



So Long

そして、たくさんのお魚をありが
とうございました

概要

このプロジェクトは、とても小さな2Dゲームで
す。

その目的は、テクスチャやスプライトなど、ごく
基本的なゲームプレイの要素を扱わせることです
。

バージョン : 1

内容

I	序文	2
II	目的	3
III	共通説明書	4
IV	必須項目	6
	IV.1ゲーム... ..	7
	IV.2 グラフィック管理	7
	IV.3マップ.....	8
V	ボーナスパート	9
VI	事例紹介	10
VII	提出と相互評価	11

第一章 序文

開発者であることは、自分のゲームを作る上ではとても良いことです。

しかし、良いゲームには良いアセットが必要です。2Dゲームを作るには、タイル、タイルセット、スプライト、スプライトシートを探さなければなりません。

幸いなことに、才能あるアーティストの中には、itch.ioのようなプラットフォームで自分の作品を喜んでシェアしてくれる人もいます。

いずれにせよ、他人の作品を尊重することを心がけましょう。

第二章 目的

コンピュータグラフィックスの基本的な作品を作る時間です

ウィンドウ管理、イベント処理、色、テクスチャなどのスキルを向上させることができます。

学校のグラフィカルライブラリ：MiniLibXを使用することになります。このライブラリは内部で開発されたもので、ウィンドウを開いたり、画像を作成したり、キーボードやマウスのイベントを処理したりするのに必要な基本的なツールが含まれています。

その他の目標は、厳密であること、C言語のプログラミングのレベルアップ、基本的なアルゴリズムの使用、情報調査など、この1年目の他のどの目標とも似ています。

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、主体が何も指定しない場合は、別のファイル `_bonus.{c/h}` に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。`Deepthought` があなたの作品の採点を担当する場合、採点は以下のように行われます。

So
fish!

LongAndthanks for all the

相互評価後Deepthoughtの採点中に、あなたの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4章 必須項目

プログラム名	そうそう
ファイルを提出する	Makefile、*.h、*.c、マップ
メイクファイル	NAME、All、Clean、Fclean、Re
論証	フォーマット *.ber の地図
外部機能。	<ul style="list-style-type: none"> • open, close, read, write, malloc, free, perror, strerror, exit. • 数学ライブラリの全機能 (-lm コンパイラオプション, man man 3 math) • MiniLibXの全機能 • ft_printf と同等の YOU コード。
リポート認可	はい
商品説明	<p>イルカが登場する基本的な2Dゲームを作成する必要があります。</p> <p>は、魚を 食べた後、地球を脱出します。イルカ、魚、地球の代わりに、好きなキャラクター、好きな収集品、好きな場所を使うことができます。</p>

あなたのプロジェクトは、以下のルールに従わなければなりません。

- MiniLibXを使用する必要があります。学校のマシンで利用できるバージョンか、そのソースを利用してインストールする。
- ソースファイルをコンパイルするMakefileを提出する必要があります。再リンクはしてはいけません。
- プログラムは、.berで終わる地図記述ファイルをパラメータとして受け取る必要があります。を拡張しています。

IV.1 ゲーム

- プレイヤーは、マップ上に存在するすべてのアイテムを集め、最短距離で脱出することが目的です。
- 主人公の移動は、W、A、S、Dキーで行う必要があります。
- プレイヤーは上下左右の**4方向**に動ける必要があります。
- プレイヤーが壁の中に入っていけないようにする。
- **一手ごとに、現在の移動回数をシェルに表示する必要があります。**
- **2Dビュー**（トップダウンまたはプロファイル）を使用する必要があります。
- ゲームはリアルタイムである必要はないんです。
- 例ではイルカをテーマにしていますが、自分の好きな世界を作ることができます。

IV.2 グラフィック 管理

- あなたのプログラムは、画像をウィンドウに表示しなければなりません。
- ウィンドウの管理（別のウィンドウへの切り替え、最小化など）がスムーズであること。
- ESCを押すと、ウィンドウを閉じ、プログラムをきれいに終了させる必要があります。
- ウィンドウのフレームにある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必要があります。
- MiniLibXの画像の使用は必須です。

IV.3 地図

- マップは、「壁」「収集物」「空き地」の3つの要素で構成する必要があります。
- この5文字だけで地図を構成することができる。
 - 0** を、空いたスペースに置く。
 - 1** を壁面に設置します。
 - コレクターズアイテムに**C**。
 - Eで地図の出口を表示します。
 - Pはプレイヤーのスタートポジションを表します。

ここでは、簡単な有効マップを紹介します。

```
11111111111111
10010000000C1
1000011111001
1P0011E000001
11111111111111
```

- マップには少なくとも**1つの出口、1つの収集品、1つのスタート地点**が含まれていなければなりません。
- 地図は長方形である必要があります。
- マップは閉じていなければならない／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。
- 地図上に有効なパスがあるかどうかを確認する必要はありません。
- 上記のルールを守る限り、どんな種類の地図でもパースできること。
- 最小限の.berマップのもう一つの例。

[illegible]

- もし、ファイルに何らかの設定ミスがあった場合、プログラムはきれいな形で終了し、"Errorn "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

通常は、自分でオリジナルの追加機能を開発することが推奨されます。しかし、この後、もっと面白いグラフィックプロジェクトが待っています。それらはあなたを待っています!この課題ではあまり時間を無駄にしないように!

ボーナスパートを完成させるために他の機能を使用することは、評価の際にその使用が正当化される限り、許されます。賢くならう!

以下のような場合、さらにポイントが加算されます。

- 敵のパトロールに触れると負けるようにする。
- スプライトのアニメーションを追加する。
- 移動回数をシェルに書き込まず、直接画面に表示する。



ボーナスパーツは、必須パーツがPERFECTである場合にのみ査定されます。パーフェクトとは、必須パーツが統合的に行われ、誤動作することなく動作することを意味します。必須条件をすべてクリアしていない場合、ボーナスパーツの評価は一切行われません。

第VI章 例



グラフィックデザインのひどいセンスを示す長い例（ほとんどボーナスポイントに値する）！

第七章

提出と相互評価

通常通り、Git リポジトリに課題を提出してください。レポート内の作品だけが、ディフェンスで評価されます。ファイル名が正しいかどうか、遠慮なく再確認してください。

これらの課題は、プログラムによる検証は行われませんので、必須ファイルを提出し、要件を満たしていれば、自由にファイルを整理してください。