



Глава II

Цели

Цели этого проекта аналогичны всем целям первого года обучения: строгость, использование C, использование

базовых алгоритмов, информационные исследования и т.д.

В качестве проекта графического дизайна sub3D позволит вам улучшить свои навыки в этих области: окна, цвета, события, формы заливки и т.д.

В заключение можно сказать, что sub3D - это замечательная игровая площадка для изучения игровых практических приложений.

катионы математики без необходимости разбираться в специфике.

С помощью многочисленных документов, доступных в Интернете, вы будете использовать математика как инструмент для создания элегантных и эффективных алгоритмов.

Мы рекомендуем вам протестировать оригинальную игру перед началом этого проекта:

<http://users.atw.hu/wolf3d/>

3

Глава III

Общие инструкции

•

Ваш проект должен быть написан на C.

•

Ваш проект должен быть написан в соответствии с Нормой. Если у вас есть бонусные файлы / функции, они включены в проверку нормы, и вы получите 0, если внутри есть ошибка нормы.

•

Ваши функции не должны завершаться неожиданно (ошибка сегментации, ошибка шины, двойное освобождение и т.д.), Кроме неопределенного поведения. Если это произойдет, ваш проект будет считаться нефункциональным и получит 0 баллов во время оценки.

•

Все пространство памяти, выделенное для кучи, должно быть должным образом освобождено при необходимости. Утечки недопустимы.

•

Если этого требует тема, вы должны отправить Makefile, который скомпилирует ваши исходные файлы в требуемый результат с флагами -Wall , -Wextra и -Werror , используйте cc , и ваш Makefile не должен повторно связываться.

•

Ваш Makefile должен содержать, по крайней мере, правила \$(NAME), all, clean, fclean и re

•

•

Чтобы включить бонусы в свой проект, вы должны включить бонус к правилу в свой Makefile, который добавит все различные заголовки, библиотеки или функции, запрещенные в основной части проекта. Бонусы должны быть в другом файле _bonus.{c/h}.

Обязательная и бонусная части оцениваются отдельно.

•

Если ваш проект позволяет вам использовать ваш libft, вы должны скопировать его исходные тексты и

связанный с ним Makefile в папку libft с соответствующим Makefile.

Makefile вашего проекта

необходимо скомпилировать библиотеку с помощью ее Makefile, а затем скомпилировать проект.

•

Мы рекомендуем вам создавать тестовые программы для вашего проекта, даже если эту работу **не нужно будет отправлять и она не будет оцениваться**

. Это даст вам возможность

легко проверить свою работу и работу ваших коллег. Вы найдете эти тесты особенно полезными во время вашей защиты. Действительно, во время защиты вы можете свободно использовать свои тесты

и / или тесты коллег, которых вы оцениваете.

•

Отправьте свою работу в назначенный вам репозиторий git.

Оцениваться будет только работа в репозитории git. Если для оценки вашей работы назначена Deert Thought, это будет сделано

после ваших экспертных оценок. Если во время оценки Deert Thought в каком-либо разделе вашей работы возникнет ошибка

, оценка будет остановлена.

4

Глава IV

Обязательная часть - cub3D

Название программы

cub3D

Сдача файлов

Все ваши файлы

Makefile Создать файл

все, чисто, чисто, повторно, бонус

Аргументы

карта в формате *.cub

Внешние функции.

•

открывать, закрывать, читать, записывать,

printf, malloc, свободно

, ошибка, ошибка, выход

•

Все функции математической

библиотеки (-lm man man 3 math)

•

Все функции MinilibX

Разрешенный Libft

ДА

Описание

Вы должны создать “реалистичное” 3D-графическое представление внутренней части лабиринта с точки зрения первого лица.

Вы должны создать это представление, используя принципы отбрасывания лучей, упомянутые ранее.

Ограничения заключаются в следующем:

-

Вы должны использовать miniLibX. Либо версия, доступная в операционной системе, либо из ее исходных текстов. Если вы решите работать с исходными текстами, вам нужно будет применить к вашему libft те же правила, что и те, которые описаны выше в Общих инструкциях часть.

-

Управление вашим окном должно оставаться плавным: переход на другое окно, сворачивание и т.д.

-

Отобразите различные текстуры стен (выбор за вами), которые варьируются в зависимости от того, с какой стороны обращена стена (на север, Юг, Восток, Запад).

5

sub3D

Мой первый RayCaster с miniLibX

-

Ваша программа должна иметь возможность устанавливать для пола и потолка два разных цвета.

-

Программа отображает изображение в окне и соблюдает следующие правила:

-

Клавиши со стрелками влево и вправо на клавиатуре должны позволять вам смотреть влево и вправо в лабиринте.

-

Клавиши W, A, S и D должны позволять вам перемещать точку обзора по лабиринту.

-

Нажатие клавиши ESC должно закрыть окно и полностью завершить работу программы.

-

Нажатие на красный крестик на рамке окна должно закрыть окно и завершить работу программы.

-

Настоятельно рекомендуется использовать изображения minilibX.

-

Ваша программа должна принимать в качестве первого аргумента файл описания сцены с расширением .cub

-

-

Карта должна состоять только из 6 возможных символов: **0** для пустого места,

1

для стены и **N,S,E** или **W** для начальной позиции игрока и

ориентации появления.

Это простая действительная карта:

```
111111
100101
101001
1100N1
111111
```

◦

Карта должна быть закрыта / окружена стенами, в противном случае программа должна вернуть ошибку.

◦

За исключением содержимого карты, каждый тип элемента может быть разделен одной или несколькими пустыми строками.

◦

За исключением содержимого карты, которое всегда должно быть последним, каждый тип элемента может быть установлен в любом порядке в файле.

◦

За исключением карты, каждый тип информации из элемента может быть разделен одним или несколькими пробелами.

◦

Карта должна быть проанализирована так, как она выглядит в файле. Пробелы являются допустимой частью карты, и их обработка зависит от вас. Вы должны уметь анализировать карты любого типа, при условии, что они соответствуют правилам карты.

6

cub3D

Мой первый RayCaster с miniLibX

◦

Для каждого элемента (кроме карты) первой информацией является идентификатор типа (состоящий из одного или двух символов), за которым следует вся конкретная информация для каждого объекта в строгом порядке, например:

* Северная текстура:

```
NET /path_to_the_north_texture
```

- идентификатор: **NET**
- путь к северной текстуре

* Южная текстура:

```
HTAK/path_to_the_south_texture
```

- идентификатор: **SO**
- путь к южной текстуре

* Западная текстура:

```
MBI /path_to_the_west_texture
```

- идентификатор: **МЫ**
- текстура пути на запад

* Восточная текстура:

```
EA /path_to_the_east_texture
```

- идентификатор: **советник**
- путь к восточной текстуре

* Цвет пола:

```
F 220,100,0
```

- идентификатор: **F**
- Цвета R,G,B в диапазоне [0,255]: **0, 255, 255**

* Цвет покрытия:

- идентификатор: **C**
- Цвета R,G,B в диапазоне [0,255]: **0, 255, 255**

Пример обязательной части со сценой minimalist .cub:

Если в файле обнаружена какая-либо неправильная конфигурация любого рода, программа должна завершиться должным образом и вернуть "Error \ n", за которым следует явное сообщение об ошибке по вашему выбору.

Глава V

Бонусная часть

Бонусы будут оценены только в том случае, если ваша обязательная часть будет **ВЫПОЛНЕНА ИДЕАЛЬНО**.

Под СОВЕРШЕННЫМ мы, естественно, подразумеваем, что оно должно быть полным, что оно не может потерпеть неудачу, даже в случае неприятных ошибок, таких как неправильное использование и т.д.

Это означает, что если ваша обязательная часть не наберет ВСЕ баллы во время оценки ваши бонусы будут полностью ПРОИГНОРИРОВАНЫ.

Список бонусов:

-

Столкновения со стеной.

-

Система миникарты.

-

Двери, которые могут открываться и закрываться.

-

анимированный спрайт.

-

Поворачивайте точку обзора с помощью мыши.

Позже вы сможете создавать лучшие игры, не тратя слишком много времени!

Вы можете использовать другие функции или добавлять символы на карту, чтобы завершайте бонусную часть до тех пор, пока их использование оправдано во время вашего оценка.

Вам также разрешено изменять ожидаемый файл сцены форматируйте в соответствии с вашими потребностями.

Будь умным!

9

Глава VI

Примеры

Рисунок VI.1:

Wolfeinstein3D

оригинальная игра using RayCasting.

Рисунок VI.2:

Пример того, как мог бы выглядеть ваш проект в соответствии с обязательной частью.

10

sub3D

Мой первый RayCaster с miniLibX

Рисунок VI.3:

Пример бонусной части с миникартой, текстурами пола и потолка и анимированным знаменитым спрайтом ежа.

Рисунок VI.4:

Еще один пример бонуса с HUD, полосой здоровья, эффектом тени и оружием, который может стрелять

11

sub3D

Мой первый RayCaster с miniLibX

Рисунок VI.5:

Еще один пример бонусной игры с оружием по вашему выбору и игроком, который ищет в потолок

12