



## カブ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 である世界的に有名な Wolfenstein 3D ゲームからインスピレーションを得ています。レイ キャスティングを探索できます。目標は、迷路の中に動的なビューを作成し、その中で道を見つけることです。



バージョン: 10

# コンテンツ

私	序文	2
II	目標	4
III	共通指示	5
IV	必須部分 - cub3D	7
V	ボーナスパート	11
VI	例	12
VII	提出とピア評価	15



## 第1章

### 序文

超有名なジョン・カーマックとジョン・ロメロによって Id Software によって開発され、1992 年に Apogee Software によって公開された Wolfenstein 3D は、ビデオ ゲーム史上初の真の「一人称シューティング ゲーム」です。



図 1.1: ジョン・ロメロ (左) とジョン・カーマック (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) などのゲームの祖先であり、ビデオ ゲームの世界における永遠のマイルストーンとなっています。

さあ、歴史を再現する番です...



Wolfenstein 3Dは元々ナチスドイツを舞台にしており、最終的には不快な内容になるかもしれませんが、このゲームの写真と歴史は、技術的な理由とポップ/オタク文化の理由からのみ提供されています。このゲームは1980年代に傑作と見なされていました。

両方。

## 第2章

### 目標

このプロジェクトの目的は、厳密さ、C の使用、基本的なアルゴリズムの使用、情報調査など、この最初の 1 年間のすべての目的と同様です。

グラフィック デザイン プロジェクトとして、cub3D を使用すると、ウィンドウ、色、イベント、図形の塗りつぶしなどの分野でスキルを向上させることができます。

結論として、cub3D は、詳細を理解する必要なく、数学の遊び心のある実用的な応用を探索できる素晴らしい遊び場です。

インターネット上で入手可能な多数の文書の助けを借りて、エレガントで効率的なアルゴリズムを作成するためのツールとしての数学。



よろしければ、このプロジェクトを開始する前にオリジナルのゲームをテストできます: <http://users.atw.hu/wolf3d/>

## 第3章

### 共通手順

- プロジェクトは C 言語で記述する必要があります。
- プロジェクトは、標準に従って記述する必要があります。ボーナス ファイル/関数がある場合は、標準チェックに含まれ、標準エラーがある場合は 0 が返されます。
- 関数は、未定義の動作を除き、予期せず終了してはなりません (セグメンテーション エラー、バス エラー、二重解放など)。予期せず終了した場合、プロジェクトは機能しないものとみなされ、評価時に 0 が与えられます。
- ヒープに割り当てられたメモリ空間は、必要に応じて適切に解放する必要があります。リークはありません許容されます。
- 主題で必要な場合は、フラグ -Wall、-Wextra、-Werror を使用してソース ファイルを必要な出力にコンパイルする Makefile を提出する必要があります。cc を使用し、Makefile を再リンクしないでください。
- 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提出するには、プロジェクトのメイン部分で禁止されているさまざまなヘッダー、ライブラリ、または関数をすべて追加するルール ボーナスを Makefile に含める必要があります。件名で他に何も指定されていない場合、ボーナスは別のファイル \_bonus.{c/h} になければなりません。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 の使用が許可されている場合は、そのソースと関連する Makefile を、関連する Makefile とともに libft フォルダにコピーする必要があります。プロジェクトの Makefile は、その Makefile を使用してライブラリをコンパイルしてから、プロジェクトをコンパイルする必要があります。
- この作業は提出する必要はなく、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 プログラムにより、自分の作業と仲間の作業を簡単にテストできます。これらのテストは、特に審査中に便利です。審査中は、自分のテストや評価対象の仲間のテストを自由に使用できます。
- 割り当てられた git リポジトリに作業を提出してください。git リポジトリ内の作業のみが採点されます。Deepthought があなたの作業の採点を担当する場合は、

ピア評価の後、Deepthought の採点中に作業のどのセクションでもエラーが発生した場合、評価は停止します。

## 第4章

### 必須部分 - cub3D

プログラム名	cub3D
ファイルを提出する	すべてのファイル、クリー
Makefile 引数	ン、fclean、re、ボーナスマップ、*.cub 形式
外部関数。	
	<ul style="list-style-type: none"> <li>• 開く、閉じる、読む、書く、 printf、malloc、free、perror、strerror、exit、 gettimeofday</li> <li>• 数学のすべての機能 ライブラリ (-lm man man 3 math)</li> <li>• MinilibXのすべての機能</li> </ul>
Libft 認定はい	
説明	一人称視点で迷路内部の「リアルな」3D グラフィック表現を作成する必要があります。この表現は、前述のレイキャスティングの原則を使用して作成する必要があります。

制約は次のとおりです。

- miniLibX を使用する必要があります。オペレーティング システムで使用可能なバージョン、またはそのソースからのバージョンのいずれかです。ソースを使用する場合は、上記の共通手順の部分に記載されているのと同じルールを libft に適用する必要があります。
- ウィンドウの管理はスムーズでなければなりません。別のウィンドウに切り替えるには、  
ダウ、最小化など。
- 異なる壁テクスチャを表示します（選択はあなた次第です）。  
壁が向いている側（北、南、東、西）。



- プログラムでは、床と天井の色を 2 つの異なる色に設定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、次の規則に従います。
  - ¶ キーボードの左矢印キーと右矢印キーを使用すると、迷路内で左右を見ることができます。
  - ¶ W、A、S、Dキーで視点を移動する必要があります。  
迷路。
  - ¶ ESC キーを押すとウィンドウが閉じられ、プログラムが正常に終了する必要があります。
  - ¶ ウィンドウのフレーム上の赤い十字をクリックすると、ウィンドウが閉じられ、プログラムが正常に終了する必要があります。
  - ¶ minilibX のイメージの使用を強くお勧めします。
- プログラムは最初の引数として、cubのシーン記述ファイルを受け取る必要があります。  
拡大。
  - ¶ マップは、空きスペースの場合は0、壁の場合は1、プレイヤーの開始位置とスポーン方向の場合はN、S、E、またはWの6つの文字のみで構成する必要があります。

これは単純な有効なマップです:

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

  - ¶ マップは閉じられているか壁で囲まれている必要があります。そうでない場合はプログラムはエラー。
  - ¶ マップコンテンツを除いて、各要素は1つ以上の  
空行がさらにあります。
  - ¶ 常に最後にあるマップコンテンツを除いて、  
要素はファイル内で任意の順序で設定できます。
  - ¶ マップを除き、要素からの各情報タイプは 1 つ以上のスペースで区切ることができます。
  - ¶ マップはファイル内のとおりに解析する必要があります。スペースはマップの有効な部分であり、処理はあなた次第です。マップのルールを尊重する限り、あらゆる種類のマップを解析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、その後に各オブジェクトのすべての固有情報が厳密な順序で続きます。例:

北のテクスチャ:

```
いいえ ./path_to_the_north_texture
```

- ・ 識別子: NO
- ・ 北への道のテクスチャ

南のテクスチャ:

```
つまり、 ./path_to_the_south_texture
```

- ・ 識別子: SO
- ・ 南への道のテクスチャ

西側のテクスチャ:

```
WE ./西への道テクスチャ
```

- ・ 識別子: WE
- ・ 西への道のテクスチャ

東のテクスチャ:

```
EA ./東へのテクスチャへのパス
```

- ・ 識別子: EA
- ・ 東への道テクスチャ

床の色:

```
220,100,0 イギリス
```

- ・ 識別子: F
- ・ R、G、B 色の範囲は [0,255]: 0、255、255

天井の色:

225,30,0 円

・ 識別子: C

・ R,G,B 色の範囲は [0,255]: 0,255,255

¶ ミニマリスト.cubシーンの必須部分の例:

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

```
220,100,0 イギリス
225,30,0 円
```

```
11111111111111111111111111111111
100000000011000000000000000001
101100000111000000000000000001
100100000000000000000000000001
111111110110000011100000000000001
```

```
10000000001100000111011111111111110111111101111111011100000010001 1111011111110111010100 10001 110000001101010111000000.0001 1000000000.0000011000
11000001110101011111011110N0111
11110111 1110101 101111010001
11111111 11111111 111111111111
```

¶ ファイル内で何らかの設定ミスが見つかった場合、プログラムは適切に終了し、「Error\n」に続いて明示的に選択したエラー メッセージを返す必要があります。

## 第5章

### ボーナス部分



ボーナスは必須部分が完璧だった場合にのみ評価されます。  
完璧とは、当然、誤った使用などの厄介なミスがあった場合でも、完全で、故障しないことを意味します。  
つまり、採点中に必須部分がすべてのポイントを獲得しなかった場合、ボーナスは完全に無視されます。

ボーナスリスト:

- 壁との衝突。
- ミニマップ システム。
- 開閉可能なドア。
- アニメーションスプライト。
- マウスで視点を回転します。



後でもっと良いゲームが作れるようになるので、あまり無駄にしないでください  
時間！



評価中にその使用が正当化される限り、ボーナス部分を完了するために他の機能を使用したり、マップ上にシンボルを追加したりすることができます。また、想定されるシーン ファイル形式をニーズに合わせて変更することもできます。賢く行いましょう。

## 第六章

### 例



図 VI.1: RayCasting を使用した Wolfenstein3D オリジナル デザインのレプリカ。

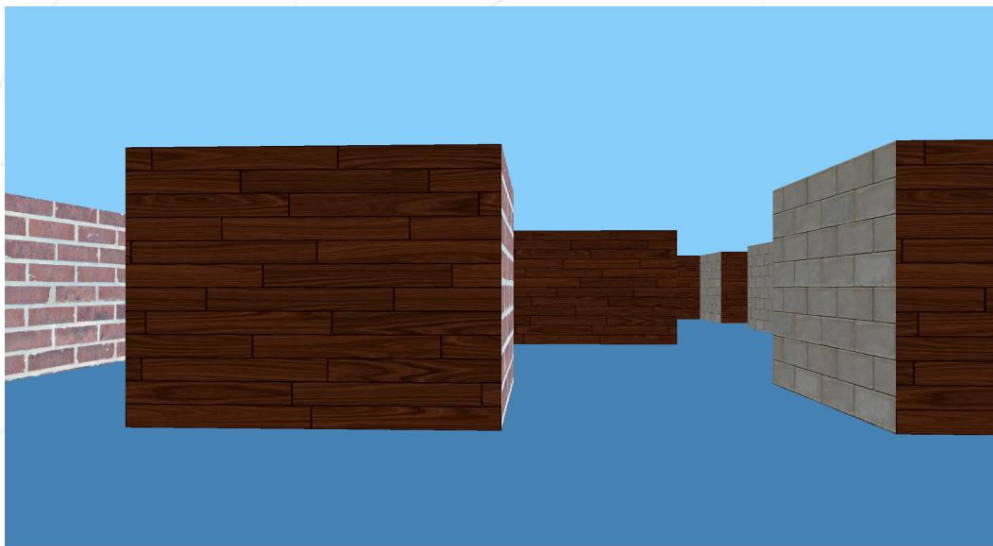
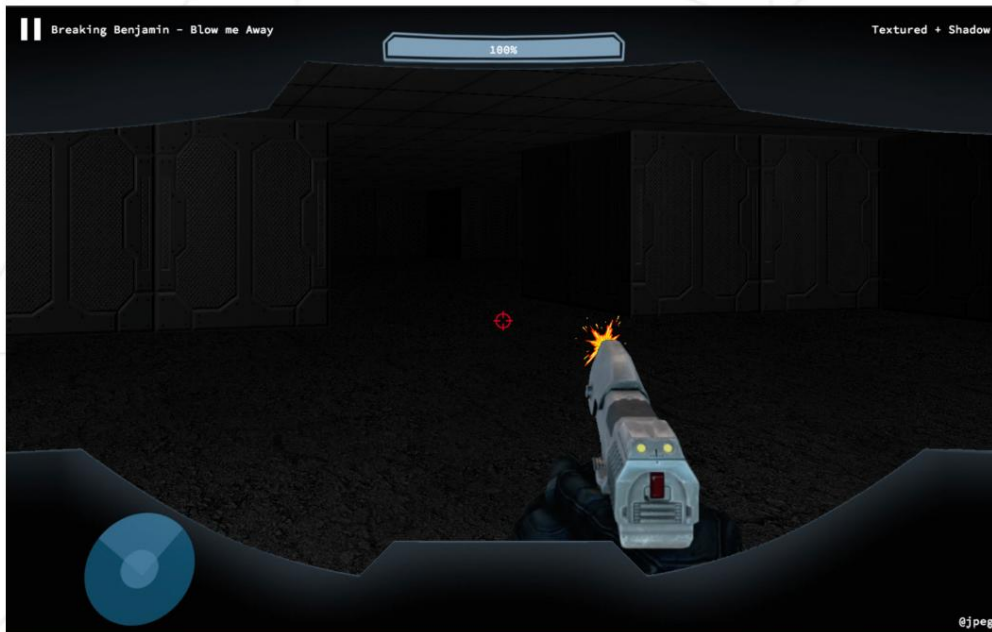


図 VI.2: 必須部分に従ってプロジェクトがどのようになるかを示した例。



図 VI.3: ミニマップ、床と天井のテクスチャ、アニメーション化された有名なハリネズミの спраイトを含むボーナス部分の例。



図VI.4: HUD、ヘルスバー、影効果、射撃可能な武器を備えたボーナスの別の例



図VI.5:武器を選択してプレイヤーが天井を見ているボーナスゲームの別の例

注意: 上記の画像には、文書全体と同様に著作権で保護された画像が含まれている場合があります。  
教育目的で作成されたこの主題は、フェアユースのガイドラインに該当すると考えています。

## 第7章

### 提出とピア評価

いつものように Git リポジトリに課題を提出してください。審査中はリポジトリ内の作業のみが評価されます。ファイル名が正しいことを必ず再確認してください。



??????????? XXXXXXXXXX = \$3\$796ba5a53df1352e06cc7b0f3ad2a41d