SIT323 Notes and Objectives

Author: Arran (Musa) Fletcher Date: 17/8/18

Contents

[Objective 2](#_Toc522271278)

[Testing Requirements 2](#_Toc522271279)

[Crozzle (.czl) Rules 2](#_Toc522271280)

[Configuration (.cfg) Rules 3](#_Toc522271281)

[Letter Sequence (.seq) Rules 5](#_Toc522271282)

[Validation of Crozzle 5](#_Toc522271283)

[Previous Program Constraints 6](#_Toc522271284)

[Deliverable 6](#_Toc522271285)

# Objective

Update processing for new file formats.

Design and use new program test cases.

Update program to coding conventions and standards.

Create 2 portfolio documents

## Testing Requirements

* Validator.IsBoolean (Contains yes or no)
* Validator.IsInt32 (Contains integer without decimal point, no letters or special characters)
* Validator.IsHexColourCode (First character is ‘#’, length 7 characters, only 0-9 and A-F)
* KeyValue.TryParse (If cannot parse throw new exception “Error. Couldn’t parse key value **x**.”)
* Crozzle.Score (Calculate score for letter score, intersections, cannot be run until **after crozzle validation**)
* CrozzleSequences.CheckDuplicateWords (For each sequence add to array [word, occurrence] if more than one occurrence add to duplicate array [word, occurrence])
* Crozzle.Validate (Check no duplicate or group min/max invalid, duplicates not invalid, min/max rows and columns not invalid, minimum never exceeds maximum for any value, etc.)
* Crozzle.ToStringHTML (Build HTML <html><head><style>…</style><body><table><tr><td></td></tr></table></body></html>)
* CrozzleMap.GroupCount (Check for groups, add first sequence to group array, create integer of groups groupCount = 1, groupArray[] = “word”, for each intersecting sequence add to groupArray, if any sequences not in array check first unavailable sequence and all intersections from that sequence, add to groupArray and add new group to group count groupCount++, repeat until all groups accounted for.)
* Config.TryParse (**Run first**, parse configuration of min/max, settings, etc., throw new exception if parse fails “Error. Couldn’t parse config.cfg file. Check config file line **x**.”)
* WordList.TryParse (Check for word list parsing, only alpha no numeric or special characters, if failed throw new exception “Error. Couldn’t parse word **x** in word list. Check word list and try again.”)
* Crozzle.TryParse (Try parsing crozzle, if min exceeds max, crozzle row/column exceeds config, any other config mismatch throw new error “Error. Line **x** of crozzle.czl is invalid with current configuration config.cfg.”, or internal file error “Error. Couldn’t parse crozzle.czl at line **x** please check file and try again.”)

# Crozzle (.czl) Rules

* Leading white space allowed, line can have infinite white spaces permissible
* Characters // to end of line (\n, \r\n) is a comment
* Data and comment can’t be on same line, data line cannot contain “//”
  + Therefore if line has “//” only leading white space can exist
* Absolute or relative paths in file, for OS compliancy if no drive letter is given (C:\) change any \ to / for relative path/UNIX compatibility.
* Crozzle size given as 2D array [x,y]
  + Each cell [x,y] has either letter or null (Example: [1,1] = { “letter”, “A”} [2,1] = { “letter”, null})
  + Can save sequence count per cell for intersecting point count
    - CrozzleMap [1,1] = { “letter” , A }, { “sequences” , 2 }
* Each word given as SEQUENCE=WORD,LOCATION=x,y
  + If horizontal sequence location y (Column start point) cannot exceed crozzle y (Crozzle column limit)
  + If WordLocation.y + count(WORD).characters > CrozzleSize.y throw new Exception(“Word **x** is too large to fit in Crozzle from this location.”)
  + If vertical sequence location x (Row start point) cannot exceed crozzle x (Crozzle row limit)
  + If WordLocation.x + count(WORD).characters > CrozzleSize.x throw new Exception(“Word **x** is too large to fit in Crozzle from this location.”)
* No ordering to data sections given (FILE-DEPENDENCIES, CROZZLE-SIZE, HORIZONTAL-SEQUENCES, VERTICAL-SEQUENCES) can be parsed in any order.

# Configuration (.cfg) Rules

* **Must parse this file first** as it determines rules for Crozzle parsing
* Leading white spaces allowed, infinite white spaces permissible
* From // to new line (\n, \r\n) is considered comment
* Data and comment cannot be on same line
  + Therefore any line with // can only contain leading white space
* SEQUENCES-IN-FILE has minimum and maximum
  + Minimum cannot exceed maximum
  + Must contain only positive integers
  + Total sequences (Vertical + horizontal) must be >= minimum and <= maximum
* CROZZLE-OUTPUT contains output information
  + INVALID-CROZZLE-SCORE contains string to display for an invalid Crozzle score (Example: Invalid. This Crozzle is not valid and cannot be scored.)
  + UPPERCASE must be Boolean (true/false) not colloquial (yes/no)
  + STYLE contains basic HTML styling (Not CSS)
    - Format <STYLE> element { type: value; } </STYLE>
  + BGCOLOUR-EMPTY-TD and BGCOLOUR-NON-EMPTY-TD both contain hex colour value
    - Must begin with #, contain 4 or 7 characters, only 0-9 and A-F
    - If invalid fail parsing and throw new exception “Error. Hex colour **x** is an invalid colour.”
  + CROZZLE-SIZE gives min/max for column and row sizing
    - If min > max throw exception (“Error. Cannot parse line **x** where minimum is greater than maximum.”)
    - If CROZZLE-SIZE x < MINIMUM-COLUMNS or CROZZLE-SIZE x > MAXIMUM-COLUMNS throw new exception “Error. Crozzle doesn’t meet config.cfg sizing configuration at line **x**.”
    - If CROZZLE-SIZE y < MINIMUM-ROWS or CROZZLE-SIZE y > MAXIMUM-ROWS throw new exception “Error. Crozzle doesn’t meet config.cfg sizing configuration at line **x**.”
  + SEQUENCES-IN-CROZZLE gives limits on vertical/horizontal sequence min/max
    - If count(VERTICAL-SEQUENCES) < MINIMUM-VERTICAL throw new exception “Error. Not enough vertical sequences for Crozzle validation.” or count(VERTICAL-SEQUENCES > MAXIMUM-VERTICAL throw new exception “Error. Too many vertical sequences for crozzle validation.”
    - If count(HORIZONTAL-SEQUENCES) < MINIMUM- HORIZONTAL throw new exception “Error. Not enough horizontal sequences for Crozzle validation.” or count(VERTICAL-SEQUENCES > MAXIMUM- HORIZONTAL throw new exception “Error. Too many horizontal sequences for crozzle validation.”
    - All options (MINIMUM-VERTICAL, MAXIMUM-VERTICAL, MINIMUM-HORIZONTAL, MAXIMUM-HORIZONTAL) must be valid positive integer
  + INTERSECTIONS-IN-SEQUENCE gives min/max intersecting sequences for Crozzle.
    - Must count intersections for validation
    - Cannot overlap 2 vertical or 2 horizontal sequences
    - Counting sequences by cell
      * For each vertical sequence check each cell in sequence against horizontal sequence cells, if vertical and horizontal sequence share a cell VIntersection += 1
      * If VIntersection < MINIMUM-VERTICAL or VIntersections > MAXIMUM-VERTICAL throw new exception “Error. Vertical intersection count invalid, couldn’t parse Crozzle.”
      * Check vice versa against horizontal intersections
        + Count of intersections should be the same as each vertical intersection is also a horizontal intersection
  + DUPLICATE-SEQUENCES gives min/max for duplicates
    - Create vertical sequence array, for each sequence add to array [word, occurrence] if more than one occurrence add to duplicate array [word, occurrence], count duplicates (Add all occurrences – 1 per word, as 2 occurrences of a word is simply one duplicate)
      * If duplicate count < minimum or > maximum throw new exception “Error. Too many duplicate words in word list for word **x**.”
    - If duplicate minimum > maximum config is invalid, throw new exception “Error. Duplicate minimum exceeds duplicate maximum at line **x**. Invalid config.cfg.”
  + VALID-GROUPS gives a valid group count for Crozzle.
    - Check for groups, add first sequence to group array, create integer of groups groupCount = 1, groupArray[] = “word”, for each intersecting sequence add to groupArray, if any sequences not in array check first unavailable sequence and all intersections from that sequence, add to groupArray and add new group to group count groupCount++, repeat until all groups accounted for.
    - If groupCount > maximum throw new Exception “Error. Crozzle contains too many groups to be valid with this configuration.”
    - If minimum < 1 throw new exception “Error. Crozzle cannot have less than one group. File config.cfg invalid at line **x**.”
  + INTERSECTING-POINTS gives the letter score for each intersecting letter
    - Must contain all 26 English alphabet characters
    - Cannot contain any number or special character before =
    - Must be a valid integer after =
  + NON-INTERSECTING-POINTS
    - Must contain all 26 English alphabet characters
    - Cannot contain any number or special character before =
    - Must be a valid integer after =

# Letter Sequence (.seq) Rules

* Formatted as comma separated values file (.csv)
* Example
  + [A- Z]+,40,14,1042,1096   
    APPLE,10,5,370,385   
    BAG,10,3,202,215
* Header field
  + Regex pattern for identifying allowed characters in letter sequence
  + Batch total for field 2
  + Batch total for field 3
  + Batch total for field 4
  + Batch total for field 5
* Columns
  + Letter sequence (Example: APPLE)
  + Points obtained when using sequence in valid crozzle (According to matching .czl and .cfg)
  + Character length for validation
  + Sum of ASCII values of all characters combined for sequence
  + Hash total, sum of fields 2, 3, and 4

# Validation of Crozzle

* Parse and validate 3 files (.cfg, .czl, .seq)
* Write any errors and exceptions to log.txt, display through GUI
  + Config error (Example: missing data fail, minimum exceeds maximum logic fail, data and comment on same line syntax fail, options title incorrect syntax fail)
  + Crozzle error (Example: missing data fail, data and comment on same line syntax fail, Crozzle conflicts with config logic fail, sequences conflict with config logic fail, sequences conflict with crozzle logic fail)
  + Sequences error (Invalid sequence syntax fail, invalid regex logic fail, missing data column syntax fail, data invalid size (Hash incorrect) logic fail, data invalid type (Not a valid integer) syntax fail, data conflicts with config logic fail, data conflicts with crozzle logic fail)
* Attempt to display invalid Crozzle on GUI
  + Invalid fields displayed in red?
  + Ignore invalid fields?
  + Render valid info and display Crozzle invalid score text?
* Use NON-INTERSECTING and INTERSECTING to calculate Crozzle score
  + - For each cell [1,1] to [x,y] (CrozzleSize)  
      if cell.letter == null  
       break  
      else if cell.letter != null and cell.sequences > 1   
       CrozzleScore += intersectingScore[cell.letter]  
      else  
       CrozzleScore += nonIntersectingScore[cell.letter]
    - For invalid Crozzle display invalid text and exception error message.

# Previous Program Constraints

* Valid Crozzle must conform to configuration settings
* Each sequence of 2+ horizontal characters must form a letter sequence from within .czl
* Horizontal must be left to right
* Vertical must be top to bottom
* Diagonal sequences not validated

# Deliverable

* Produce 2 word documents of unit tests (pre-test) and test output
* Portfolio document 1 program
  + Title
  + Contents
  + Introduction
    - One section per test (File set 1, 2 and 3)
    - Summary and output (GUI, log file, exception message)
  + Document body
  + Appendix
    - One section per test (File set 1, 2 and 3)
    - Include copies of input data provided for test
* Portfolio document 2 unit tests
  + Title
  + Contents
  + Introduction
  + Document body
    - One section per test (File set 1, 2 and 3)
    - Brief summary and result of test output
  + Appendix
    - One section per test (File set 1, 2 and 3)
    - Copies of input data and files