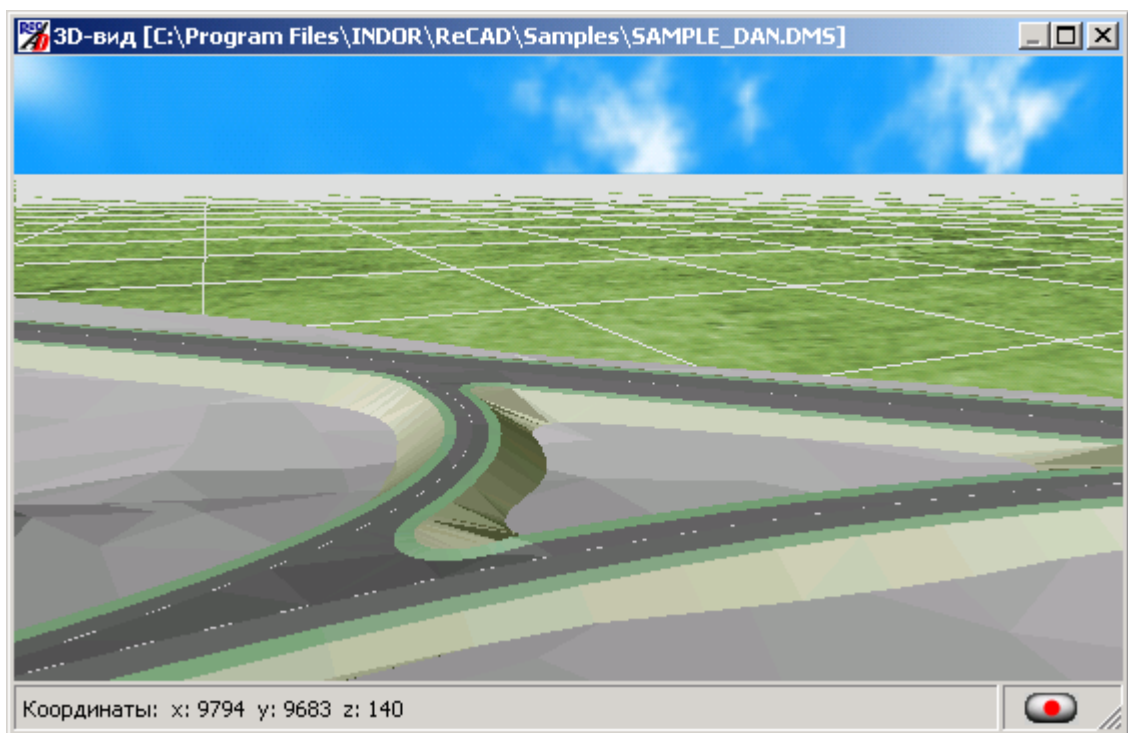


Рис. 12. Окно 3D-вида

Отображение происходит в реальном времени, т.е. все изменения в проекте (перестройка триангуляции, добавление/удаление трасс, изменение цвета поверхности и т.д.) пользователь сразу же увидит на экране. Для придания большей реалистичности отображается небо и земляное покрытие.

Пользователю предоставляется возможность свободного перемещения по поверхности в любом направлении. Если в проекте существуют трассы, то можно перейти в режим проезда по текущей выбранной трассе (как будто Вы едите по дороге на автомобиле).

Клавиши управления движением следующие:

UP, MOUSE_LEFT_BUTTON



- вперед

DOWN, MOUSE_RIGHT_BUTTON	- назад
LEFT	- налево
RIGHT	- направо
CTRL+UP	- вверх
CTRL+DOWN	- вниз
CTRL+LEFT, CTRL+MOUSE_LEFT	- поворот налево
CTRL+RIGHT, CTRL+MOUSE_RIGHT	- поворот направо
SHIFT+UP, SHIFT+MOUSE_FORWARD	- посмотреть вверх
SHIFT+DOWN, SHIFT+MOUSE_BACKWARD	- посмотреть вниз
HOME	- переместиться в начало трассы
END	- переместиться в конец трассы
PAGE_UP	- вперед по трассе
PAGE_DOWN	- назад по трассе

Область перемещения ограничена существующей поверхностью. Отсутствует возможность опуститься ниже поверхности.

Все параметры перемещения (шаг, угол поворота и наклона), а также некоторые другие настройки можно задать, вызвав окно настроек путем выбора пункта меню «Сервис – Настройка 3D-вида...». Значение всех параметров настроек будет приведено ниже.

1.2. Запись видеоролика

Интересной особенностью является запись проезда по трассе в .AVI файл с использованием любой установленной в системе программой видеосжатия. Делается это нажатием кнопки «Запись видеоролика» , расположенной в правом нижнем углу окна. При этом будет вызван диалог, где указывается путь и имя сохраняемого файла, после чего начнется проезд по трассе в автоматическом режиме с одновременной записью в файл. Внимание, движение по трассе будет замедленным, так как программе видеосжатия необходимо время для сжатия каждого кадра. Запись будет производиться до тех пор, пока пользователь не нажмет кнопку «Остановка записи видеоролика» .

Как уже было сказано, запись производится с помощью любого установленного в системе видеокodeка (программы видеосжатия), выбор и настройка параметров которого осуществляется выбором пункта меню «Сервис – Настройка 3D-вида...» и переходом на закладку «Видео».

Здесь будет выведено имя текущего видеокodeка, а также кнопка «Настроить...».

В поле «Количество кадров в секунду» указывается число кадров в секунду. Зная шаг перемещения по трассе, а также указывая различные значения этого поля, можно управлять реальной скоростью проезда, эффект которой почувствуется при просмотре видеоролика.

При нажатии кнопки «Настроить...» покажется непосредственно само окно выбора программы видеосжатия (см. рис. 13), где можно изменить настройки текущего видеокодека или выбрать другой.

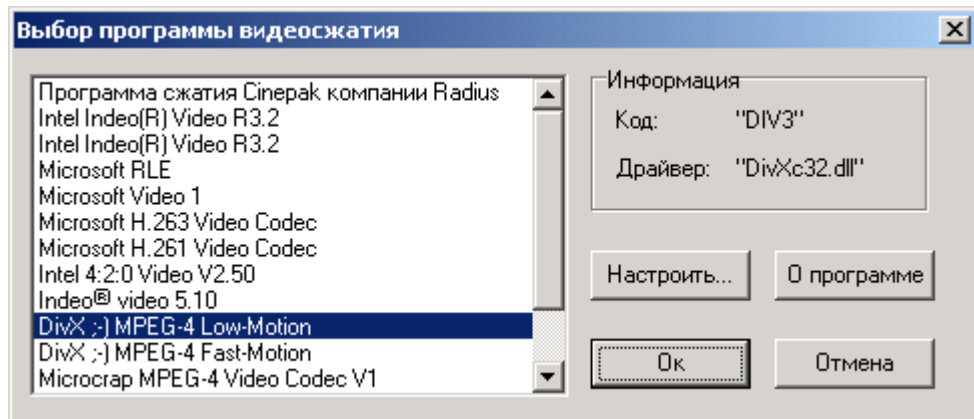


Рис. 13. Диалоговое окно выбора программы видеосжатия

1.3. Окно настроек 3D-вида

Вызов окна настроек параметров 3D-вида осуществляется при помощи выбора пункта меню «Сервис – Настройка 3D-вида...». Внешний вид окна можно увидеть на рис. 14.

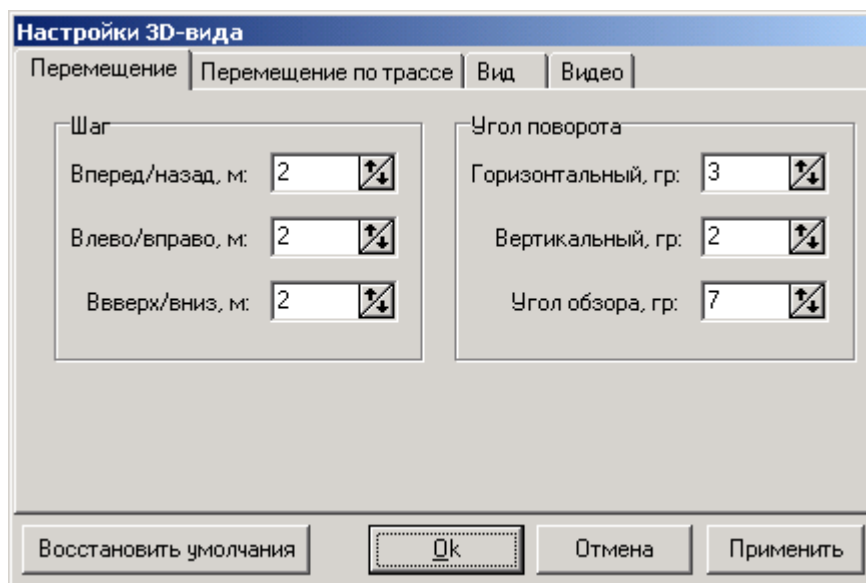


Рис. 14. Окно настроек 3D-вида

Вкладка «Перемещение»

Здесь пользователю предоставляется возможность установки параметров перемещения: шаг и угол поворота.

Описание полей:

«Вперед / назад»	- шаг перемещения в горизонтальном направлении;
«Влево / вправо»	- шаг перемещения в горизонтальном направлении;
«Вверх / вниз»	- шаг перемещения в вертикальном направлении;
«Горизонтальный»	- угол поворота в горизонтальной плоскости, гр;
«Вертикальный»	- угол наклона в вертикальной плоскости, гр;
«Угол обзора»	- угол обзора относительно линии горизонта.

Вкладка «Перемещение по трассе»

Настройка параметров перемещения по трассе.

Описание полей:

«Шаг перемещения»	- шаг перемещения, м;
«Удаленность от осевой линии»	- удаленность от осевой линии, м. Положительное значение говорит о том, что движение будет осуществляться по правой стороне дороги, отрицательное – по левой;
«Высота над дорогой»	- высота точки взора смотрящего над дорогой, м;
«Удаленность точки зрения»	- удаленность точки зрения от точки взора смотрящего в горизонтальной плоскости, м.

Изменением значений полей «Высота над дорогой» и «Удаленность точки зрения» пользователь фактически изменяет угол обзора между линией горизонта и дорожным полотном.

Вкладка «Вид»


Управляет параметрами отображения:

«Земля»	- отображать / не отображать землю;
«Небо»	- отображать / не отображать небо;
«Сетка»	- отображать / не отображать сетку;
«Цвет земли»	- цвет земли;
«Цвет сетки»	- цвет сетки.

Вкладка «Видео»

Выбор и настройка параметров видеосжатия. Назначение и описание полей было описано выше в пункте «Запись видеоролика».

2. ДОРОЖНЫЕ ЗНАКИ

Для установки дорожных знаков необходимо переключиться в соответствующий режим. Осуществляется это посредством нажатия кнопки «Дорожные знаки»  расположенной на панели инструментов. Выход из режима происходит при повторном нажатии кнопки. Далее необходимо установить курсор мыши в нужные координаты и нажать левую кнопку мыши. При этом будет выведено диалоговое окно редактирования знаков (рис. 15). Редактирование ранее установленного знака происходит при наведении курсора мыши на плане на этот знак и нажатии левой кнопки мыши. При этом в строке состояния будет выведена надпись «Дорожный знак». Для удаления знака необходимо произвести ранее описанные действия, а затем в окне редактирования дорожных знаков нажать кнопку «Удалить».

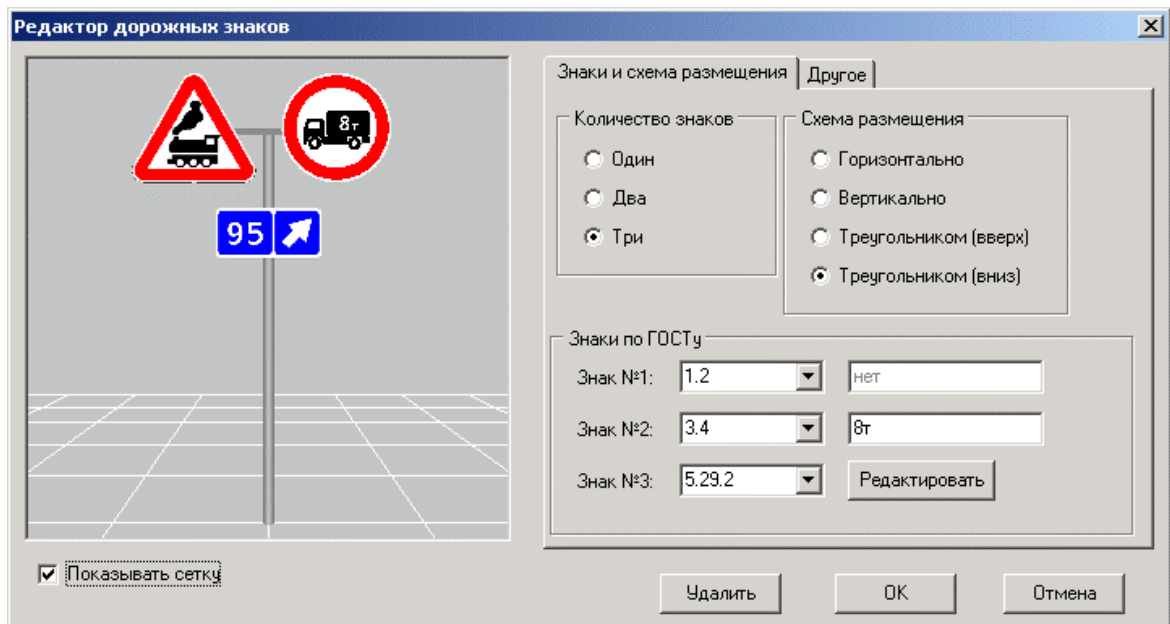


Рис. 15. Окно редактора дорожных знаков

2.1. Окно редактора дорожных знаков

Под редактированием понимается выбор необходимых знаков и схемы их размещения на одной стойке, выбор типоразмера, выбор диаметра стойки, а также указание координат установки (с привязкой к трассам или без привязки) и ориентации. Параметры настроек вкладки «Знаки и схема размещения» интуитивно понятны из рис. 15. Стоит отметить, что в зависимости от выбора количества знаков доступ к некоторым полям будет автоматически разрешаться или запрещаться.

При выборе знака, у которого можно изменить надпись, будет открыт доступ к полю для ввода текста. Текст должен вводиться в одну строку, а программа в зависимости от типа знака сама произведет необходимые действия по форматированию и переносу.

При выборе знаков индивидуального проектирования 5.20.1, 5.20.2, 5.21.1 – 5.27, 5.29.1, 5.29.2 вместо поля для ввода текста появится кнопка «Редактировать», при нажатии на которую будет выведено окно для редактирования знаков индивидуального проектирования.

На вкладке «Другое» (рис. 16) пользователю предоставляется возможность настройки дополнительных параметров: высоты и диаметра стойки, типоразмера знака (категорию), а также координаты установки знака и его ориентировочный угол.

Рис.16. Вкладка «Другое»

Отображение знаков с редактируемыми параметрами и знаков индивидуального проектирования происходит со значениями, принятыми по умолчанию, за исключением тех случаев, когда пользователь произвел действия по их редактированию.

2.2. Окно редактора знаков индивидуального проектирования

Как видно из названия, два знака с одним и тем же номером по ГОСТ могут внешне различаться ввиду своих индивидуальных особенностей. Для таких знаков существуют общие правила компоновки элементов на щите, которые полностью описываются ГОСТ 10807-78. Редактор знаков индивидуального проектирования (рис. 17) позволяет производить компоновку элементов на щите. В зависимости от редактируемого знака

пользователю предоставляется возможность добавления, удаления и изменения параметров тех или иных элементов, а компоновка их происходит в автоматическом режиме.

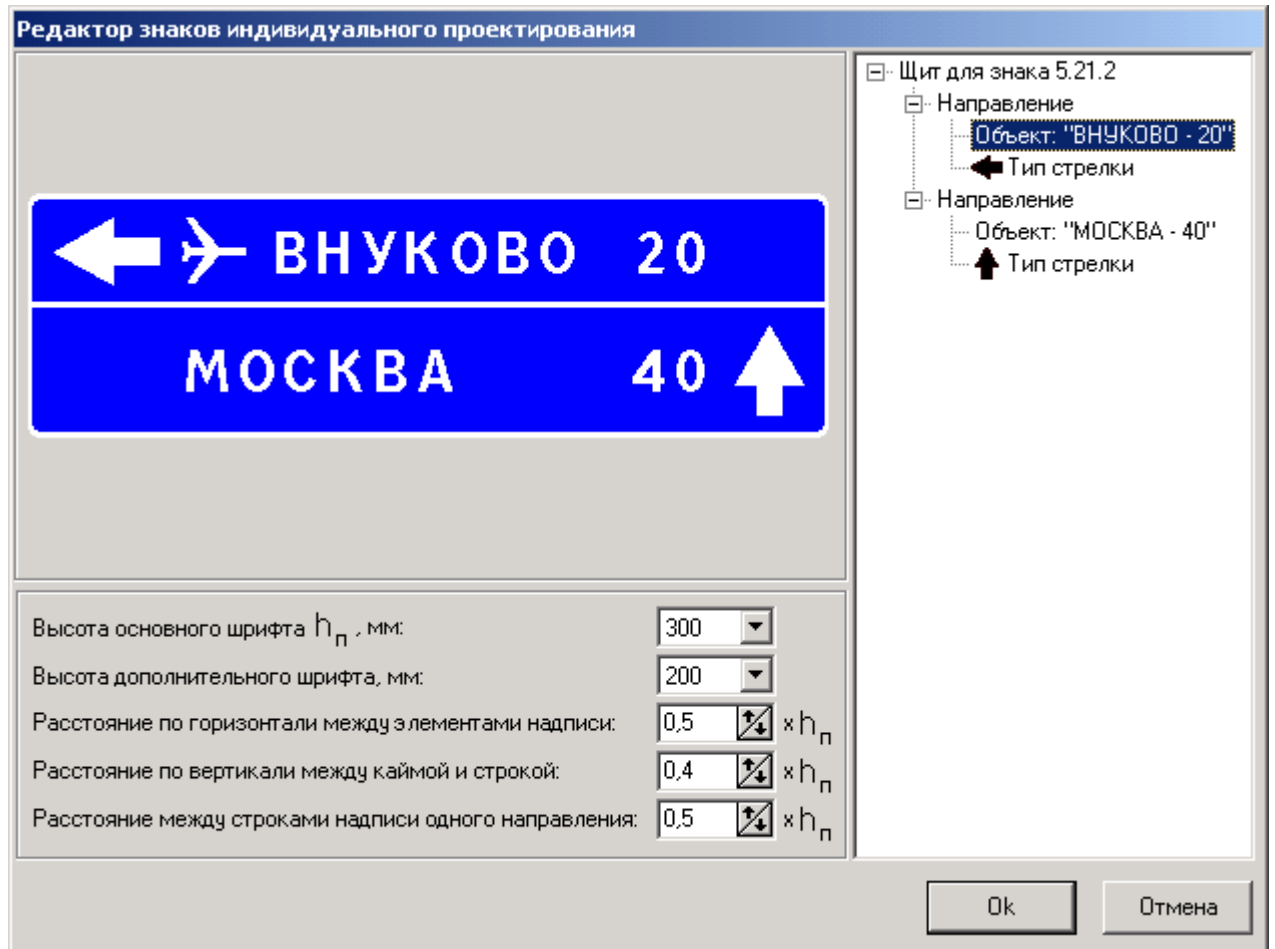


Рис. 17. Окно редактора знаков индивидуального проектирования

Окно редактора состоит из трех частей:

- Изображение редактируемого знака;
- Поле для ввода параметров (снизу);
- Дерево объектов (справа).

Изображение редактируемого знака. Здесь пользователь может визуально отслеживать все изменения, происходящие при действиях с объектами.

Поле для ввода параметров. Здесь задаются значения параметров, принимаемых во внимание при автоматической компоновке объектов. Все расчеты размера шрифта для надписей, высоты пиктограммы, ширины каймы знака, размера стрелки, а также значений параметров производятся в зависимости от высоты основного шрифта h_n . Надписи на знаке могут выполняться двумя шрифтами: основным и дополнительным.

Дерево объектов. Это дерево – есть иерархическая структура объектов. Здесь хорошо просматривается принадлежность одних объектов другим.

В зависимости от типа знака над объектами можно производить те или иные действия. Перечень этих действий будет приведен ниже.

Следует заметить, что селекцию объекта можно производить как при помощи дерева объектов, так и при помощи изображения редактируемого знака, путем указания курсором мышки необходимого объекта. В любом случае выделенный объект обводится абрисом.

Настройка свойств объекта производится либо двойным щелчком мыши на объекте, либо вызовом контекстного меню как в области дерева объектов, так и в области изображения знака.

Далее приводится список объектов и их свойств, а также связанных с ними контекстных меню (действий) для каждого знака индивидуального проектирования.

Знак 5.20.2 «Предварительный указатель направлений»

Возможные объекты:

- Щит;
- Объект;
- Стрелка.

Свойства и действия объекта «*Щит*»:

- Цвет щита;
- Добавить объект.

Свойства и действия объекта «*Объект*»:

- Название объекта;
- Шрифт (основной или дополнительный);
- Цвет шрифта;
- Удалить.

Свойства и действия объекта «*Стрелка*»:

- Тип стрелки;
- Положение (слева или справа).

Знак 5.21.1 «Указатель направления»

Возможные объекты:

- Щит;
- Название объекта.

Свойства и действия объекта «*Щит*»:

- Цвет щита;

- Направление щита (налево или направо).

Свойства и действия объекта «*Объект*»:

- Название объекта;
- Расстояние до объекта;
- Шрифт (основной или дополнительный);
- Цвет шрифта;
- Пиктограммка (нет, аэропорт, автомагистраль);
- Размер пиктограммки.

Знак 5.21.2 «Указатель направлений»

Возможные объекты:

- Щит;
- Направление;
- Объект;
- Стрелка.

Свойства и действия объекта «*Щит*»:

- Цвет щита;
- Добавить направление.

Свойства и действия объекта «*Направление*»:

- Цвет фона;
- Добавить объект;
- Удалить.

Свойства и действия объекта «*Объект*»:

- Название объекта;
- Расстояние до объекта;
- Шрифт (основной или дополнительный);
- Цвет шрифта;
- Пиктограммка (нет, аэропорт, автомагистраль);
- Удалить.

Свойства и действия объекта «*Стрелка*»:

- Тип стрелки;
- Положение (слева или справа).

Знаки 5.22, 5.23, 5.24, 5.25, 5.26 «Начало/конец населенного пункта»

Возможные объекты:

- Щит;

- Объект;
- Красная полоса (только у знаков 5.23 и 5.25);

Свойства и действия объекта «Щит»:

- Цвет щита;
- Объект.

Свойства и действия объекта «Объект»:

- Название объекта;
- Шрифт (основной или дополнительный);
- Цвет шрифта.

Знак 5.27 «Указатель расстояний»

Возможные объекты:

- Щит;
- Объект.

Свойства и действия объекта «Щит»:

- Цвет щита;
- Добавить объект.

Свойства и действия объекта «Объект»:

- Название объекта;
- Расстояние до объекта;
- Шрифт (основной или дополнительный);
- Цвет шрифта;
- Удалить.

Знак 5.28 «Километровый знак»

Возможные объекты:

- Щит;
- Километраж.

Свойства и действия объекта «Щит»:

- Цвет щита.

Свойства и действия объекта «Километраж»:

- Значение.

Знаки 5.29.1, 5.29.2 «Номер маршрута»

Возможные объекты:

- Щит;
- Номер маршрута;

- Стрелка.

Свойства и действия объекта «Щит»:

- Цвет щита.

Свойства и действия объекта «Номер маршрута»:

- Номер маршрута.

Свойства и действия объекта «Стрелка»:

- Тип стрелки;
- Положение (слева или справа).

После всех действий по выбору и компоновке знаков и закрытии окна редактора спроектированный дорожный знак появится в окне 3D-вида, как показано на рис. 18.

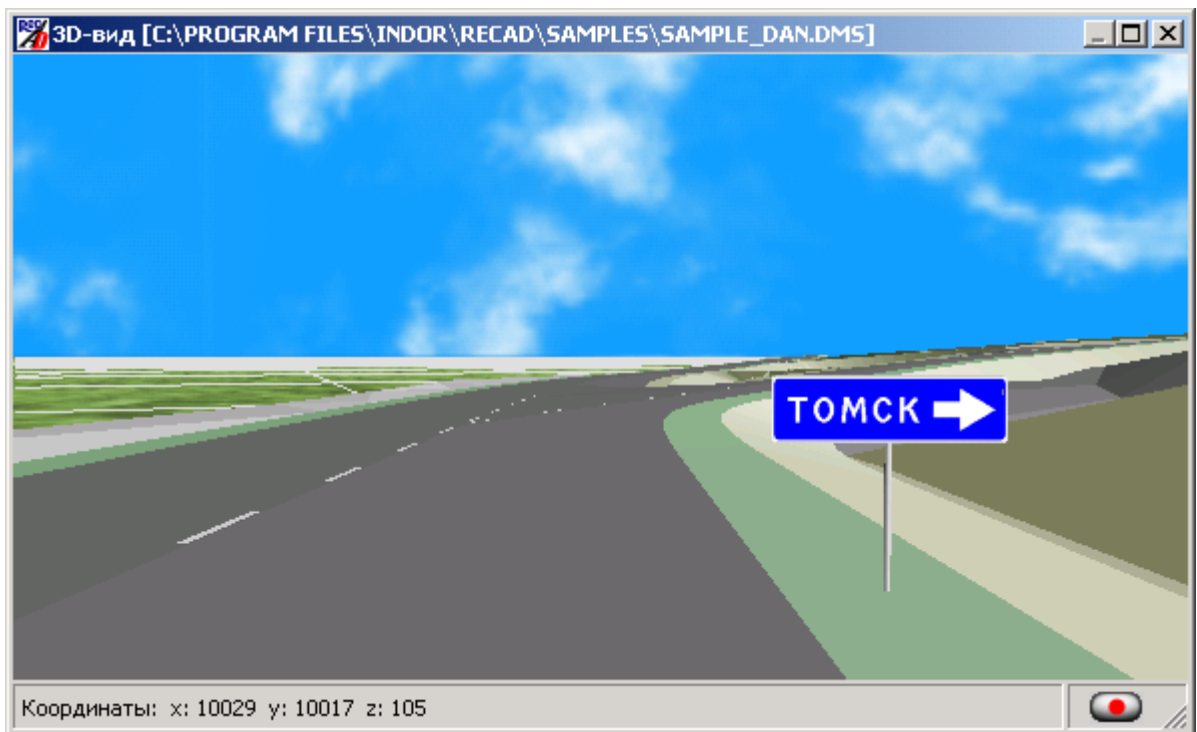


Рис. 18. Установленный дорожный знак

ПРИЛОЖЕНИЕ 2. РУКОВОДСТВО ПРОГРАММИСТА

Вся работа состоит из двух основных частей: модуля визуализации трехмерного вида и модуля редактора дорожных знаков.

1. МОДУЛЬ ВИЗУАЛИЗАЦИИ ТРЕХМЕРНОГО ВИДА

Этот модуль отвечает за отображение существующих поверхностей и различных объектов в трехмерном виде.

Модуль состоит из следующих блоков:

- Основной модуль;
- Модуль настроек;
- Модуль выбора и настройки видеокodeка.

1.1. View3DMainUnit.pas - основной модуль

Данный модуль отвечает непосредственно за саму визуализацию, а также является связующим звеном для всех остальных модулей. Входит в состав проекта и компилируется вместе с ним.

Классы и описание из методов:

Класс «TView3DForm»

Основной класс, он же – окно 3D-вида.

Переменные класса:

<i>DX: DirectX8</i>	- экземпляр интерфейса IDirectX8
<i>D3D: Direct3D8</i>	- экземпляр интерфейса Direct3D8
<i>D3DX: ID3DX8</i>	- экземпляр интерфейса ID3DX8
<i>Dev: Direct3DDevice8</i>	- устройство вывода изображения, ассоциированное с Direct3D
<i>DeviceCreated: Boolean</i>	- флаг успешного создания устройства вывода Direct3D
<i>Layers: array of TLayer</i>	- массив объектов-слоёв
<i>Camera: TCamera</i>	- камера
<i>CarCamera: TCarCamera</i>	- камера для движения по трассе.
<i>Video: TVideo</i>	- экземпляр класса TVideo, отвечающий за запись видеоролика
<i>GroundEnabled: Boolean</i>	- отображать землю
<i>SkyEnabled: Boolean</i>	- отображать небо
<i>GridEnabled: Boolean</i>	- отображать сетку

Методы класса:

function CreateDirect3DDevice(hWnd: THandle): Boolean

Создать устройство Direct3D, ассоциированное с окном hWnd. В случае успеха возвращает TRUE, иначе FALSE.

function ResetDirect3DDevice(hWnd: THandle): Boolean

Сбросить и заново проинициализировать устройство Direct3D, ассоциированное с окном hWnd. Вызывается при смене видеорежима.

procedure SetupMatrices

Установить матрицы перспективы, положения и вида в соответствии с текущим положением.

procedure UpdatePosition

Сменить позицию в соответствии с текущим положением и вывести текущие координаты в строке статуса.

function ReloadLayers: Boolean

Перезагрузить все слои. Вызывается при любых изменениях на плане проекта.

procedure LoadOptionsFromDoc(DOC: TRecadDocument)

Загрузить все настройки документа DOC.

procedure SaveOptionsToDoc(DOC: TRecadDocument)

Сохранить настройки в документе DOC.

procedure LoadVideoOptions

Загрузить настройки видеокодека.

procedure SaveVideoOptions

Сохранить настройки видеокодека.

procedure DrawSky

Отрисовать небо.

procedure DrawScene;

Основная процедура, в которой происходит отрисовка 3D-вида и всех трехмерных объектов.

Класс «TSimpleWindowView3D»

Небольшой класс, экземпляр которого всегда находится в памяти, независимо от того, активирован 3D-вид или нет. Отслеживает все изменения в документе и оповещает об этом окно 3D-вида.

Переменные класса:

CurrentDOC: TrecadDocument - текущий документ

Методы класса:

procedure WndProc(var M:TMessage)

Принимает сообщения, приходящие от основного окна и в зависимости от них вызывает те или иные функции окна 3D-вида.

Класс «TLayer»

Класс, представляющий из себя один слой (цифровую модель местности). Используется как контейнер мешей.

Переменные класса:

<i>Device: Direct3DDevice8</i>	- устройство вывода Direct3D
<i>MeshCount: integer</i>	- количество мешей
<i>Meshes: array of D3DXMesh</i>	- массив мешей, представляющих поверхность
<i>Razmetka: array of D3DXMesh</i>	- массив мешей, представляющих разметку дороги

Методы класса:

constructor Create(Dev: Direct3DDevice8)

Создание экземпляра класса и ассоциация его с устройством вывода Direct3D.

procedure Draw

Отрисовка всех мешей. Заполнение мешей производится в случае перезагрузки слоя в функции ReloadLayers основного окна 3D-вида.

Класс «TCamera»

Класс предназначенный для управления камерой. Под камерой понимается инструмент, реализующий функции перемещения.

Переменные класса:

<i>EyePos: D3DVector3</i>	- текущие координаты глаза наблюдателя
<i>AtPos: D3DVector3</i>	- текущие координаты точки взгляда
<i>HorizStep: single</i>	- шаг перемещения в горизонтальной плоскости
<i>VertStep: single</i>	- шаг перемещения в вертикальной плоскости
<i>SideStep: single</i>	- шаг перемещения, при движении влево/вправо
<i>HorizAngle: single</i>	- угол поворота в горизонтальной плоскости
<i>VertAngle: single</i>	- угол поворота в вертикальной плоскости
<i>KrenAngle: single</i>	- угол наклона

Методы класса:

procedure SetKrenAngle(Angle: single)

Установить угол наклона.

function GetKrenAngle: single

Возвращает угол наклона.

procedure SetEyePosition(x, y, z: single)

Установить координаты глаза наблюдателя.

function GetEyePosition: D3DVECTOR3

Возвращает координаты глаза наблюдателя.

procedure SetAtPosition(x, y, z: single)

Установить координаты точки взгляда.

function GetAtPosition: D3DVECTOR3

Возвращает координаты точки взгляда.

procedure SetHorizStep(s: single)

Установить горизонтальный шаг перемещения.

function GetHorizStep: single

Возвращает горизонтальный шаг перемещения.

procedure SetVertStep(s: single)

Установить вертикальный шаг перемещения.

function GetVertStep: single

Возвращает вертикальный шаг перемещения.

procedure SetSideStep(s: single)

Установить шаг перемещения влево/вправо.

function GetSideStep: single

Возвращает шаг перемещения влево/вправо.

procedure SetHorizAngle(angle: single)

Установить угол горизонтального поворота.

function GetHorizAngle: single

Возвращает угол горизонтального поворота.

procedure SetVertAngle(angle: single)

Установить угол вертикального наклона.

function GetVertAngle: single

Возвращает угол вертикального наклона.

procedure SetViewAngle(angle: single; Update: boolean)

Установить угол обзора относительно горизонта.

function GetViewAngle: single

Возвращает угол обзора относительно горизонта.

procedure MoveForward(Update: boolean)

Переместиться вперед.

procedure MoveBack(Update: boolean)

Переместиться назад.

procedure MoveLeft(Update: boolean)

Переместиться влево.

procedure MoveRight(Update: boolean)

Переместиться вправо.

procedure MoveUp(Update: boolean)

Переместиться вверх.

procedure MoveDown(Update: boolean)

Переместиться вниз.

procedure RotateLeft(Update: boolean)

Повернуться налево.

procedure RotateRight(Update: boolean)

Повернуться направо.

procedure LookUp(Update: boolean)

Наклониться вверх.

procedure LookDown(Update: boolean)

Наклониться вниз.

function CheckBound(NewPos: D3DVECTOR): boolean

Проверить, не выйдет ли положение камеры NewPos за пределы допустимой зоны перемещения.

Класс «TVideo»

Этот класс создан для записи ролика. При инициализации экземпляр класса связывается с окном 3D-вида, и каждый отрисованный кадр передается программе видеосжатия для компрессии в необходимый видеоформат.

Переменные класса:

<i>Form: TView3DForm</i>	- форма с которой будет браться изображение
<i>Bitmap: TBitmap</i>	- битмап с изображением
<i>FileName: String</i>	- имя AVI файла
<i>AVIFile: IAVIFile</i>	- интерфейс, поддерживающий создание AVI файла и управляющий его потоками
<i>AVIStream: IAVIStream</i>	- интерфейс, поддерживающий работу с видеопотоками
<i>AVICompressedStream: IAVIStream</i>	- интерфейс, поддерживающий работу с

	компрессированными видеопотоками
<i>Options: TAVICompressOptions</i>	- опции программы-видеокомпрессора
<i>nFrame: integer</i>	- текущее количество записанных кадров
<i>Recording: Boolean</i>	- индикация режима записи
<i>Codec: Tcodec</i>	- настройки видеокодека.

Методы класса:

constructor Create(aForm: TView3DForm)

Создание экземпляра класса и ассоциация с окном aForm, с которого будет происходить запись.

function StartRecording(const FName: string): Integer

Начало записи видеоролика в файл FName.

procedure StopRecording

Остановка записи видеоролика.

function AddFrame: Boolean

Добавление очередного кадра. В случае ошибки компрессии возвращает FALSE.

Структура «TCodec»

Структура, которая содержит настройки конкретного видеокодека.

Переменные:

<i>fccHandler: Integer</i>	- уникальный идентификатор видеокодека
<i>Name: String</i>	- название видеокодека
<i>DriverName: String</i>	- имя драйвера программы видеосжатия
<i>Configure: Boolean</i>	- есть ли окно конфигурации у текущего видеокодека
<i>About: Boolean</i>	- есть ли окно информации у текущего видеокодека
<i>FPS: Integer</i>	- количество кадров в секунду
<i>OptionsSize: Integer</i>	- размер блока параметров видеокодека
<i>Options: array[0..1023] of byte</i>	- параметры видеокодека

1.2. View3DSetupUnit.pas – модуль настроек

Этот модуль представляет собой окно настроек.

Переменные:

<i>OldHorizStep: single</i>	- старое значение шага перемещения в горизонтальной плоскости
<i>OldVertStep: single</i>	- старое значение шага перемещения в вертикальной плоскости
<i>OldSideStep: single</i>	- старое значение шага перемещения, при движении влево/вправо

<i>OldHorizAngle: single</i>	- старое значение угла поворота в горизонтальной плоскости
<i>OldVertAngle: single</i>	- старое значение угла поворота в вертикальной плоскости
<i>OldCarStep: single</i>	- старое значение шага перемещения
<i>OldOsiOffset: single</i>	- старое значение смещения от оси
<i>OldEyeHeight: single</i>	- старое значение высоты наблюдателя над дорогой
<i>OldLookDistance: single</i>	- старое значение удаленности точки взгляда
<i>OldGroundColor</i>	- старое значение цвета земли
<i>OldGridColor: integer</i>	- старое значение цвета сетки
<i>OldCodec: TCodec</i>	- старое значение параметров видеокодека
<i>Codec: TCodec</i>	- параметры видеокодека

1.3. View3DCodecUnit.pas – модуль настроек программы видеосжатия

Данный модуль представляет собой окно выбора и настройки программы видеосжатия.

2. МОДУЛЬ РЕДАКТОРА ДОРОЖНЫХ ЗНАКОВ

Редактор дорожных знаков оформлен как модуль расширения (плагин) и в конечном варианте поставляется как отдельный DLL файл. Для его работы необходимо наличие библиотеки дорожных знаков Indor3DZnaks.dll.

С точки зрения программиста редактор состоит из следующих составных частей:

- Сам модуль расширения;
- Модуль редактора дорожных знаков;
- Модуль редактора знаков индивидуального проектирования.

2.1. Znaks3D.pas – модуль расширения дорожных знаков (плагин)

Данный модуль является модулем расширения (плагином) для использования дорожных знаков. Он содержит интерфейс взаимодействия с основной программой ReCAD и включает в себя редактор основных дорожных знаков и редактора знаков индивидуального проектирования.

В модуле описаны два класса: TPlug и TRoadZnaks и одна процедура инициализации. Оба класса реализуют поддержку нескольких интерфейсов.

Для более детального понимания далее приводится список интерфейсов и их краткое описание:

IReCADDataPlugin

Интерфейс, который отвечает за создание экземпляра объекта-плагина. По сути это и есть сам плагин.

ISimpleResponder

Интерфейс, отвечающий за обработку нажатий кнопок и выполнения вызванных пунктов меню.

IReCADObject

Все объекты с данными системы ReCAD представлены в виде дерева и каждый такой объект должен реализовывать этот интерфейс, при помощи которого затем он может обратиться к любому объекту по имени.

IRecadCommand

Командой назовём режим работы системы ReCAD (обработка событий мыши, клавиатуры и пр.), при котором устанавливается какой-то режим или отменяется предыдущий. Объект, реализующий «команду» должен реализовать этот интерфейс.

IReCADCommandMouse

Если при работе команды (при включенном режиме) вы хотите получать события мыши, аналогичные «MouseMove», «MouseDown», «MouseUp», вы должны реализовать этот интерфейс.

IReCADCommandDraw

Если вы хотите производить отрисовку на план, то необходимо реализовать этот интерфейс.

ISuperStreamed

При необходимости сохранения данных вместе с файлом проекта необходимо реализовать этот интерфейс.

IObject3D

Если вы хотите производить отрисовку каких-то трехмерных объектов в окне 3D-вида, то необходимо реализовать этот интерфейс.

Инициализация плагина производится вызовом процедуры *ReCADPluginInit* извне, в которой происходит бавление кнопки активизации на панель инструментов программы ReCAD.

Описание классов и их методов:

Класс «TPlug»

Данный класс реализует поддержку двух интерфейсов: *IReCADDataPlugin* и *ISimpleResponder*.

Методы класса:

function CreateInstance(const Context: IUnknown; const DOC: IDocument): IRecadObject

Реализует метод интерфейса *IReCADDataPlugin*. Создает экземпляр объекта-плагина.

procedure OnButtonClick(IsDown: WordBool; UserData: Integer)

Реализует метод интерфейса *ISimpleResponder*. Генерирует команду перехода в режим установки и редактирования дорожных знаков.

procedure RefreshParams(UserData: Integer)

Реализует метод интерфейса *ISimpleResponder*. Ничего не делает.

Класс «TRoadZnaks»

Этот класс содержит в себе информацию о знаках и является связующим звеном между системой ReCAD, окном 3D-вида и редактором знаков.

Переменные класса:

Layer: ILayer - текущий слой

DOC: IDocument - текущий документ

ZnakList: TZnakList - контейнер объектов-знаков

ZnakInfo: TZnakInfo - структура описания знаков

Методы класса:

procedure Add(const aZnakParam: TZnakParam)

Добавить знак, который описывается структурой aZnakParam.

procedure Delete(index: integer)

Удалить знак.

function GetZnakParam(Index: integer): PZnakParam

Получить параметры у знака под номером index.

function GetCount: integer

Возвращает количество знаков в контейнере.

function SearchZnak(xx, yy, r: extended): integer

Попытка найти знак, установленный в заданные координаты. В случае успеха возвращает его номер в контейнере, иначе возвращает -1.

procedure EnterCommand

Метод интерфейса IReCADCommand. Переход в режим установки дорожных знаков.

procedure ExitCommand

Метод интерфейса IReCADCommand. Выход из режима установки дорожных знаков.

function ProcessMouse(Reason: Integer; xx: Double; yy: Double; Shift: Integer;

Reserved1: Integer; Reserved2: Integer): integer

Метод интерфейса IReCADCommandMouse. Пытается найти знак, установленный в место с координатами x, y. Если такой знак найден, то вызывается редактор знаков для его дальнейшего редактирования, иначе редактор вызывается для создания нового знака.

procedure DrawToD3D(const aDirect3DDevice8: IUnknown)

Метод интерфейса IObject3D. Рисует все знаки в окне 3D-вида.

procedure FreeD3DResources

Метод интерфейса IObject3D. Освобождает все графические ресурсы, занятые дорожными знаками.

procedure SaveTo(const ISS: IUnknown)

Метод интерфейса ISuperStreamed. Сохранение всех знаков.

procedure LoadFrom(const ISS: IUnknown)

Метод интерфейса ISuperStreamed. Загрузка всех знаков.

2.2. View3DZnakUnit.pas – редактор дорожных знаков

Данный модуль предназначен для редактирования дорожных знаков. Цель работы данного модуля является формирование на конечном этапе структуры типа TZnakParam, которая полностью описывает дорожный знак.

Структура «TZnakParam»:

<i>Scheme: integer</i>	- схема размещения знаков на одной стойке
<i>Category: integer</i>	- категория знака
<i>Height: single</i>	- высота стойки
<i>Diameter: single</i>	- диаметр стойки
<i>x, y, z: single</i>	- координаты установки знака
<i>angle: single</i>	- угол ориентирования знака
<i>Znak1: string[15]</i>	- название 1-го знака
<i>Znak2: string[15]</i>	- название 2-го знака
<i>Znak3: string[15]</i>	- название 3-го знака
<i>Text1: string[31]</i>	- текст 1-го знака
<i>Text2: string[31]</i>	- текст 2-го знака
<i>Text3: string[31]</i>	- текст 3-го знака
<i>Structure1: TIndividualZnakProperty</i>	- структура 1-го индивидуального знака
<i>Structure2: TIndividualZnakProperty</i>	- структура 2-го индивидуального знака
<i>Structure3: TIndividualZnakProperty</i>	- структура 3-го индивидуального знака

В этом модуле созданы два класса: TZnakInfo и TZnak.

Класс «TZnakInfo»

Этот класс содержит всю информацию о дорожных знаках: размеры, параметры по умолчанию и другую информацию, необходимую при их размещении на стойке.

Переменные класса:

<i>Name: TStringList</i>	- знаки по имени
<i>Gost: TStringList</i>	- знаки по номеру ГОСТ
<i>Typ: array of integer</i>	- знаки по их типу
<i>Description: TStringList</i>	- описания знаков
<i>ZnakText: TZnakText</i>	- надписи на знаках, принятые по умолчанию

Методы класса:

constructor Create(DLLName: string)

Создать экземпляр класса, загрузить информацию из библиотеки дорожных знаков DLLName и заполнить параметры.

function GetParam(Тип, Category: integer; var H, B, Radius: single): boolean

Возвращает ширину, высоту и радиус (для круглых знаков) знака типа Тип и категории Category.

function GetSize(Тип, Category: integer; var H, B: single): boolean

Возвращает ширину и высоту знака типа Тип и категории Category.

function GetTypeByName(const AName: string): integer

Возвращает тип знака по его имени.

function GetZnakTextByName(const AName: string): TZnakTextRecord

Возвращает текст по умолчанию, наносимый на знак, по его имени.

Класс «TZnak»

Экземпляр этого класса представляет собой знак, который по заданной структуре может «собрать» объемный знак и отобразить его в 3D-виде.

Переменные класса:

<i>ZnakInfo: TZnakInfo</i>	- ссылка на экземпляр класса TZnakInfo, содержащего необходимую информацию
<i>ZnakCount: integer</i>	- количество знаков, размещаемых на стойке (от 1 до 3)
<i>StoykaHeight: single</i>	- высота стойки
<i>StoykaDiameter: single</i>	- диаметр стойки
<i>StoykaMesh: D3DXMesh</i>	- меш стойки
<i>CrossbarMesh: D3DXMesh</i>	- меш для всех соединительных планок
<i>Category: integer</i>	- категория знака
<i>Scheme: integer</i>	- схема размещения знаков
<i>Names: TStringList</i>	- имена знаков
<i>ZnakMesh: array [0..2] of D3DXMesh</i>	- меши для 1-го, 2-го и 3-го знаков соответственно
<i>Textures: array [0..2] of Direct3DTexture8</i>	- текстуры с изображениями для 1-го, 2-го и 3-го знаков соответственно

Методы класса:

constructor Create(aDevice: Direct3DDevice8; aZnakInfo: TZnakInfo)

Создать экземпляр класса, увязать его с устройством вывода изображения Direct3D (aDevice) и экземпляром класса TZnakInfo (aZnakInfo), содержащего информацию о знаках.

procedure Clear

Освободить все ресурсы, занятые знаком.

procedure Draw

Отрисовать знак в окне 3D-вида.

procedure Build(zp: TZnakParam)

«Собрать» знак в памяти, т.е. создать все меши и текстуры, в соответствии со структурой описания знака zp.

*procedure CreateTextureFromWMFResource(SignName: string; Text: string;
out Texture: Direct3DTexture8)*

Создать текстуру с изображением для знака SignName.

function CreateCylinder(Radius, Height: single; Color: integer): D3DXMesh

Создать меш стойки радиуса Radius и высоты Height.

procedure CreateCircle(Radius: single; Center: D3DVector; out Mesh: D3DXMesh)

Создать меш для знака круглой формы радиуса Radius.

procedure CreateTriangle(SideLen: single; Center: D3DVector; out Mesh: D3DXMesh)

Создать меш для знака треугольной формы со стороной SideLen.

procedure CreateTriangle2(SideLen: single; Center: D3DVector; out Mesh: D3DXMesh)

Создать меш для перевернутого знака треугольной формы со стороной SideLen.

procedure CreateRectangle(Width, Height, R: single; Center: D3DVector; out Mesh: D3DXMesh)

Создать меш для знака прямоугольной формы размером Width x Height с закругленными углами с радиусом закругления R.

procedure CreateRectangle90(Width, Height: single; Center: D3DVector; out Mesh: D3DXMesh)

Создать меш для знака прямоугольной формы размером Width x Height.

procedure CreateRomb(Width, R: single; Center: D3DVector; out Mesh: D3DXMesh)

Создать меш для знака в форме ромба со скругленными углами с ребром Width.

procedure CreateStop(Width: single; Center: D3DVector; out Mesh: D3DXMesh)

Создать меш для знака «STOP».

procedure CreateRailway(Center: D3DVector; out Mesh: D3DXMesh)

Создать меш для знака «Однопутная железная дорога».

procedure CreateRailway2(Center: D3DVector; out Mesh: D3DXMesh)

Создать меш для знака «Многопутная железная дорога».

*procedure CreateZnakMesh(const ZnakName: string; Category: integer; Center: D3DVector;
out Mesh: D3DXMesh)*

Создать меш для знака ZnakName категории Category с центром в точке Center.

2.3. IndividualSignUnit.pas – редактор знаков индивидуального проектирования

Редактор индивидуальных знаков предназначен для редактирования знаков индивидуального проектирования. Под редактированием в данном случае понимается размещение различных элементов на щите знака. Данный редактор как дополнение входит в состав редактора дорожных знаков. Целью редактора является формирование на выходе такой структуры, по которой затем возможно будет произвести компоновку всех его элементов в автоматическом режиме, и в конечном результате получить его изображение для использования в качестве текстуры.

Структура описания знака представляет собой дерево, каждый элемент которого является элементом знака и описывается некоторыми свойственными для него параметрами. Все возможные параметры описываются в структуре TShieldObjectData.

Структура TShieldObjectData:

<i>Typ: integer</i>	- тип объекта
<i>ShieldType: integer</i>	- тип щита
<i>Text: string</i>	- текст
<i>Number: integer</i>	- числовое значение
<i>MainFont: boolean</i>	- использовать основной шрифт или дополнительный
<i>FontColor: TColor</i>	- цвет шрифта
<i>BkColor: TColor</i>	- цвет фона
<i>Arrow: integer</i>	- тип указательной стрелки
<i>ArrowPos: integer</i>	- положение стрелки на щите (слева или справа)
<i>Pic: integer</i>	- пиктограммка
<i>PicSize: double</i>	- размер пиктограммки

В редакторе описаны два основных класса: TIndividualZnak и TShieldObject.

Класс «TShieldObject»

Экземпляр этого класса представляет собой один элемент щита.

Переменные класса:

<i>OwnerZnak: TIndividualZnak</i>	- знак родитель элемента
<i>Typ: integer</i>	- тип объекта
<i>ShieldType: integer</i>	- тип щита
<i>Text: string</i>	- текст
<i>MainFont: boolean</i>	- использовать основной шрифт или дополнительный

<i>FontColor: TColor</i>	- цвет шрифта
<i>BkColor: TColor</i>	- цвет фона
<i>Number: integer</i>	- числовое значение
<i>Arrow: integer</i>	- тип указательной стрелки
<i>ArrowPos: integer</i>	- положение стрелки на щите (слева или справа)
<i>Pic: integer</i>	- пиктограммка
<i>PicSize: double</i>	- размер пиктограммки
<i>ScrX, ScrY: double</i>	- координаты объекта
<i>Width, Height: double</i>	- размер объекта
<i>ObjectList: TShieldObjectList</i>	- список объектов-потомков

Методы класса:

function AddObject(const ObjectData: TShieldObjectData): TShieldObject

Добавить объект, описываемый структурой ObjectData.

procedure Delete

Удалить объект.

procedure GetSize(PS: TPaintScroll; out W, H: double)

Вернуть ширину и высоту объекта.

procedure Draw(PS: TPaintScroll; px, py: double)

Отрисовать объект в заданных координатах.

function GetPropertyString: string

Вернуть строку-структуру описания объекта.

Класс «TIndividualZnak»

Экземпляр этого класса представляет собой знак индивидуального проектирования.

Переменные класса:

<i>ZnakType: integer</i>	- тип знака
<i>FontSize: double</i>	- размер основного шрифта
<i>FontSize2: double</i>	- размер дополнительного шрифта
<i>HorizSpace: double</i>	- горизонтальное расстояние между элементами
<i>VertSpace: double</i>	- вертикальное расстояние между элементами и каймой
<i>LineSpace: double</i>	- расстояние между строками надписи
<i>Shield: TShieldObject</i>	- объект «щит», который содержит в себе все элементы знака
<i>SelectedObject: TShieldObject</i>	- текущий выделенный элемент знака

Методы класса:

procedure DrawShield(PS: TPaintScroll)

Отрисовать весь знак.

procedure CalculateSize(PS: TPaintScroll; out W, H: double)

Вычислить и вернуть размеры знака.

function GetSignStructure: string

Вернуть структуру описания знака.

procedure FillFromStructure(Struct: string)

Создать и разместить элементы на знаке в соответствии с заданной структурой описания знака.

ПРИЛОЖЕНИЕ 3. АКТ ВНЕДРЕНИЯ