CAB202 Assignment 2: Teensy in Space

Assignment 2 will be marked during your assigned tutorial session in Week 13 (30/05 to 3/06) using the files submitted to AMS by 29/05
Marks: 40 (40% of your final mark)

For this assignment, you will be writing your own version of the arcade classic, Space Invaders.

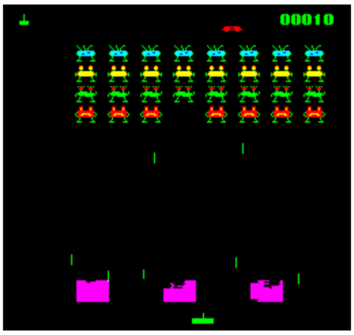

Figure 1. Game initial concept

Requirements

Your task is to implement a game in which several aliens will appear on the top of the LCD display moving sideways. The player controls a sprite at the bottom of the display, moving it left and right with the goal being to maximise the score before losing all lives. Points are gained when the player's bullets hit aliens. Lives are lost when aliens' bullets hit the player, as described in detail below. You will use the skills and techniques learned in the first half of the semester, together with microcontroller programming techniques covered since week 7 to complete the assignment.

WARNING: QUT takes plagiarism very seriously. If your assignment has material which has been copied from other class members, previous assignments, or code from any other source you will be penalised severely. If you cannot explain exactly what your code is doing, it is clearly not your own work, and you should not be submitting it as such!

Game Overview

The basic aim of the game is for the player to use buttons to control the teensy and move a small sprite around the screen until the player achieves 15 points. The sprite may be a simple character, such as the letter 'H', or something more imaginative. Aliens appear at the top of the screen moving sideways back and forth while dropping bullets. The user has no control over the aliens' motion. The player starts with 3 lives. The aim of the game is to exterminate the aliens before losing all lives. The game ends when the player has no remaining lives, or when a score of 15 points is achieved.

Level 1: Basic functionality [20 marks total]

All versions require the following basic functionality (*a reasonable attempt to complete all aspects of this level is required to qualify for marks on advanced levels*):

1. Screen setup: a score, remaining lives and game level appear at the top of the screen. **[2 marks]**
2. Aliens setup: 15 aliens appear on the screen below the score box. Aliens should be designated by a visually distinctive alien-like icon. That is, they should not be a simple geometric shape such as a square or circle. They should move in formation from left to right and back again, never leaving the screen. Hints: you can display the aliens in 3 rows of 5 aliens. **[2 marks]**
3. Any alien can drop bullets, which may pass freely over other aliens, so that collision detection between aliens and their own bullets is not required. **[2 marks]**
4. Player shoots bullets by pressing the 5-way push button upwards (located on the left of the teensy). More than one bullet on screen at once is allowed. **[3 marks]**
5. Bullet's size must be 1 pixel. **[1 marks]**
6. Pressing the right button at the start of play changes game level. **[2 marks]**
7. Motion of the player is controlled by the left and right buttons (located on the right side of the teensy). **[2 marks]**
8. When an alien is hit by a player's bullet, the alien and the bullet disappear and the score is updated (1 point). **[2 marks]**
9. The player loses a life when hit by an aliens' bullet and the "Lives remaining" score is updated. **[2 marks]**
10. "Game over" appears on the screen when game finishes. This occurs when a score of 15 points is reached or the player loses all lives, whichever comes first. **[2 marks]**

Level 2: Descending Motion and Obstacles [10 marks total]

In this mode, a more elaborated version of the game is played with the player's motion controlled by the potentiometers and the motion of the aliens showing a different behaviour. Same specifications as for level 1 apply, with the following changes:

1. The motion of the player is controlled by the top right potentiometer. The motion of the player should be smooth, and allow the player to move to all possible positions on the bottom of the screen. **[2.5 marks]**
2. Three horizontal walls or platforms appear above the player (see Figure 1). Their positions are fixed, and they are spaced out evenly along the screen width. The gap between walls is at least 6 pixels. **[2.5 marks]**
3. The aliens move sideways back and forth while descending until half of the LCD height, and then back up. Same vertical separation is maintained during the descending motion. **[2.5 marks]**
4. Bullets will chip away the walls. Both aliens' or player's bullets will disintegrate a section of the wall every time they hit it. The section of the wall that disappears per bullet should have be the size of a single pixel. **[2.5 marks]**

Level 3: Independent row motion and curved bullets [10 marks total]

In this mode, aliens will show a different behaviour. Same specification as for level 2 apply with the following changes:

1. The aliens appear in 3 rows of 5. Each row moves independently of the others. **[4 marks]**
2. The movement of the Player's bullets follow a parabolic trajectory. **[3 marks]**
3. Curvature of the bullet path is controlled by the top left potentiometer. The bullet should be able to curve both left and right. **[3 marks]**

   Note: to make the game easier to play, you are allowed to use the 5-way push button to move the spacecraft laterally and shoot bullets at the same time (pressing upwards), while the potentiometer adjusts bullet curvature.

Marking

The assignment will be out of 40 marks and will be worth 40% of your total grade in this subject. The following should be noted about marking:

- **If your code does not compile, you will get 0 marks for the entire assignment.**
- If the game crashes or locks up during testing, you will receive marks for what has been tested up that point. No more of your assignment will be marked. We will not debug your code to make it compile or run.
- Your game must be easily playable. If timings, settings, or controls are set in a manner that makes it difficult to play (e.g. not using the key inputs specified, ridiculously fast movement, etc.), you **will receive 0 for the assignment**.
- The assignment will be marked during your scheduled tutorial in week 13. To receive a mark for the assignment, you are required to:
  - attend the tutorial in person,
  - bring your student ID card,
  - demonstrate your program to a tutor,
  - explain details of your implementation, and
  - hand in your Teensy.

If you fail to do any of these things you will receive a mark of 0 for the assignment.

Submission

Your assignment is due on Sunday May 29th, 2015 at 11:59pm. Submission will be online through the AMS used in the tutorials. You must submit through the submission page (available at http://bio.mquter.qut.edu.au/CAB202/Exercise?TopicNumber=6&ProblemNumber=2), and follow all of the instructions.

When submitting to the AMS, it is your responsibility to make sure that your assignment compiled correctly. In general, AMS will compile your code and return any errors that occur. There are circumstances when very poor code causes the compiler on the server to terminate abnormally, or to hang. *You are responsible for ensuring that this does not happen* by submitting only valid source code which you have already compiled using the same settings as those used by AMS. As mentioned above, if you do not submit a compiled assignment, you will receive 0 marks. You will have 50 submission attempts for the assignment, so there are no excuses for not resolving any compilation issues.

AMS operates in unattended mode outside regular Monday to Friday office hours. Please note that issues such as software failure, hardware failure, and network congestion problems DO NOT constitute exceptional circumstances for the purpose of Special Consideration. It is in your best interest to complete this assignment early to eliminate the possibility of a last-minute disaster. To ensure success you should aim to submit a substantially complete version of your assignment by the beginning of Week 12.