1) Popis

Aplikace na FITkit ovládající Led display o velikosti 8x8 LED diod. Aplikace reprezentuje Mini-Tetris. Skóre dosazene při hrani se zobrazuje na LCD displej FITkitu. Jednotlivé kostky jsou reprezentovány 4 ledkami o různých tvarech. Je možnost tvary pohybovat dolů, do prava a do leva. Dále je možno útvary otáčet. Směr otáčení byl zvolen po směru hodinových ručiček, nicméně jsou implementovány oba směry, později popíši proč, tudíž je možné otáčet kostky v obou směrech, ale pouze jeden je použit přímo v aplikaci. Pro připojení FITkitu a LED displaje bylo zboleno nepájívé pole.

1.2) Ovládnání

Ovládání pohybu kostek po hracím poli je realizováno klávesnicí na FITkitu. Tlačítkem 4 lze kostku posouvat doleva, tlačítkem 6 lze posouvat kostku do prava. Tlačítkem 2 potom dolů. Realizoval jsem pohyb dolu manuálně kvůli rozměrům maticového displeje. Měl jsem posun dolů, nicméně po zaplnění hracího polo, je docela těžké hrát, tudíž jsem zvolil pro hráče jednodušší variantu a to manuální posun dolů.

Tlačítkem 5 poté lze měnit orientaci kostky. Používám obě otočení, jak protisměru hodinovch ručiček, tak po směru. Využil jsem na toto vzorečky, které jsem našel na internetu. Nicméně pro otáčení používám pouze otáčení po směru hodinových ručiček a po otečení, pokud zjistím že by objekt byl vykreslen mimo hrací pole použiji druhé otočení a navrátím jej do původní polohy. Zde by šlo točit znovu na opačnou stranu, čímž bych docílil otočení na druhou stranu, ale tohle chování by mohlo hráče mást, tudíž jsem tuto možnost neimplementoval. Ale znamenalo by to napsání pár řádků do kodu, možnost zde je, ale je nevyužita.

2) Schéma zapojení

Použil jsem JP9 a na něm jsem použil P6M pro ovládání aktivity jednotlivých řádků. Pro zobrazení prvků na jednotlivých řádcích jsem použil polovinu P2M a zbylé 4 bity jsem doplnil P4M.

Na maticovém displeji jsem se nechal inspirovat schématem z INP a to tak že jsem zapojil drátky na nepájivém poli 1 zapojený 2 volné, 2 zapojené 2 volné.. 1 zapojený. Vyvedl jsem takhle drátky z každé strany pro větší přehlednost. A poté jsem zača zapojovat fitkit k tomuto poli. A to způsovem že na každé straně se nachází čtvreřice pinů ovládající řádek a čtveřice pinů ovládající sloupce.

Tudíž P6 ovládá vběr řádků a spodní 4bity proměnné ovládají pomocí P4 polovinu prvků na jednotlivých řádcích a horní 4bity proměnné ovládají pomocí P2 zbytek displeje.

3) Implementace

Pro implementaci jednotlivých tvarů jsem použil dvou rozměrné pole o velikosti 4x4 kam jsem zapisoval buď 1 nebo 0. Z čehož vznikli jednotlivé tvary. Pro hrací plochu jsem poté použil dvou rozměrné pole o velikosti 8x8 kam jsem postupně zadával jednotlivé tvary a pokud již nejde jít s útvarem níže, generuji nový a kontroluji, zda se nevyskytl nový řádek.

Pokud kostka již nemůže jít níže kontroluji zaplněnost řádků, pokud takový řádek naleznu, smažu jej a řádky o jedna vyšší posunu o tolik řádků, kolik jsem smazal níž.

Pro zobrazení používám funkci, která postupně vezme čísla z matice, počítám s tím že tam jsou uložené jedničky a nuly, což si lze představit jako číslo ve dvojkové soustavě, tudíž toto číslo převedu do desítkové a poté vypíši na P4OUT a zároveň na P6OUT. Jednotlivé bity se poté postarají o korektní zobrazení na dipleji. Pro rotování řádků používám switch, zde by se dala použít bitová rotace, nicméně jsem již neměl čas a prostředky pro takovou implementaci.

Rotaci jsem již popsal v sekci 1.2 nicméně čerpal jsem z wikipedie [1] sekce „Square matrices“. Mohl jsem použít pro každé natočení speciální matici, nicméně to mi nepřijde jako

dostatečně dobrý způsob implementace.

Pro práci s klávesnicí a LCD displejem FITkitu jsem použil funkce, které jsem získal z demo ukázky v SVN repozitáři FITkitu.

Vykreslení jednotlivých tvarů na displej je realizován, tak že se pokusím posunout daný tvar ve směru, který hráč chce. Pokud vykreslím méně jak 4 prvky z dané matice (prvek je nenulová hodnota v matici tvaru), posunu útvar zpět na jeho pozici, ze které vycházel. Tudíž, pokud stisknu 4 a do výsledné maice lze uložit jenom 2 prvky, kvůli tomu že je zde již njaký prvek neposouvám útvar v tomto směru.

Pokud lze vykreslit všechny 4 části daného obrazce uložím tento obrazec do výsledné matice. Tudíž výsledná matice obsahuje také 1 a 0, kde každá 1 znázorňuje že se na daném indexu nachází prvek z nějakého tvaru. Po položení prvku do výsledné matice, dále kontroluji, zda se nenachází v této matici plný řádek. Pokud ano, smažu jej a posunu všechny prvky o jedna dolů.

Takže výsledek uchovávám v matici 8x8 kam postupně ukládám prvky, při posunu se pokouším překrývat tuto matici s menší maticí a pokud nelze uložit daný prvek do této matice neposunu tento prvek ve výsledné matici a až v momentě, kdy prvek posune hráč dolů, kde zapadne buď na dno hrací plochy, nebo na předešlé prvky, uložím tento nově vygenerovaný prvek do výsledné matice.

4) Závěr – známé chyby

Při testování jsem si všiml že občas pro vygenerování nového prvku musí užvatel opětovně mačkat tlačítko dolů. Nepřišel jsem na důvod tohotot bugu. Dále je nižší kontrast jednotlivých řádků, opět toto bude jistě spůsobené buď nedostatečným zapojením, nebo nedostatečným napájením z FITkitu.

Nicméně jsem si psaní tohoto projektu náramně užil a přinesl mi spoustu zkušeností.

[1] In-place matrix transposition. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-12-16]. Dostupné z: http://en.wikipedia.org/wiki/In-place_matrix_transposition

Obrázek zapojení: