



キューブ3D

miniLibXを使った初めてのRayCaster

概要 このプロジェクトは、史上初のFPSとして世界的に有名な「*Wolfenstein 3D*」ゲームにインスパイアされています。このプロジェクトでは、レイキャスティングを研究することができます。あなたのゴールは、迷路の中にダイナミックなビューを作ることです。



バージョン : 9

内容

I	序文	2
II	目標	3
III	共通説明書	4
IV	必須項目 - cub3D	5
V	ボーナスパート	9
VI	事例紹介	10



第一章 序文

1992年にApogee Software社から発売された『Wolfenstein 3D』は、ゲーム史上初の本格的な「ファーストパーソンシューティング」であり、John Carmack氏とJohn Romero氏によってId Software社で開発されました。



図I.1:ジョン・ロメロ氏（左）とジョン・カーマック氏（右）が後世のためにポーズをとっている。

Wolfenstein 3Dは、Doom (Id Software, 1993), Doom II (Id Software, 1994), Duke Nukem 3D (3D Realm, 1996), Quake (Id Software, 1996) など、ビデオゲームの世界ではさらに永遠のマイルストーンとなるゲームの先祖にあたります。

今度は、あなたが歴史を追体験する番です...

第二章 目標

このプロジェクトの目標は、この1年目の目標全てに共通しています。厳密さ、C言語の使用、基本的なアルゴリズムの使用、情報調査など。

cub3Dは、グラフィックデザインプロジェクトとして、ウィンドウ、カラー、イベント、フィルシェイプなど、これらの分野のスキルを向上させることが可能です。

結論として、cub3Dは、数学の遊び心ある実用的な応用を、具体的な内容を理解することなく探求できる、注目すべき遊び場である。

インターネット上で公開されている数多くの資料を参考にしながら、数学を道具として使い、エレガントで効率的なアルゴリズムを作成することができます。



このプロジェクトを始める前に、オリジナルのゲームをテストすることをお勧めします。

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

第三章

共通事項

- プロジェクトはC言語で書かれている必要があります。
- あなたのプロジェクトはNormに則って書かれていなければなりません。ボーナスファイル/関数がある場合、それらはノームチェックに含まれ、内部にノームエラーがある場合は0が返されます。
- 未定義の動作とは別に、関数が予期せず終了する (セグメンテーションフォールト、バスエラー、ダブルフリーなど) ことがないようにしてください。このような場合、プロジェクトは非機能とみなされ、評価時に0点が与えられます。
- ヒープで確保されたメモリ空間は、必要なときに適切に解放されなければなりません。リークは許されません。
- 対象がそれを要求している場合、ソースファイルを要求された出力にコンパイルする Makefile を -Wall、-Wextra、-Werror フラグで提出し、cc を使用し、その Makefile は再リンクしてはいけません。
- Makefileには、少なくとも\$(NAME)、all、clean、fclean、および再
- プロジェクトにボーナスを回すには、Makefile にルールボーナスを含める必要があります、それによってプロジェクトのメイン部分で禁止されている様々なヘッダー、libraries または関数がすべて追加されます。ボーナスは別のファイル _bonus.{c/h} に記述する必要があります。必須部分とボーナス部分の評価は別々に行われます。
- プロジェクトで libft を使用できるようにする場合、そのソースと関連する Makefile を libft フォルダにコピーしておく必要があります。あなたのプロジェクトの Makefile は、その Makefile を使ってライブラリをコンパイルし、その後プロジェクトをコンパイルしなければなりません。
- テストプログラムは提出する必要がなく、採点もされませんが、プロジェクトのテストプログラムを作成することをお勧めします。テストプログラムを作成することで、自分の作品や仲間の作品を簡単にテストすることができます。このようなテストは、特にデフェンスの時に役立つと思います。実際、プレゼンテーションでは、自分のテストや評価する仲間のテストを自由に使用することができます。
- 指定された git リポジトリに作品を提出する。git リポジトリにある作品だけが採点されます。Deepphought があなたの作品を採点することになった場合、そ

れはあなたの相互評価後に行われます。Deepthought の採点中に作品のいずれかのセクションでエラーが発生した場合、評価は停止されます。

第四章

必須項目 - cub3D

プログラム名	キューブ3D
ファイルを提出する	すべてのファイル
メイクファイル	all, clean, fclean, re, bonus.
論証	フォーマット *.cub の地図
外部機能。	<ul style="list-style-type: none">• open、close、read、write、printf、malloc、free、perror、strerror、exit• 数学ライブラリの全機能 (-lm man man 3 math)• MinilibXのすべての機能
リポート認可	はい
商品説明	リアルな」3Dグラフィックを作成する必要があります。迷路の内部を一人称視点で表現したものです。この表現は、先に述べたレイキャスティングの原理を利用して作成する必要があります。

制約条件は以下の通りです。

- miniLibXを使用する必要があります。オペレーティングシステムで利用可能なバージョンか、そのソースのいずれかを使用します。ソースを使用する場合、あなたのlibftに、上記の共通指示の部分に書かれているのと同じルールを適用する必要があります。
- ウィンドウの管理は、他のウィンドウへの切り替え、最小化など、スムーズである必要があります。
- 壁の向き（北、南、東、西）によって異なる壁のテクスチャーを表示（選択可能）。

- プログラムでは、床と天井の色を2種類に設定できるようにする必要があります。
- プログラムは画像をウィンドウに表示し、以下のルールを尊重します。
 - キーボードの左右の矢印キーで、迷路の中を左右に見ることができること。
 - W、A、S、Dキーで迷路の中で視点を移動させる必要があります。
 - ESCを押すと、ウィンドウが閉じられ、プログラムがきれいに終了するはずです。
 - ウィンドウの枠にある赤い十字をクリックすると、ウィンドウが閉じられ、プログラムがきれいに終了するはずです。
 - minilibXの画像の使用を強く推奨します。
- プログラムは、第一引数にシーン記述ファイル（.cub）を取る必要があります。を拡張しています。
 - マップは6つの文字のみで構成されなければならない。0は何もない空間、1は壁、N、S、E、Wはプレイヤーのスタート位置とスポーニングの方向です。
これはシンプルな有効マップです。

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

- マップは壁で囲まれていなければならない。そうでない場合、プログラムはエラーを返さなければならない。
- 地図の内容を除いて、各要素は1つ以上の空白行で区切ることができる。
- マップコンテンツが必ず最後になる以外は、各種要素をファイル内で任意の順番で設定することができます。
- 地図以外の要素では、各情報を1つ以上のスペースで区切ることができる。
- マップはファイルの見た目通りにパースされる必要があります。スペースはマップの有効な一部であり、その扱いはあなた次第です。マップのルールを尊重する限り、どのような種類のマップでもパースできなければなりません。

- 。各要素（マップを除く）の最初の情報は、タイプ識別子（1文字または2文字で構成）であり、その後、各オブジェクトの具体的な情報が、以下のような厳密な順序で続きます。

- * ノーステクスチャー

```
NO ./path_to_the_north_texture
```

- 識別子です。NO
- 北方領土への道

- * South texture:

```
SO ./path_to_the_south_texture
```

- の識別子を使用しています。SO
- 南方領土への道

- * ウェストテクスチャー

```
WE ./path_to_the_west_texture
```

- を識別することができます。WE
- 西暦のパス

- * イーストの質感。

```
EA ./path_to_the_east_texture
```

- の識別が可能です。EA
- とうろくけいせい

- * 床の色。

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

- の識別が可能です。F
- R,G,Bの色を範囲[0,255]で指定します。0, 255, 255

* シーリングカラー

```
C 225,30,0
```

- の識別子を使用しています。C
 - R,G,Bの色を範囲[0,255]で指定します。0, 255, 255
- ミニマムな.cubのシーンで必須部分の例。

```
NO ./path_to_the_north_texture
SO ./path_to_the_south_texture
WE ./path_to_the_west_texture
EA ./path_to_the_east_texture

F 220,100,0
C 225,30,0

      11111111111111111111111111111111
      10000000001100000000000001
      10110000011100000000000001
      10010000000000000000000001
1111111111011000001110000000000001
10000000001100000111011111111111
11110111111111011100000010001
111101111111110111010010001
11000000110101011100000010001
100000000000000001100000010001
100000000000000001101010010001
11000001110101011111011110N0111
11110111 1110101 101111010001
11111111 1111111 111111111111
```

- もしファイルに何らかの設定ミスがあった場合、プログラムは正しく終了し、"Errorrn" に続いて任意の明示的なエラーメッセージを返さなければなりません。

第五章 ボーナス

パート



ボーナスは、あなたの必須パーツが**PERFECT**である場合にのみ評価されます。完璧というのは、当然ながら完全である必要があり、間違った使い方などの厄介なミスがあったとしても、**失敗することはない**ということです。つまり、あなたの必須項目が採点中にすべてのポイントを獲得できなかった場合、あなたのボーナスは完全に無視されることになります。

ボーナスリスト

- 壁の衝突
- ミニマップシステム。
- 開閉可能なドア。
- アニメーションのスプライトです。
- マウスで視点を回転させる。



時間を無駄にすることなく、より良いゲームを作ることができるようになります。



ボーナスパートを完成させるために、他の機能を使用したり、地図上にシンボルを追加したりすることは、評価時に正当化される限り、許可されます。また、想定されるシーンのファイル形式を、あなたのニーズに合うように変更することも可能です。 賢くなろう

第VI章 例



図 VI.1: RayCastingを用いたWolfenstein3Dオリジナルゲーム.

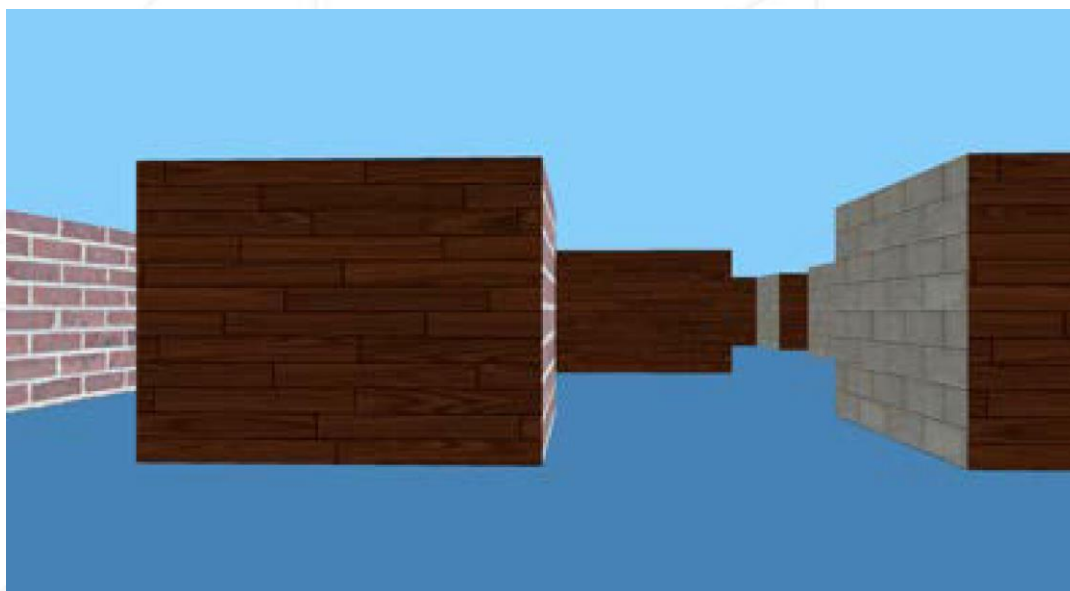


図 VI.2: 必須項目に沿ったプロジェクトのイメージ例



図 VI.3: ミニマップ、床と天井のテクスチャ、有名なハリネズミのSpriteをアニメーション化したボーナスパーツの例。



図6.4 : HUD、ヘルスバー、シャドーエフェクト、射撃可能な武器を持つボーナスの別の例

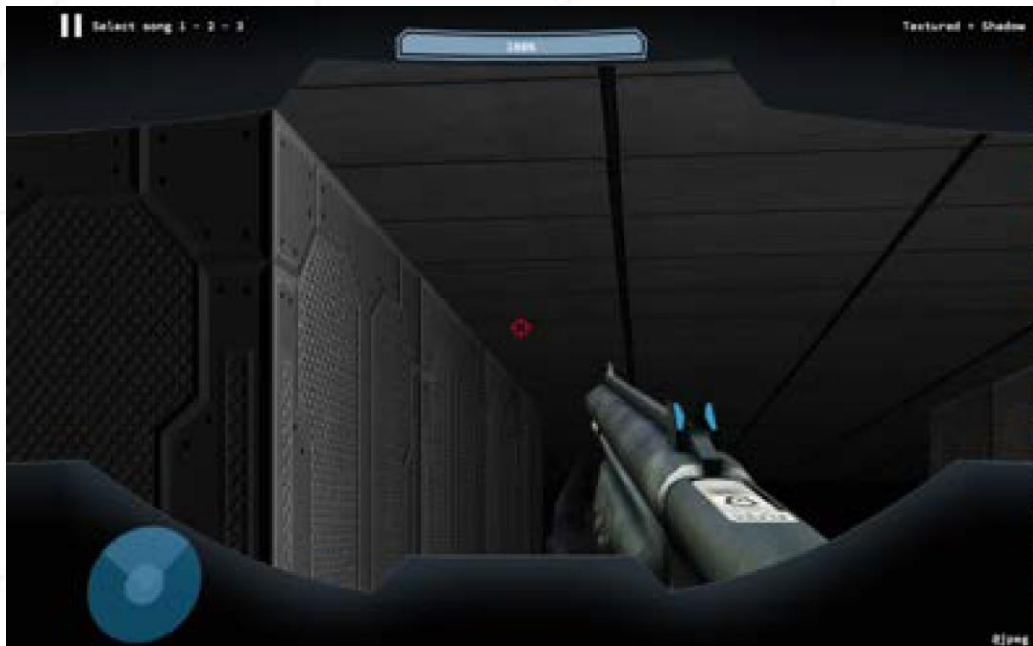


図6.5: 選択した武器とプレイヤーが天井を見るボーナスゲームの別の例