

CAB202 Assignment 2, Semester 2 2019

Assessment Weight: 40%

Context

You are tasked with designing a Tom and Jerry themed game for the teensy pewpew platform. Jerry, who will be controlled by the user, just wants to eat cheese and dodge Tom. After he has had a few pieces of cheese, he likes to have a bit of fun by setting off fireworks in pursuit of Tom. Tom, ever the strategist, moves randomly around the room, dropping mousetraps periodically in hopes of catching Jerry. After becoming increasingly frustrated at Jerry for thwarting his plans, when the characters advance to the second room, Tom decides to get smart and mix a chemical concoction into some bowls of milk to leave for Jerry. However, this has some unexpected consequences, with the milk giving Jerry temporary superpowers! For a short burst of time after consuming the milk, super-Jerry can not only walk through walls, but is also invincible to Tom and his traps. The characters roam through increasingly variable environments, with moving walls to dodge in addition to the other game elements. These behaviours are explained further in the task breakdown below.

Level 1 Specifications

Task 1 – Basic Behaviours (14 marks)

a) Basic Game Requirement - Debouncing

You will need to debounce all switches on the teensy for playability. If a behaviour cannot be easily demonstrated because a switch is not debounced, no marks will be awarded for that criteria.

b) General Game Setup

Your game should begin with a start screen, which displays your name, student number, and the name of your game (you may choose this, but it must be appropriate). The game should remain on this start screen until the right switch is pressed. Once the right button press has been detected, the initial game screen should be drawn. The characters and walls should be drawn in their initial positions, as described further in Task 1c. There should also be a status bar at the top of the screen, above the gameplay area, which should contain **in order** the following information:

- i. The current level (initially 1),
- ii. Jerry's remaining lives (initially 5),
- iii. The score (initially 0),
- iv. The game time, formatted with leading zeros, displayed as MM:SS (initially 00:00).

You will also need to draw a horizontal line across the entire screen under this information, separating the gameplay area from the status bar.

c) Character and Wall Setup

The initial positions of the characters and walls in level 1 can be manually programmed. The characters should be denoted by bitmaps with width in the range of 3-5 pixels and height in the range of 3-7 pixels (you may choose the bitmap, as long as the characters can be easily identified). The initial game state is given by the following absolute coordinates:

- i. Tom should be drawn at (LCD_X - 5, LCD_Y - 9),
- ii. Jerry should be drawn at (0, STATUS_BAR_HEIGHT + 1),
- iii. A wall should be drawn from (18, 15) to (13, 25),
- iv. A second wall should be drawn from (25, 35) to (25, 45),
- v. A third wall should be drawn from (45, 10) to (60, 10),
- vi. A fourth wall should be drawn from (58, 25) to (72, 30).

d) Pause Mode

If the user closes the right switch at any time during game play, the game should enter a pause mode. In pause mode, the game time stops, and all timer-based or automated game elements will cease to update. Jerry can still be moved manually, and should be able to interact with the cheese, mousetraps, Tom and walls as normal, but no new cheese, mousetraps or milk should appear, and Tom and the walls should not move. The game can be unpaused by pressing the right switch again.

e) Character Movement

Jerry's movement should be controlled by the joystick up, left, down and right switches (with Jerry moving one pixel in the direction of joystick movement per switch press). Tom's movement should be automated and random, such that each time Tom collides with a wall or the edges of the screen, he proceeds to move at a new random speed and direction. The random movement parameters should be different each time the game is played. When Jerry is moving at full speed (a switch is held closed), Jerry can outrun Tom. No part of the characters can ever leave the screen, nor can any part of the characters move through walls (except the behaviours described in Task 3).

f) Collision Detection

You will need to implement a collision detection algorithm to detect interactions between game elements. Tom and Jerry must be entirely visible on screen at all times, and no part of either character can move into walls or the status bar. If Tom catches Jerry (any part of the characters' bitmaps overlap), Jerry loses a life and both characters are reset to their initial positions. You must also have collision detection between all other game elements, as described in the relevant task descriptions.

g) Cheese Appearance and Interaction

One piece of cheese is spawned every 2 seconds, up to a maximum of 5 in the room at once. Cheese can appear in any unoccupied area within the room, and there should be no noticeable pattern in the cheese spawning locations. Jerry consumes a piece of cheese if any part of his bitmap overlaps any part of the cheese bitmap. Each time Jerry eats a piece of cheese, another piece of cheese appears 2 seconds after it has been consumed. If he consumes multiple very quickly, spawn one piece 2 seconds after the first piece was eaten, and each successive piece at 2 second intervals afterwards. Each piece of cheese Jerry consumes increases his score by 1. Tom has no interaction with cheese. Cheese does not move, so the number of cheese on screen should be reset when the characters move to the next room. Further cheese interactions are detailed in Tasks 1i and 1j.

h) Mousetrap Appearance and Interaction

Tom drops mousetraps at his current location as he moves around the room (and hence can never spawn overlapping existing game elements). These mousetraps appear at 3 second intervals, up to a maximum of 5 traps in the room at one time. Jerry gets caught by a trap if any part of his bitmap overlaps any part of the mousetrap bitmap. When Jerry steps on one of Tom's traps, he loses 1 life, but may continue on his path. Mousetraps become inactive and disappear from view after Jerry has stepped on them, and 3 seconds later another mousetrap is spawned at Tom's current location. Tom has no interaction with mousetraps after spawning.

i) Door Appearance and Interaction

Once Jerry has eaten 5 pieces of cheese in the current room, the door to the next level appears. You may choose the door bitmap, but it needs to be distinct and of a sensible size. The door appears at a random, unoccupied location on the screen, and cannot spawn over the current location of any other existing game elements. Cheese continues to spawn after the door spawns, and Jerry does not have to go through the door immediately. Once any part of Jerry's bitmap overlaps any part of the door bitmap, the characters are both transported to

the second room. See level 2 specifications for further game functionality. If level 2 is not implemented, or the player moves through the door in level 2, this should bring up a game over screen, with the option to restart by pressing the right button.

j) Jerry's Weapons: Fireworks

Once Jerry has eaten 3 pieces of cheese, he gains the ability to use fireworks as weaponry against Tom, and retains this ability until game over or reset. Jerry's fireworks should be a single pixel, and should continually update their direction to follow Tom. If fireworks hit any walls or the edges of the game area before reaching Tom, they disappear. Jerry has 20 fireworks at his disposal, and gains another firework each time one disappears from the screen. One firework is fired each time the joystick centre switch is closed. Fireworks cannot follow the characters between rooms.

Task 2 – Further Movement and Control (6 marks)

a) Wall Movement

The rooms offer some additional challenges for Tom and Jerry. The walls can move, sometimes closing in on the characters and trapping them, causing them to be reset to their initial location. All the walls move with a velocity directed perpendicular to the wall itself, and a speed slower than either of the characters. If the walls reach the boundaries of the screen, they should wrap around and reappear on the other side of the screen at the appropriate location to continue their trajectory. The walls have no interaction with cheese, mousetraps or milk other than the spawning requirements detailed in the relevant Task 1 specifications. See Appendix element (a) for a visual representation of the wall movement.

b) ADC Speed Control

To add an extra level of control to the game, the potentiometers on the pewpew are used to scale the playing speeds. The right potentiometer is used to scale the velocity of the walls, with the range [-MAX, MAX]. This will be such that the negative of a given scaling factor should invert the direction of movement (the wall will move in the opposite direction). The left potentiometer should scale the movement speed of the characters, with the range [0, MAX]. In the instance of both the characters and the walls, the maximum speeds must be appropriate for playability. This should satisfy the following:

- i. Wall Speed < Tom Speed < Jerry Speed,
- ii. Tom's and Jerry's speeds must remain proportional as they increase and decrease.

Level 2 Specifications

Task 3 – New Behaviours and PWM (8 marks)

a) Level 2 Requirement – Level Change Button

To qualify for level 2 marks, you **must** implement the ability to change levels by closing the left switch. If this is not implemented, you will not qualify for any marks for further functionality (i.e. you will not receive any marks for Task 3 or 4 functionality).

b) Powerful Potions

Tom, in his desperation to catch Jerry, has decided to mix chemicals into some bowls of milk to leave for Jerry. However, his methods backfired, with Jerry gaining superpowers for a short period after consuming each bowl of milk. Tom only has one bowl available for milk, and places the bowl of milk at his current location at 5 second intervals (you may design an appropriate bitmap). Once any part of the bitmap for the bowl of milk overlaps any part of Jerry's bitmap, the bowl of milk disappears from the screen, and does not respawn for 5 seconds. Jerry begins a 10 second super-Jerry mode, which involves the following behaviours:

- i. Jerry can move through walls,

- ii. Jerry has no interaction with mousetraps,
- iii. Jerry gains 1 point if he catches Tom (and cannot lose lives),
- iv. Jerry's bitmap increases by 1 size, and
- v. Both LEDs gradually flash from a low to high light intensity in a defined sequence controlled by PWM.

These behaviours can only be exhibited in super-Jerry mode.

Task 4 – Serial Communication (12 marks)

a) Establishing Connection

Upon commencement of level 2, a serial connection should be initialised. If Task 4d is implemented, the game should expect all inputs to be coordinates of game elements until an end of file command is received.

b) External Game Control

The functionalities described in level 1 can also be controlled via serial. This behaviour should be as follows:

- i. The keys 'w', 'a', 's', and 'd' can be used to move Jerry up, left, down and right, respectively,
- ii. The 'p' key toggles pause mode,
- iii. The 'f' key fires a firework (if available), and
- iv. The 'l' key changes the level.

c) Game State Information

Using the established serial connection, the user can request game state information at any time by pressing the 'i' key on the keyboard. The game information should be sent from the teensy via serial to the connected computer, and should include the following:

- i. A timestamp,
- ii. The current level number,
- iii. Jerry's lives,
- iv. The score,
- v. The number of fireworks on screen,
- vi. The number of mousetraps on screen,
- vii. The number of cheese on screen,
- viii. The number of cheese which has been consumed in the current room,
- ix. A true or false value for whether Jerry is in super-mode,
- x. A true or false value for whether the game is paused.

d) Input Parsing

The game elements for level 2 can be initialised by parsing input from a text file. Inputs should continue to be read until the end of file is detected. After the text file has been read, the serial connection should still remain open, and should still accept and respond appropriately to keyboard inputs. The text file will contain coordinates for Tom and Jerry, and coordinates for each wall. You may assume there will never be more than 6 walls. The formatting for each of the encoded elements is presented in the example below:

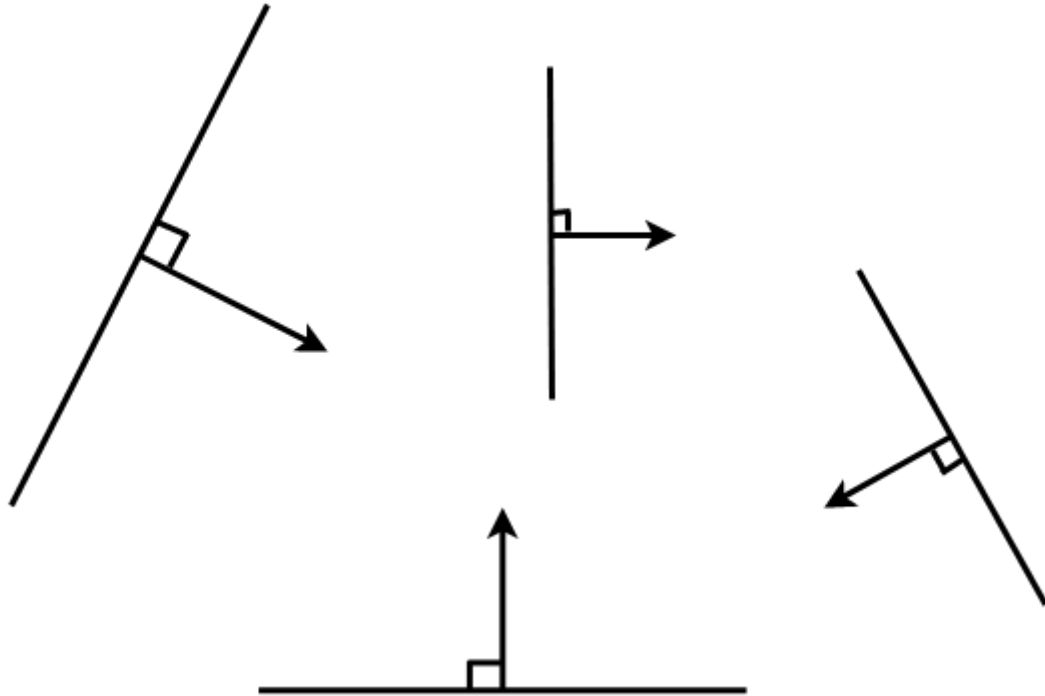
- i. T 79 39
- ii. J 0 10
- iii. W 18 15 13 25

You should create your own text files to test with, using this formatting. These will be absolute coordinates, there is no need to scale to screen size. A maximum status bar height of 10 pixels will be assumed. The length of the walls will never be greater than the largest of those drawn in Level 1.

Appendix

a) Wall Movement Diagram

The movement of the walls can be visually represented by the diagrams below. The arrow pointing perpendicular to the wall represents the direction of movement which should be exhibited by the wall as described in Task 2a.



b) Teensy Control Map

