Computer Organization & Assembly Language

Fall 2021

Final Project

Candy Crushing Game

Final Deadline: Dec 22nd 2021

Instructions:

- You are required to work in MASM615. Project incompatible of MASM615 will not be considered.
- A group of maximum size **two** is allowed.
- Cross Section groups are **not** allowed.
- User Interface is important in this project. Try to develop an attractive user interface.
- Use of extra features in the project is encouraged.
- Use good programming practices (well commented and indented code, meaningful variable names, readable code etc.).
- Only one group of the student submit the project in Zip File.
- Evaluation Criteria will be shared with you soon.
- Copy/cheating case will be awarded an "F" grade in the course.

Game Description

Candy Crush is a **"match-three" game**, where the core game play is based on swapping two adjacent random candies among several on the game board to make a row or column of at least 3 matching-random candies. On this match, the matched random candies are removed from the board, and **random candies above them fall into the empty spaces**, with new random candies appearing from the top of the board. This may create a new matched set of random candies, which is automatically cleared in the same manner. The game is split among many levels, which must be completed in sequence.

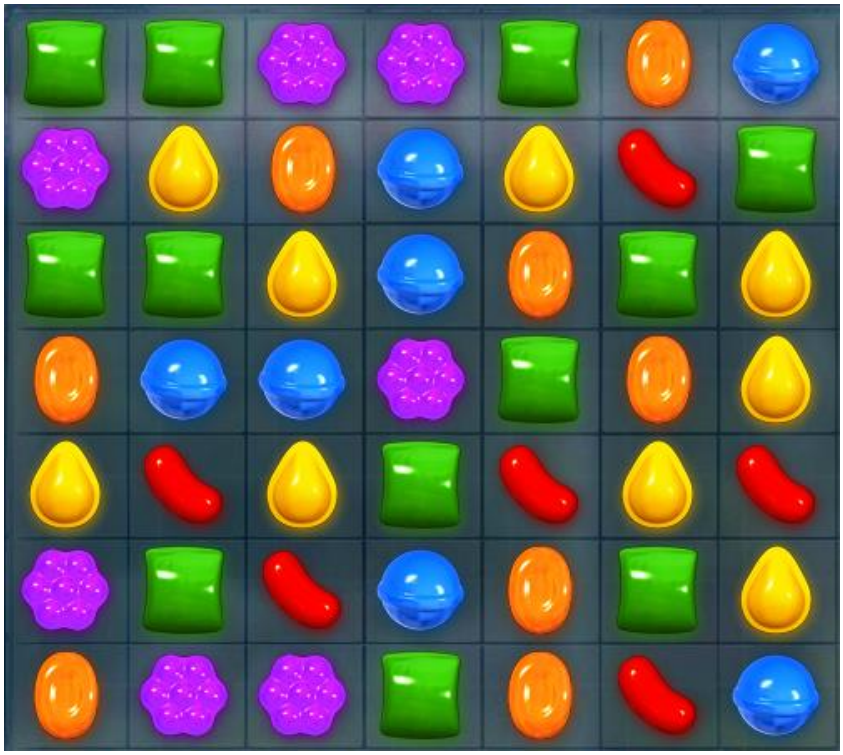
LEVELS

Level 1

First of all, a title page should appear that displays the name of the game. It must also take the name of the user and name is to be display on the screen.

Second page must display the rules of the game, how to play the game.

Then it must have a 7x7 board. When a candy is swapped with another candy, if a combo exists, the combo is crushed, dropped, and score is updated accordingly. Otherwise, the candies are swapped back! The board is filled with random candies of different colors and shapes. It has a color bomb too. When a candy is swapped with bomb, all of its occurrences are destroyed.



When candies are crushed, score is awarded. The score depends on the type of candies that are crushed as each candy has different score. All candies of same shape and color should have same score.

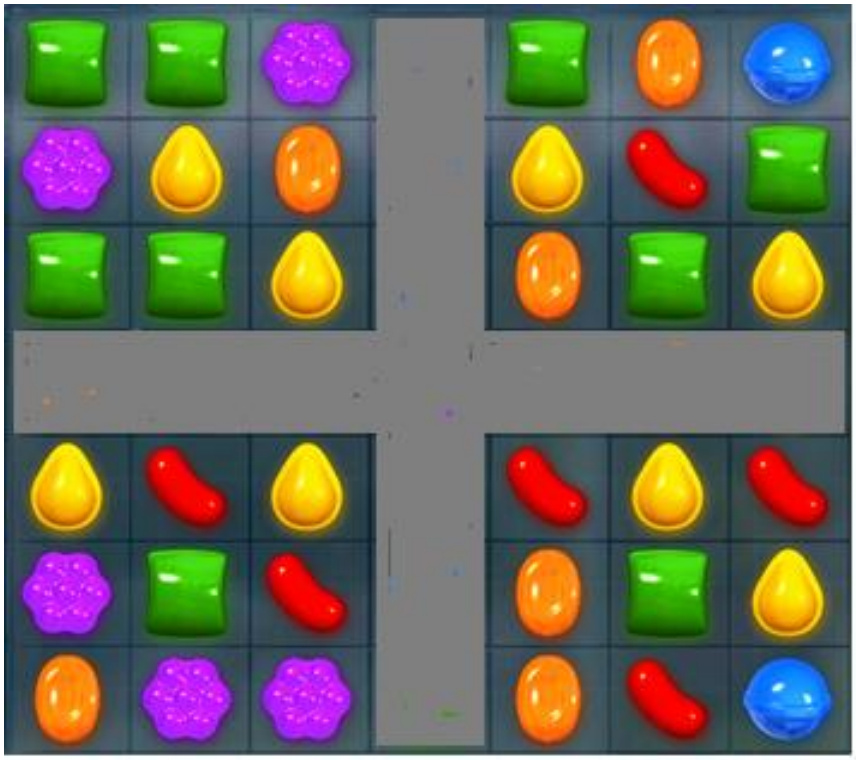
The shapes, colors and score of candies along with background color of the board are left to the students (you can choose yourself).

Level 2

It has a different shape board as shown in the figure below with same functions of level 1. The board is filled with random candies of different shapes and colors.

Level 3

It has same board shape as level 1 but we have blocked row and column for restricting player's movement.



How to Update SCORE AND MOVES

1) During crushing, the score added depends on the size of combo. A combo of 3 adds 3 to the score. A combo of 4 will add 4 and so on.

2) During explosion, it is different though. The added score depends on how many occurrences are destroyed and from which location they are destroyed. If a candy is at bottom, more candies will have to be dropped from top and hence more score.

3) The user is given a total of 15 moves in each level. There is a certain threshold score that is needed to clear each level. If any user fails to reach the threshold score, then it must ask user to try again. A keystroke from keyboard is used for that purpose.

File Handling

- All Individual levels score will be saved in the file.
- Stores the highest score and player name in a same file.
- Record in a file should look like in the format given below

Player's Name
Level 1: 20
Level 2: 30
Level 3: 46
Highest Score: 46

BONUSES

- Restricting mouse cursor to not go outside the board boundaries.
- When bomb is used, all occurrences of the exploding row/col are first highlighted for a second, then explosion proceeds.
- If after swapping, no combo exists, the candies are swapped back.
- Changed background and look of level 3.
- The string 'crushing' is displayed when combos are being crushed and score is being updated.
- 'Explosion' is displayed when a bomb destroys a row or col in board

Divide and Conquer

You need to write different procedures to help yourself out in this project. You can use these procedures/macros (not mandatory) for performing the game:

populateBoard (.)

It populates the board using random candies generated by interrupt.

drawBoard (.)

We have implemented all graphical features. This includes drawing border pixels using 10h interrupt and then drawing characters using 10h interrupt. We have used the video mode 13. The characters are read from the 7x7 'board' array. This function makes use of another function **drawString** that draws the username, score and moves using 10h character draw interrupt.

updateBoard (.)

This function checks for combinations, crushes them, auto-fills and drops until all the combos are removed from the array. It makes use of another function **checkCombo** which is called in a loop unless and until the value of 'crush' variable is set to 0. 0 means there are no more combinations in our board array. Vertical and horizontal combos are checked.

takeInput (.)

It uses the 33h mouse interrupt to check when mouse is pressed. It checks it 2 times, for two cells. Then if cells are adjacent, they are swapped. If no combo is formed after swapping, the candies are swapped back.

initiateBomb (.)

If any candy is swapped with bomb, it destroys all candies of that row or column through which candy is swapped by bomb.

drawString (.)

This function draws the player's name, score and moves left on the top. Along with this, it also shows the current level, and strings of ‘explosion’ and ‘crushing’.

Good luck ☺