Team Alpha

James Houghton, William Zhang, Jacob Blindenbach, Guy Verrier

Final Project - Overview

- Part 1 - Cube Game
  - Completed project requirements
- Part 2 - Expansion of Cube Game
  - PowerUps
    i. Life
    ii. Crosshair Size & Speed
    iii. Cube Freezing
  - Highscore System
    i. Non-volatile storage
  - Bitmap Graphics
- Lessons Learned

Motivation

- Improve gameplay dynamics
  - Implement powerups that change player attributes (crosshair size, crosshair speed, lives)
  - Implement powerups that change gameplay attributes (sprite movement)
  - Add sprites to create more interesting visuals
- Add persistence and progression
  - Keep track of high scores
  - Store history of high scores in ROM

Implementation: Powerups

- Each cube may have single powerup or no powerup
- Extended intersection-check function to run powerup effects
- Implementation for each powerup is different

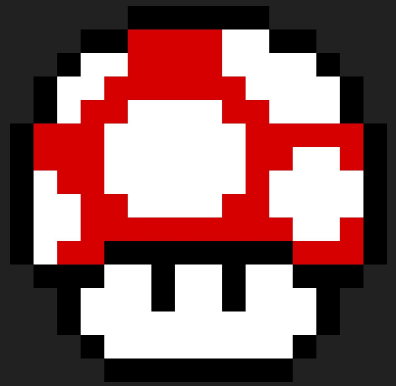
```
enum PowerUp {
  NONE = 0,
  LIFE,
  XHAIR,
  SPEED,
  FREEZE,
  SLOW
};
```

Implementation: Powerups: Life

- Easiest powerup
- When activated, Life is incremented by one
  - Read-modify-write → Life must be guarded by a lock

Implementation: Powerups: Crosshair Size

- Temporary trait: increase size for 5 seconds
  - More difficult than Life powerup
  - Must change BSP_LCD_DrawCrosshair
- When activated, spawn thread
  - Increases crosshair size
  - Sleeps
  - Resets crosshair size
- Problems with triggering it multiple times
  - Need to have a counter

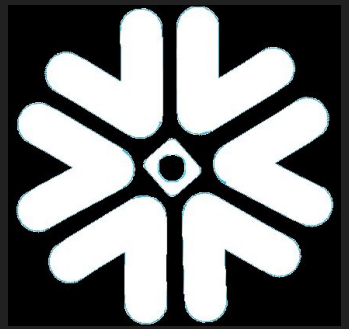


Implementation: Powerups: Crosshair Speedup

- Temporary trait: double speed for two seconds
  - Similar to previous powerup
- Similar thread/ID usage as Size powerup
- Doubled velocity used in Producer

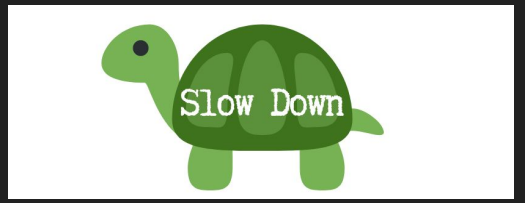
Implementation: Powerups: Cube Freezing

- Temporary trait: freezes cubes for 2 seconds
- Similar thread/ID usage as Size powerup
- Various ways to approach this
  - Modified direction-checking function
  - When powerup is active, always return that no directions are available



Implementation: Powerups: Cube Slow Down

- Temporary trait: Slows down crosshair for 2 seconds
  - Interacts with crosshair speedup
- Similar thread/ID usage as Speed powerup
- Life does not decrement



Implementation: Highscore and Leaderboard

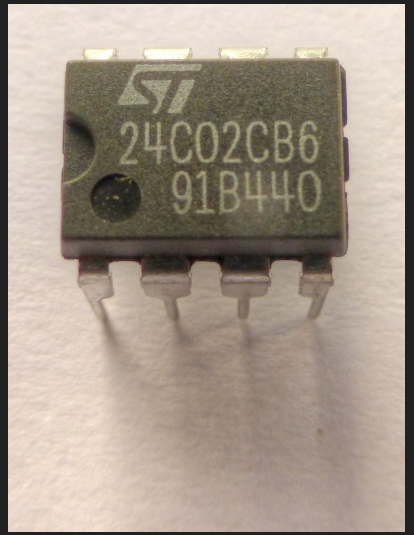
- Arcade-like
  - Enter three-letter name with score
  - Use joystick to set name
- Top 4 scores saved
  - Stored in non-volatile memory

```
struct HighScore {
    char letters[4];
    int score;
};

#define NUM_HIGHSCORES 4

struct HighScore highscores[NUM_HIGHSCORES];
```

- Challenges:
  - Changing Producer to use joystick as a char selector
    - Must clear FIFO and change its size
  - Displaying everything correctly
    - DrawChar size parameter → 2

Implementation: Saving High Score While Off

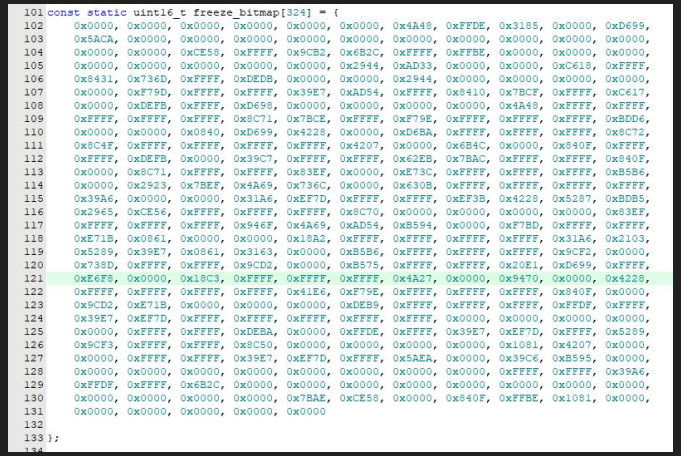
- Restore high score when power is lost
- Store high score data structure in EEPROM
- Magic Bit!
- Challenges
  - Keil / Compiling Tiva Drivers
  - Casting / Zero bug




```
EEPROMRead(arr, 0x0, 40);
if (arr[0] != MAGICBIT){
  for (i = 0; i < NUM_HIGHSCORES; ++i) {
    highscores[i].score = -1;
  }
} else{
  memcpy(highscores, arr + 1, sizeof(struct HighScore) * NUM_HIGHSCORES);
}
```

Implementation: Bitmap Graphics

- C byte array
  - Length of 324 (18x18) with uint16_t values
  - Make background of desired icon black
  - Scale down to 18x18
  - Flip red and blue color channels
  - Export as 24-bit bitmap
  - Run BmpConvert16.exe to generate C byte array
  - Add byte array to bitmap.h
- Modified DrawCubes()
  - Replaced BSP_LCD_FillRect with BSP_LCD_DrawBitmap
  - Based on type of powerup, draw corresponding Bitmap from bitmap.h
  - Adjusted y coordinate by adding block height and subtracting 1



```
const static uint16_t freeze_bitmap[324] = {
    0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4A48, 0xFFDE, 0x3185, 0x0000, 0xD699,
    0x5ACA, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
    0x0000, 0x0000, 0xCE58, 0xFFFF, 0x9CB2, 0x6B2C, 0xFFFF, 0xFFBE, 0x0000, 0x0000, 0x0000,
    0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2944, 0xAD33, 0x0000, 0x0000, 0xC618, 0xFFFF,
    0x8431, 0x736D, 0xFFFF, 0xDEDB, 0x0000, 0x0000, 0x2944, 0x0000, 0x0000, 0x0000, 0x0000,
    0x0000, 0xF79D, 0xFFFF, 0xFFFF, 0x39E7, 0xAD54, 0xFFFF, 0x8410, 0x7BCF, 0xFFFF, 0xC617,
    0x0000, 0xDEFB, 0xFFFF, 0xD698, 0x0000, 0x0000, 0x0000, 0x0000, 0x4A48, 0xFFFF, 0xFFFF,
    0xFFFF, 0xFFFF, 0xFFFF, 0x8C71, 0x7BCE, 0xFFFF, 0xF79E, 0xFFFF, 0xFFFF, 0xFFFF, 0xBDD6,
    0x0000, 0x0000, 0x0840, 0xD699, 0x4228, 0x0000, 0xD6BA, 0xFFFF, 0xFFFF, 0xFFFF, 0x8C72,
    0x8C4F, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x4207, 0x0000, 0x6B4C, 0x0000, 0x840F, 0xFFFF,
    0xFFFF, 0xDEFB, 0x0000, 0x39C7, 0xFFFF, 0xFFFF, 0x62EB, 0x7BAC, 0xFFFF, 0xFFFF, 0x840F,
    0x0000, 0x8C71, 0xFFFF, 0xFFFF, 0x83EF, 0x0000, 0xE73C, 0xFFFF, 0xFFFF, 0xFFFF, 0xB5B6,
    0x0000, 0x2923, 0x7BEF, 0x4A69, 0x736C, 0x0000, 0x630B, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
    0x39A6, 0x0000, 0x0000, 0x31A6, 0xEF7D, 0xFFFF, 0xFFFF, 0xEF3B, 0x4228, 0x5287, 0xBDB5,
    0x2965, 0xCE56, 0xFFFF, 0xFFFF, 0xFFFF, 0x8C70, 0x0000, 0x0000, 0x0000, 0x0000, 0x83EF,
    0xFFFF, 0xFFFF, 0xFFFF, 0x946F, 0x4A69, 0xAD54, 0xB594, 0x0000, 0xF7BD, 0xFFFF, 0xFFFF,
    0xE71B, 0x0861, 0x0000, 0x0000, 0x18A2, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x31A6, 0x2103,
    0x5285, 0x39E7, 0x0861, 0x3163, 0x0000, 0xB5B6, 0xFFFF, 0xFFFF, 0xFFFF, 0x9CF2, 0x0000,
    0x738D, 0xFFFF, 0xFFFF, 0x9CD2, 0x0000, 0xB575, 0xFFFF, 0xFFFF, 0x20E1, 0xD699, 0xFFFF,
    0xE6F8, 0x0000, 0x18C3, 0xFFFF, 0xFFFF, 0xFFFF, 0x4A27, 0x0000, 0x9470, 0x0000, 0x4228,
    0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x41E6, 0xF79E, 0xFFFF, 0xFFFF, 0xFFFF, 0x840F, 0x0000,
    0x9CD2, 0xE71B, 0x0000, 0x0000, 0x0000, 0xDEB9, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFDF, 0xFFFF,
    0x39E7, 0xEF7D, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000,
    0x0000, 0xFFFF, 0xFFFF, 0xDEBA, 0x0000, 0xFFDE, 0xFFFF, 0x39E7, 0xEF7D, 0xFFFF, 0x5289,
    0x9CF3, 0xFFFF, 0xFFFF, 0x8C50, 0x0000, 0x0000, 0x0000, 0x0000, 0x1081, 0x4207, 0x0000,
    0x0000, 0xFFFF, 0xFFFF, 0x39E7, 0xEF7D, 0xFFFF, 0x5AEA, 0x0000, 0x39C6, 0xB595, 0x0000,
    0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0x39A6,
    0xFFDF, 0xFFFF, 0x6B2C, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
    0x0000, 0x0000, 0x0000, 0x0000, 0x7BAE, 0xCE58, 0x0000, 0x840F, 0xFFBE, 0x1081, 0x0000,
    0x0000, 0x0000, 0x0000, 0x0000, 0x0000

};
```

Responsibilities

- Jacob Blindenbach
  - Wrote part 1 report, took video of part 1 demo
  - Coded parts 1 and 2, freeze, slow down, power-up reset
  - Persistent leaderboard storage
  - Wrote corresponding parts in final report
- James Houghton
  - Structured and wrote most of part 1 code
  - Structured and wrote most of the part 2 code:
    - Improvements: power-ups, highscore name-entering, and the leaderboard
  - Wrote the portions of the report detailing new features
- Guy “Jack” Verrier
  - Created slides for the slide deck
  - Wrote Introduction and Conclusion of report
  - Wrote random number generator
- William Zhang
  - Implemented LFSR pseudo-random number generation
  - Improved and optimized crosshair behavior
  - Implemented Bitmap icons for powerups and default “cubes”

Results

- Fully-functional arcade-like game on the TM4C!
  - Functioning sprites (bitmaps)
  - High scores (persistent on ROM!)
  - Fun and interesting powerups
- Bug-free!
  - Even when restarting the game many times

Demo

Insights & Lessons Learned

- Simplicity is crucial
  - Especially with multi-threaded systems
  - Cumbersome debugging tools
- Scope your ambitions to your schedule

Q&A

Final Project Presentations

- Each group will give a ~20 min talk to present their final project.
- All members of the team should participate in the presentation.
- The presentation should include a description of:
  - (i) the idea/feature of the project
  - (ii) design and implementation methods
  - (iii) team responsibilities (who did what?)
  - (iv) major results by showing a demo of the system on the board or analysis results in case of OS features
  - (v) insights and lessons learnt.
- Each presentation will follow with 5 min Q&A.
- The presentations will be evaluated by both me and TA and other students/teams based on a given rubric posted on Collab.

- **Top 3 teams based on evaluations will receive up to 5 bonus points towards their final grade.**