

FPGA による迷路ゲーム器の製作

研究者：伊藤 伸
伊東 巧成

1 はじめに

今回の研究における達成目標は、FPGA を使って 2D と 3D の両方での迷路ゲーム制御回路を製作することだ。迷路ゲームを製作するにあたっては、他にないオリジナリティ溢れた要素を取り入れて製作することで、FPGA への理解を深め、技術習得をすることを目的として研究を進めた。

2 研究内容

本研究では、FPGA と VHDL 言語を利用して、モニター画面に出力される迷路ゲーム器の製作をした。リセットするごとに乱数によって新しい迷路パターンが生成され、さらにキャラクターの目線に立ってゲームが出来る機能を実装した。図 1 は今回使用した FPGA ボードで、どちらも Xilinx Spartan-3E やスイッチ、LED、画像出力用の VGA 端子があり、研究を進める中で容量が多い右のものに移行した。



図 1 2つのFPGA ボード

FPGA とは、ユーザーがハードウェア記述言語を使って動作をプログラムできる論理回路の集まり（集積回路）である。

VHDL とは、そのハードウェア記述言語の 1 つであり、それを用いてスイッチの入力、迷路の画像の描画、キャラクターの移動等をプログラムした。

```
startmenu#maze_startmenu.xise - [maze_startmenu.vhd]
Window Layout Help
[78] begin
[79]   if clk25'event and clk25='1' then
[80]     if (counter<24999999) then --counter
[81]       counter <= counter+1;
[82]     else
[83]       counter <= "000000000000000000000000";
[84]     end if;
[85]     if h_counter(4 downto 0)=0 then --character
[86]       shift_reg1<=cgl(CONV_INTEGER(cg_adr));
[87]     else
[88]       shift_reg1<=shift_reg1(30 downto 0) & "0";
[89]     end if;
[90]     if (h_counter<799) then --h,vcounter
[91]       h_counter<=h_counter+1;
[92]     else
[93]       h_counter<="0000000000";
[94]       if (v_counter<520) then
[95]         v_counter<=v_counter+1;
[96]       else
[97]         v_counter<="0000000000";
[98]       end if;
[99]     end if;
[100]   end if;
[101] end process;
[102]
```

図 2 迷路設計プログラムの一部

3 研究過程

- 4月 VHDL 基礎学習
一年の到達目標設定
- 5月 部品の発注
画面表示の技術習得
- 6月 基本迷路画面の出力
- 7月 キャラクターの作成
ボタンに対応したプログラムの作成
試作版の迷路パターン設計
迷路の自動攻略プログラムの作成
- 8月 新しい迷路設計プログラムの作成改良
迷路の 3D 表示プログラムの作成
ストップウォッチプログラムの作成
- 9月 迷路を文化祭発表用に調整
迷路の自動生成プログラムの作成
キャラクターの移動方法の改良
リセットボタンの実装
ゴール判定実装
スタートメニュー実装
- 10月 ゴール後の再スタート機能実装
操作用コントローラ作成
- 11月 文化祭展示用ポスター作成
- 12月 レポート作成
- 1月 資料の作成と発表

4 研究成果

(1) 迷路の自動生成

製作当初は迷路のパターンをデータとして与えていたが、スタートメニューを表示した時には迷路が自動的に生成されるようにした。迷路をクリアした後や、途中で迷路脱出を断念しリセットボタンを押した場合でも、違う迷路が生成される。

自動生成の方法は次のとおりで、まず、すべてのますの四方を壁に初期化し、壁を掘り進めるように道を生成していく。掘り進める方向は乱数を用いて決定されるため、二度として同じ迷路が生成されることはない。また、迷路の自動生成においてループや使われない領域が出ないように条件を設定し、スタートからゴールへの最短経路が 1 通りだけ存在する迷路を生成するように工夫した。

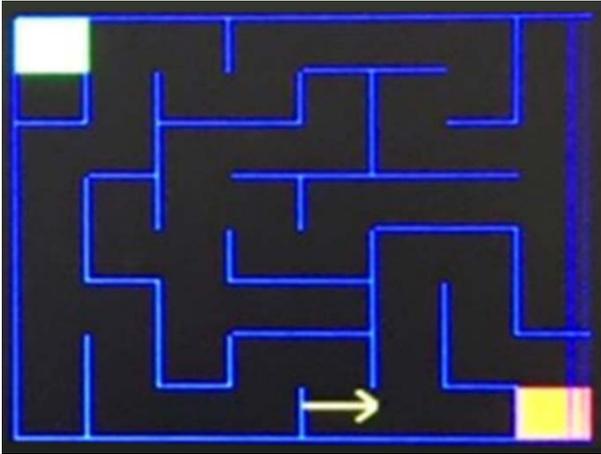


図3 自動生成された2Dの迷路画面例

(2) 迷路の自動攻略

設定されたボタンをONにすることで、自動で迷路を攻略するようにした。自動生成された迷路をコピーし、行き止まりを塗りつぶすことで、スタートからゴールまでの最短経路を生成している。その最短経路を元にキャラクタを動かし、ほんの数秒でクリアする。

(3) 2Dと3D両方の同時画面表示

画面上に2Dと3Dで両方の迷路が表示されるようにした。3Dでは、キャラクタの目線に立っているかのような一人称視点で表示される。キャラクタがいる周りのマス情報を元に、立体的に見えるように線を描画している。右上には、迷路とキャラクタを上から見ているような俯瞰視点での迷路が表示されており、3D・2Dどちらのキャラクタも同期して動くようになっている。2Dでの迷路表示は、キャラクタが通った道のみを表示することもできるようにし、ゲーム性を高めた。

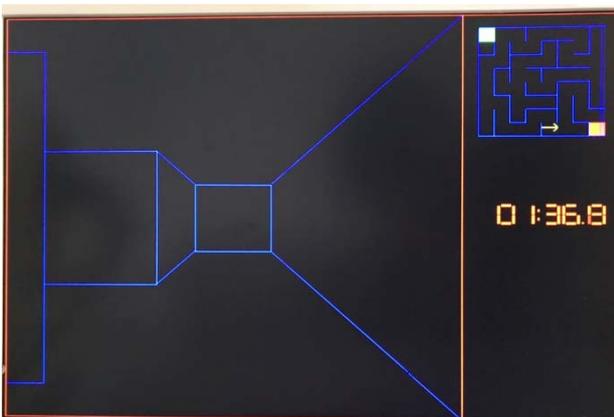


図4 画面表示された迷路例

(4) コントローラでの操作

市販のモニター台を改造して押しボタンを取り付けて、FPGAで表示された迷路のスタート、リセット、そしてキャラクタの操作をできるように

した。ボタンに使用するリミットスイッチにリード線をはんだ付けし、ボードに付いている拡張端子のピンヘッダーと接続することで、FPGAにボタンが押されたことを知らせることができる。

また、電子ブザーを利用した効果音発生回路を製作し、これもFPGAボードの拡張端子に接続し、ボタンを押したときの反応音が出るようにした。



図5 コントローラ

5 考察

今回の製作では、当初ボタンを押してキャラクタを迷路上で動かすとき、少しの押し込みでも2回同じ行動を繰り返してしまうチャタリングという現象が起きていた。その対処として、自己保持回路を用いて、ボタンの1度の押し込みに対して1度しか反応しないようにすることができた。

しかし、まだ改良したい点もある。迷路の途中にカギなどのアイテムを配置し、カギを拾うことでゴールできるようにするなど、さらに発展的な機能を実装することでゲーム性を高めたい。

6 まとめ

(1) 成果

今回の研究では迷路ゲーム器の製作を通じてFPGAやVHDLへの理解を深め、技術を習得することができた。

3Dでの迷路表示と2Dでのミニマップ表示を行うことで、臨場感のあるゲームにすることができた。また、迷路の自動生成と自動攻略の実装に成功した。コントローラを製作することで、加工技術高めるとともにゲームにおける操作性を高めることができた。

(2) 課題

チャタリングを解消したことでボタンの長押しによる連続移動機能が実装できなくなってしまった。ゲーム性を高めることを目標としていたので機会があれば改善したい。