Unityインターハイ2018

PERVERSE

作品概要資料

この資料は審査員があなたの作品を理解し、
正しく審査するためのものです。
作品の概要、ルール、操作などをこの資料に記載してください。

作品概要

区分	説明
作品名	PERVERSE
ゲーム概要	PERVERSEは上下左右逆に動く2つのブロックを動かしてゴールに持っていくゲームです。 また、友達の遊んだマップを共有して遊んだり、サーバーに遊んだマップをアップロードすることで実際に世界中の人がプレイしたマップで対戦し、レートで競うことができます。
ゲームの目的	プレイヤーがゲームをプレイし、プレイヤー同士で競うことによって楽しむこと
ゲームクリア条件	2つのキューブが同時にゴールにたどり着けばゴールですが、タイムアタック制なので制限時間が0になったらクリア（というより終わり）です。
ゲームオーバー条件	ゲームクリア、ゲームオーバーの正確な区別がないので、「ゲームクリア条件」と同様に制限時間が0になったら終わり、とします。
備考、注意点など	必ずWifiが繋がる環境でプレイしてください。 Android4.1以上のスマートフォンで遊んで下さい。タブレットなどでは表示が崩れる可能性があります。（特にアスペクト比が16:9のスマートフォンに一番最適化されています） 音が聞ける環境でプレイするとより一層楽しめます！ **Windows, WebGL, Macは動作保証はしていません。** Windowsでは動作確認ができましたが、いくつかの機能は利用できません。ウィンドウサイズを可変にしているので、16:9になるべく近づけるようにしてください。

ゲーム画面など

ゲーム画面のスクリーンショットや図、画像資料などがあれば掲載してください。

次のページからゲームの工夫点などを説明した資料を載せています！
長いので大雑把に読んでいただければありがたいです！

PERVERSE

説明資料

浅野啓・田村来希

ゲームモードについて

エンドレスモード

1. 制限時間内に無制限に自動生成されるステージ(マップ)を解いていくモード
2. 1ステージクリアごとに制限時間が少しずつ増える
3. 制限時間が0になるとゲームオーバー
4. 終了時、自分のレートとユーザーIDと生成されたマップデータ(JSON)が自動的にデータベースにアップロードされる 。
   (アップロードされるマップ数=解いたステージ数+1)

対戦モード

1. プレイ時にデータベースから自分とレートが最も近い人がアップロードしたステージを取得
2. 制限時間内に全部のステージをクリアすれば勝ち、一つ残せば引き分け、その他は負けとする
3. 勝敗によって自分のレートを変動させる(レートについては次頁)

マップコードモード

- 自分が遊んだマップを短い文字列に圧縮し、共有することができる(この文字列をマップコードという)
- 共有された文字列をゲーム内で貼り付けると、コードからマップの情報へと変換され、同じマップを遊ぶことができる

レートについて

レートについて

- レートは一人ひとりのプレイヤーがもつゲームにおける強さを示す値
- 最初は1000から始まり対戦(バトル)モードでの勝敗によって変動する

高レートの人がよりレートが上がりにくく、低レートの人がよりレートを下げにくくするために、レート計算の式は以下の通りにした。

- (補正値)=(相手のレート-自分のレート)/15
- 勝ち: (新レート)=(現在のレート)+(30+補正値)
- 負け: (新レート)=(現在のレート)-(30-補正値)
- 引き分け: (新レート)=(現在のレート)+(補正値)
- ただし (補正値)>25ならば(補正値)=25、または(補正値)<25ならば(補正値)=-25

アルゴリズムについて

マップ自動作成アルゴリズム

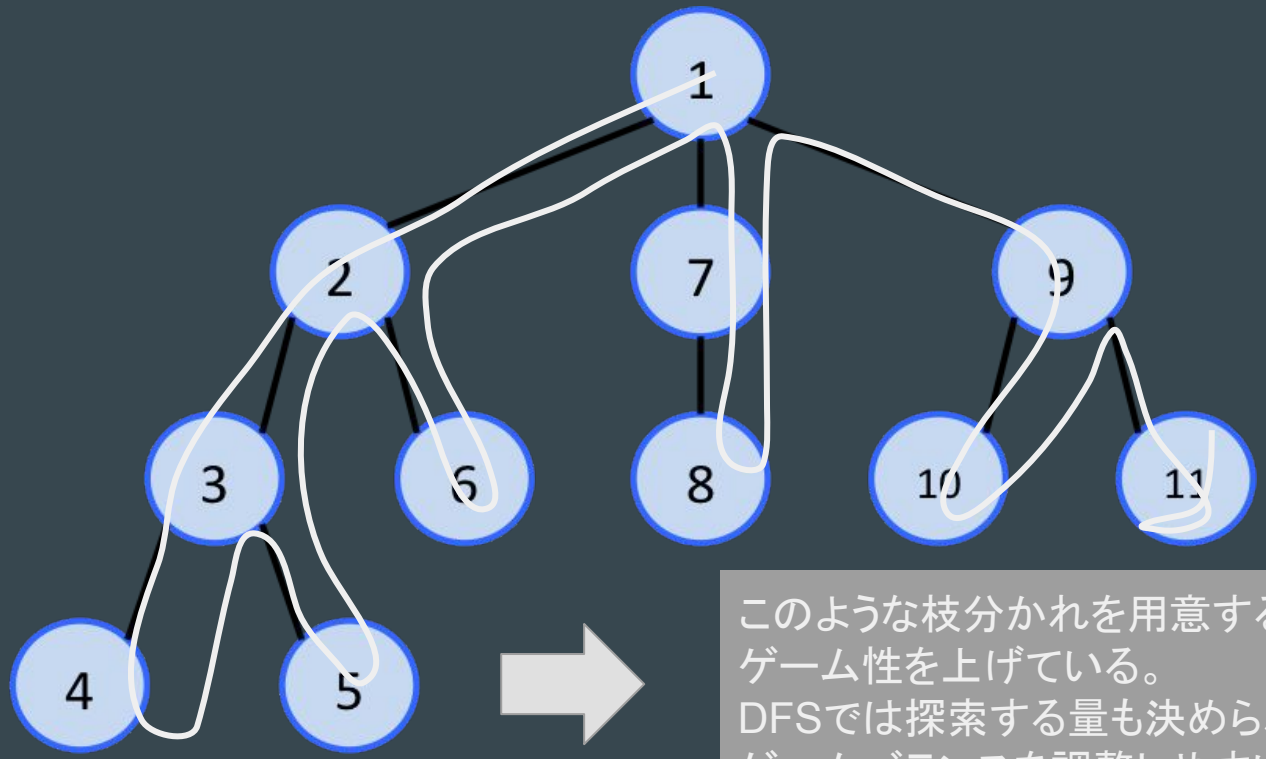
要点

- メインのモード（普通のモード）ではマップは**自動作成**される
- 自動作成には深さ優先探索(DFS)が使われている

マップ自動作成の手順を簡単に

1. 方向を決める
2. 距離を決める
3. すでに通る予定または壁のフラグが進んだ先になければ、通ることを確定させる
4. これを深さ優先探索で探索し、シミュレートする

DFSの遷移



このような枝分かれを用意することで、
ゲーム性を上げている。
DFSでは探索する量も決められるために
ゲームバランスを調整しやすい

マップコードのアルゴリズム

要点

- 自分が遊んだマップを短い文字列に圧縮し、共有することができる(この文字列をマップコードという)
- 2進数を256進数に変換することで短くしている。

マップコード

- なるべく短い文字列で、最大25×25の配列を表現したい

→256進数を定義した

```
'Σ','Τ','Υ','Φ','Χ','Ψ','Ω','α','β','γ','δ','ε','ζ','η','θ','ι','κ','λ','μ','ν','ξ','ο','π','ρ','σ','τ','υ','φ','χ','ψ','ω', '0','≠',
'≦','≧','<','>','≪','≫','∞','∽','∝','∴','∵','∈','∋','⊆','⊇','⊂','⊃','∪','∩','∧','∨','¬',
'⇒','⇔','∀','∃','∠','⊥','⌒','∂','∇','≡','√','∫','∬','━','┃','┌','┐','┘','└','├','┬','┤',
'┴','┼','━','┃','┌','┐','┘','└','├','┬','┤','┴','┼','├','┬','┤','┴','┼','├','┬','┤','┴',
'┼','#','&','*','@','§','※','〓','♯','♭','♪','†','‡','¶','仝','々','〆','ー','～','￣','＿','—','-','∥',
'｜','／','＼','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f','g','h','i','j','k','l','m','n'
,'o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D','E','F','G','H','I','J','K','L','M',
'N','O','P','Q','R','S','T','U','V','W','X','Y','Z','+','/',
'А','В','Г','Δ','Е','Z','Η','Θ','Ι','Κ','Λ','Μ','Ν','Ξ','Ο','Π','Ρ','Α','Б','В','Г','Д','Е','Ё','Ж','З','И','Й',
'К','Л','М','Н','О','П','Р','С','Т','У','Φ','Χ','Ц','Ч','Ш','Щ','Ъ','Ы','Ь','Э','Ю','Я','а','б','в','г','д','е'
,'ё','ж','з','и','й','к','л','м','н','о','п','р','с','т','у','ф','х','ц','ч','ш','щ','ъ','ы','ь','э','ю','я',
'+','−','±','×','÷','=','≒'
```

マップコード

壁を1、道を0とおいた二進数で表す

マップコード

2進数を256進数で表すと

11100011101 ➡ 7 29 ➡ α ψ となる

11マスの情報をわずか2文字で表せられる！！

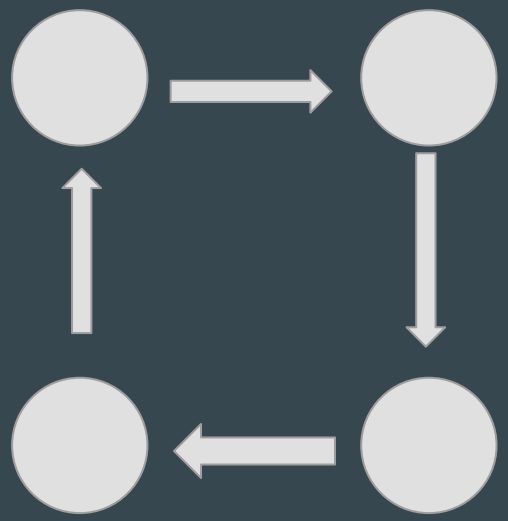
そしてSNSを通じて友達と共有できる

ヒント表示アルゴリズムについて

要点

- プログラムが最短ルートを計算し、プレイヤーが行き詰まったときはヒントを与える
- 最短ルートを求めるには幅優先探索(BFS)が使われている

ヒント表示アルゴリズムについて



普通にやると、左のように頂点が遷移して無限ループしてしまう

↓

すでに探索した場所は2回以上通らないように保存しておく！（メモ化）

↓

無限ループを防げる！

ヒント表示アルゴリズムについて

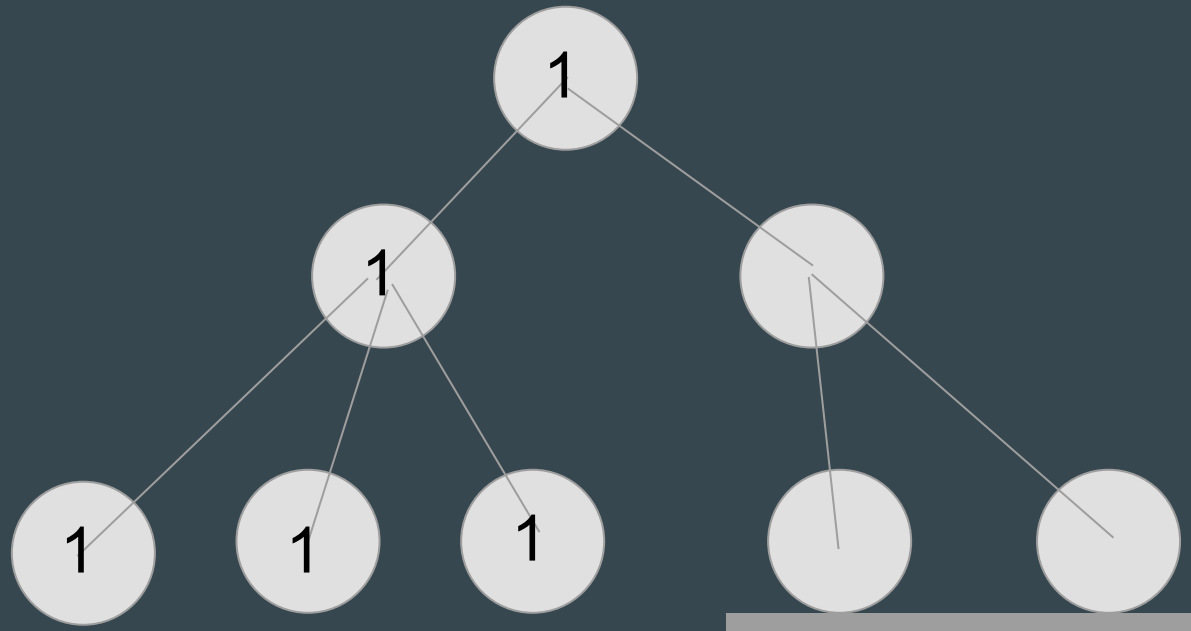
メモ化の工夫

普通にやると、2つの位置を保存するので1e8の大きさの配列が必要で、**オーバーフロー**する

→”1(x1)(y1)(x2)(y2)”という文字列をつくり、さらに数字に再び変換する

→これを保存し、探すときは**二分探索**を使うことで**対数時間、省メモリーでメモ化ができる！**

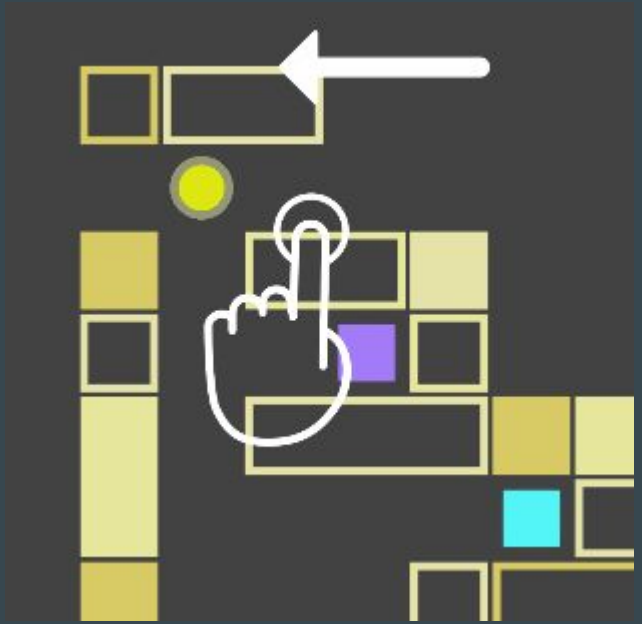
BFSの遷移



このような枝分かれを用意することで、
ゲーム性を上げている。
DFSでは探索する量も決められるために
ゲームバランスを調整しやすい

ヒント表示アルゴリズムについて・結果

行き詰まったらヒントを表示してくれる！！



データベースについて

データベース、通信について

- herokuのpostgreSQLを使用。
- 3つのテーブルを用意した。(残り一つは後述)
  - ユーザー情報(uinfo)テーブル
    カラム:(ユーザーID、ユーザー名、ハイスコア、レート、最高レート、勝ち数、対戦回数)
  - マップデータ(map)テーブル
    カラム:(アップロード者のID、アップロード者のマップアップロード時のレート、マップデータ、マップ識別用のID)
- heroku上にpSQLにアクセスするwebページをPHPで作成し、そのページにUnityのWWWクラスを用いてwebアクセスを行う。
  このときPOSTするテキストにクライアントが要求するリクエストの情報を含ませる。

ユーザー情報について

- ゲームの初回起動時にユーザー登録を行う。

  (サーバーに任意のユーザー名を送信すると、ユニークなユーザーIDが発行され、レートが1000に設定される)

- タイトル画面ロード時に、ユーザーデータの同期処理を行う。

  (クライアントが保存しているユーザー IDを用いてSQLサーバーの持つユーザー情報を取得する)

- リザルト画面ではスコア、レート、勝ち数、対戦回数のアップロードが行われる。

具体的なデータの送信例

http://tamachanapi.herokuapp.com/getData.php でクライアントから以下のようなテキストをPOSTすることでサーバーとのデータの受け渡しを単純にした。

- /uploadMap {"uid":73,"mapcode":["1┤βκd┬#∝Y","1γ┴XΞ3#∽η"]}

  マップのアップロードリクエスト:ユーザーIDとマップコードのJSON

- /uploadScore {"uid":73,"score":1}

  スコアのアップロードリクエスト:サーバー側でハイスコアの更新も行う

データベースの内容は下のURLから見ることができます

URL : http://tamachanapi.herokuapp.com/getData.php

サーバー側のプログラム)

URL : https://github.com/simauma1203/Single-Programs/blob/master/getData.php

対戦履歴について

- 対戦履歴の管理は同じ対戦モードで同じマップと二回以上マッチングしないためである。(複数回同じマップとマッチングするのは、ユーザーにとって退屈であるため)
- エンドレスモードでプレイしたマップが自動アップロードされる際に、データにハンドル(ユーザーIDのようにデータを識別するためのユニークな番号)が付与される。
- 対戦履歴を保存するための対戦履歴(history)テーブルを用意した。

  カラム: (ユーザーID、プレイしたマップのハンドル(下記)の配列)
- 対戦モードでマップを取得する際に、サーバー側でmapテーブルとhistoryテーブルを照合し、未プレイのマップを返すようにした。

音楽について

音楽

- 自作
- 落ち着きのある雰囲気にしたかった（タイムアタック制がメインとはいえ、パズルゲームには落ち着いて遊ぶことのほうが重要なため）
- 背景にはコーラスを入れ、主音にはピアノを少しだけ入れることによって落ち着いた音楽が出来上がった
- プレイ動画の音楽にもなっているので、ぜひ聞いてみてください！

モーション・デザインについて

キューブのモーション

仕組み

1. キューブの位置の遷移をグラフで表す
2. グラフで表すことである時間における位置を簡単に求めることができる
3. 軌跡を表すため、0.15秒前の位置と現在位置の距離を横幅としたキューブ（軌跡）をおく
4. 移動中のみメインのキューブにブラーをかけ、モーションブラーに似たモーションを表す

デザイン

- 色を6種類用意
- 設定画面で選べる（6色+ランダム）
- タイトル画面の色、スコア表示、タイムバーの色なども選択した色によって自在に変わる

（頑張ったということをアピールしたかっただけです …）

最後まで読んでいただき、有難うございました！！

操作方法

テキストで操作方法を記入してください。ページ数が足りない場合や、モバイルやタブレット端末、
ヘッドマウントディスプレイ、キネクトなど特殊な操作が必要となる場合はご自身で操作説明ページをご用意ください。

上下左右のフリックで2つのキューブが逆方向に動きます！

- 青キューブ→フリックした方向に
- 紫キューブ→フリックした方向の反対に