File Input/Output

1. Learning aims

At the end of these exercises you should be able to:

- Read from and write into text-files

2. Course Material

To make the exercises of this lab, you need the following parts of the book

- Processing: A programming Handbook for Visual Designers and Artists (C. Reas & B. Fry)
    - Chapter 28: Arrays p 428
    - Chapter 32: Text-files p 489-500
- Slides PDF file
- https://www.youtube.com/results?search_query=processing+tutorial+from+beginner+to+games
- How to read and write from a txt-file:
    - https://processing.org/reference/loadStrings.html
    - https://processing.org/reference/PrintWriter.html
    - https://processing.org/reference/saveTable.html

3. Lab exercises

3.1. Demo_08_01_fileInputOutput

A demo is given with the different functions for file input/output.

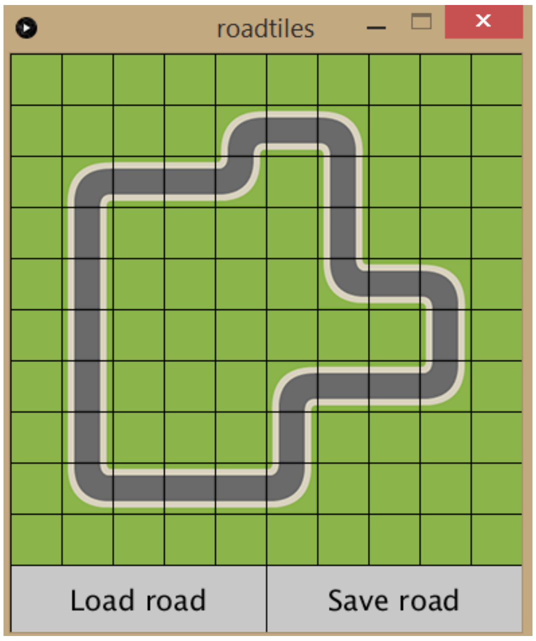
3.2. Lab_08_02_roadTiles

The goal of the application is to build a grid with road parts loaded from a text file. You should be able to save the grid (field) and load it again later.
Note: to lower the complexity, the number of rows and columns in the grid (field) is always exactly the same.

Save the road tile images and roadimages.txt in the sketch data folder. Set the size of the sketch to 380 by 430, a green background (#8bb54a) and create a variable **fieldSize** with a fixed value of 38.

Before you continue, have a look at the screenshot below and make sure you understand:

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	4	0	1	-1	-1	-1
-1	4	0	0	3	-1	2	-1	-1	-1
-1	2	-1	-1	-1	-1	2	-1	-1	-1
-1	2	-1	-1	-1	-1	5	0	1	-1
-1	2	-1	-1	-1	-1	-1	-1	2	-1
-1	2	-1	-1	-1	4	0	0	3	-1
-1	2	-1	-1	-1	2	-1	-1	-1	-1
-1	5	0	0	0	3	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1



* each value in the matrix represents a road tile

* if the value is -1, no road tile is displayed

* if the value is 0 or higher, a corresponding road tile is displayed

STEP 1 : BUILDING THE FIELD

- Use the *roadimages.txt* file to fill an array with road tile images (the size of the image array is based on the number of lines in the textfile)
- Create a variable "**gameField**" that is a 2D array of integers.
    - Each value in this array will indicate an index of the image array (fe., if the index is 2 then the image in position two of the road tile images will be displayed), as was explained in the screenshot above.
    - Provide 10 rows and 10 columns in the 2D array and initialize all values to -1 (indicating that no road tile is present in this row/column). Draw the grid.
    - Every time the user clicks inside the field, the value of the corresponding row/column is increased by 1. If this index value reaches the end of the road image array length, reset it back to -1 (erasing any row that was there before in this row/column).

    - Create a method **drawField**() that loops through the 2D array:
        - if the index in this row/column is not -1, the corresponding image from the road image array is drawn. You must repeat the black border, draw a transparent rectangle above the image, using the correct coordinates and size.

STEP 2 : LOAD/SAVE THE GAME FIELD

- Display two buttons (height=50) at the bottom of the sketch (rectangle + text).

- If the user clicks the "**Load road**" **button**, a default road field (defaultRoad.txt) is loaded directly into the game field 2D array. You can use the given file from your source material for this and place it in the data folder.

Hints:

- loop through the gameField 2D array as you did before.
- each line in the text field is a row in the array
- if you split the line by the ';' character, each part of the line is a column in the array
- convert each part to an integer so you can save it in the gameField array in the correct position

If the user clicks the "**Save road**" **button**, the 2D game field is written to a new file (newRoad.txt)

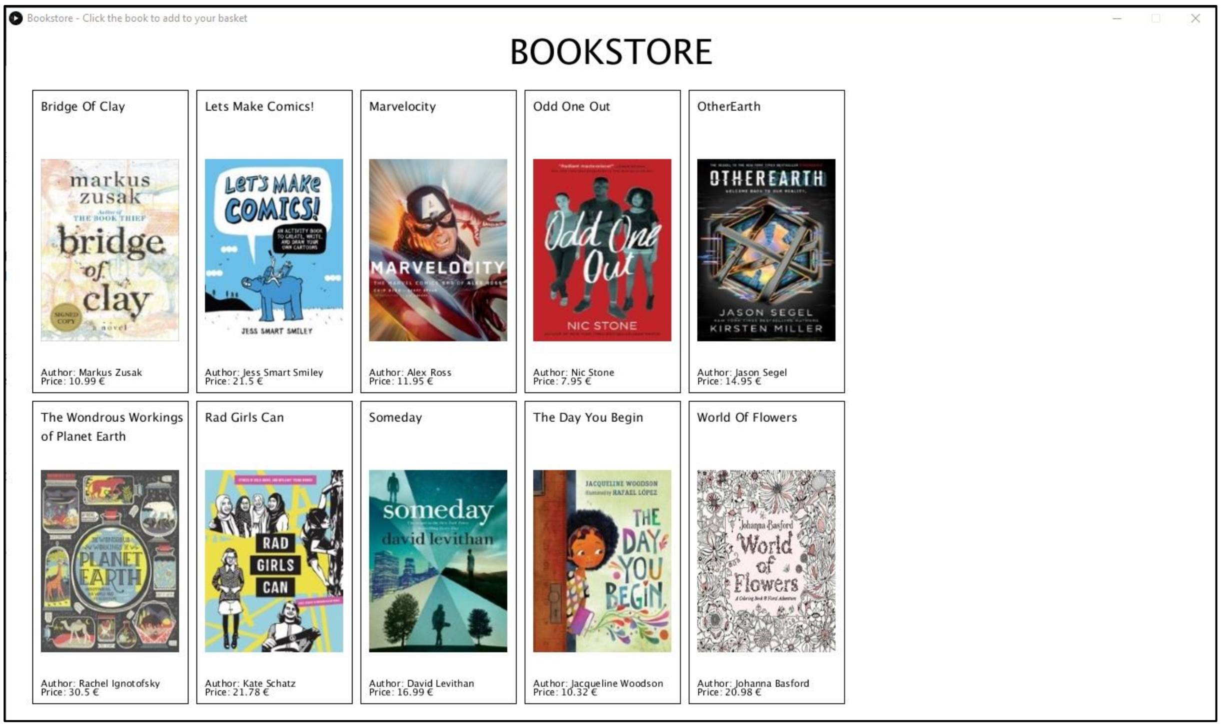
Hints:

- loop through the gameField 2D array as you did before.
- create an empty String "line" in the OUTER for loop, before the inner for loop starts
- in the inner for loop, add the value from the gameField array to the string, followed by a semicolon.

3.3. Lab_08_03_bookStore

You make an application to order books in a Bookstore. The data (title, author and price) can be found in a txt-file ‘books.txt’. The bookstore contains 10 books which are displayed in 2 rows. When you click a book, the book is added to your basket. Yu can not add a book twice to the basket!

When you click the button “ORDER BOOKS”, the content of the basket is written to a txt-file ‘basket.txt’.



Bookstore - Click the book to add to your basket

BOOKSTORE

Bridge Of Clay
Author: Markus Zusak
Price: 10.99 €

Lets Make Comics!
Author: Jess Smart Smiley
Price: 21.5 €

Marvelocity
Author: Alex Ross
Price: 11.95 €

Odd One Out
Author: Nic Stone
Price: 7.95 €

OtherEarth
Author: Jason Segel
Price: 14.95 €

The Wondrous Workings of Planet Earth
Author: Rachel Ignotofsky
Price: 30.5 €

Rad Girls Can
Author: Kate Schatz
Price: 21.78 €

Someday
Author: David Levithan
Price: 16.99 €

The Day You Begin
Author: Jacqueline Woodson
Price: 10.32 €

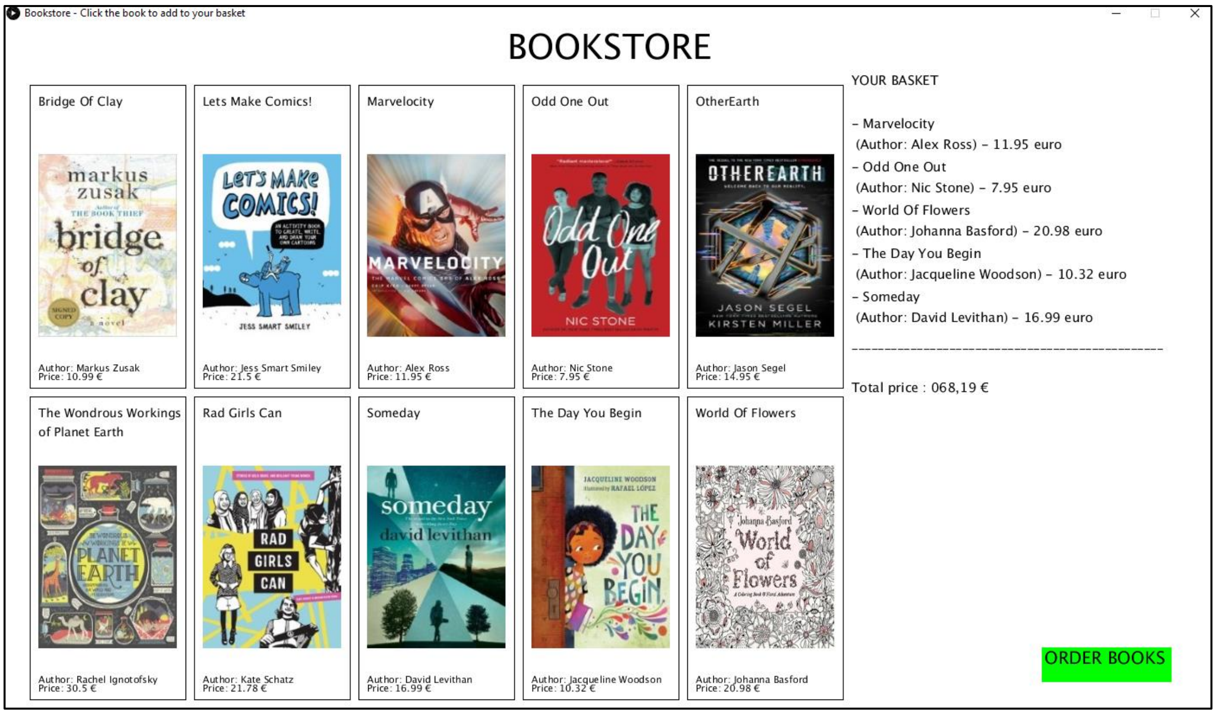
World Of Flowers
Author: Johanna Basford
Price: 20.98 €

- The size of the window is 1400 x 800 and the background is white.
- Display the window title: **Bookstore - Click the book to add to your basket**
- Make a **class Book** with:
  - **Data members** title, author, price, image and position to display a book, a **constructor** and if necessary getters and setters.
  - a method **display** to display a book with a gray rectangle (width = 190, height = 360) and title, author, image and price (cf printscreen) .
- a method **hitTest()** in which you check if the user clicked a book

- **Loading and displaying the books**: While starting the application you read the txt-file "books.txt" and you make book-objects stored in an array of books. Show 5 books per row.

- Make a class **Basket** with:
  - an **array of ordered books**.
  - a method **addBook(Book b)** where you add a book to the basket when a book is clicked.
  - a method **getTotalPrice()** where you calculate and **return** the total price of the basket. Use the nf() function!
  - a method **displayBasket(PVector pos)** where you display the basket on your screen (cf printscreen).

- Make a **class Button (use the button class of the demo!)** to display a button with a green background at the bottom of your window. The text is "**ORDER BOOKS**". The button is only visible when there are books selected to the basket.



Bookstore - Click the book to add to your basket

BOOKSTORE

Bridge Of Clay Author: Markus Zusak Price: 10.99 €	Lets Make Comics! Author: Jess Smart Smiley Price: 21.5 €	Marvelocity Author: Alex Ross Price: 11.95 €	Odd One Out Author: Nic Stone Price: 7.95 €	OtherEarth Author: Jason Segel Price: 14.95 €
The Wondrous Workings of Planet Earth Author: Rachel Ignotofsky Price: 30.5 €	Rad Girls Can Author: Kate Schatz Price: 21.78 €	Someday Author: David Levithan Price: 16.99 €	The Day You Begin Author: Jacqueline Woodson Price: 10.32 €	World Of Flowers Author: Johanna Basford Price: 20.98 €

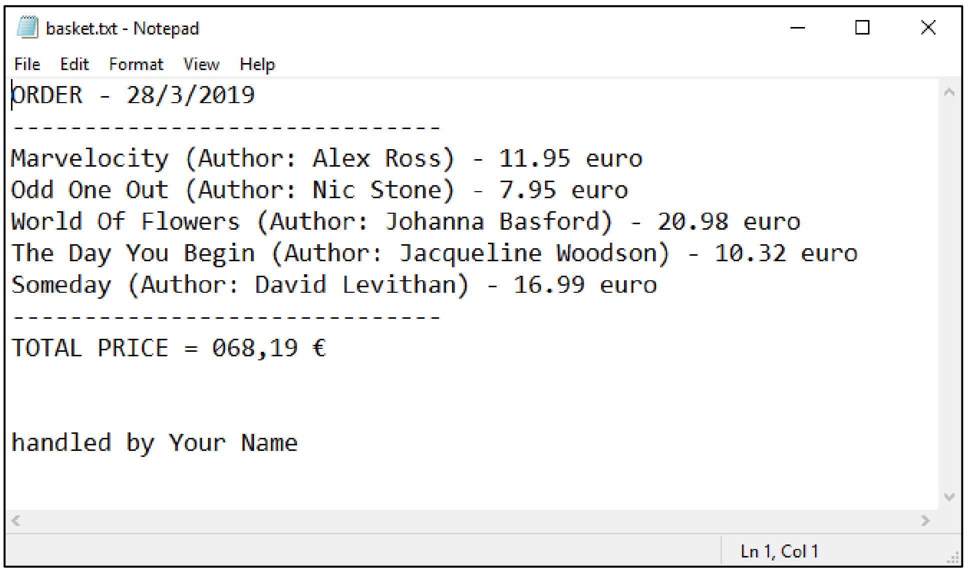
```
YOUR BASKET

- Marvelocity
 (Author: Alex Ross) - 11.95 euro
- Odd One Out
 (Author: Nic Stone) - 7.95 euro
- World Of Flowers
 (Author: Johanna Basford) - 20.98 euro
- The Day You Begin
 (Author: Jacqueline Woodson) - 10.32 euro
- Someday
 (Author: David Levithan) - 16.99 euro

------------------------------------------------

Total price : 068,19 €
```

❖ When the **Order-button** is pressed, the content is written to a txt-file named "basket.txt" . The lay-out is displayed below. The date (first line) is not hard-coded but is the current date. You add your name in the last line.



```
ORDER - 28/3/2019
------------------------------
Marvelocity (Author: Alex Ross) - 11.95 euro
Odd One Out (Author: Nic Stone) - 7.95 euro
World Of Flowers (Author: Johanna Basford) - 20.98 euro
The Day You Begin (Author: Jacqueline Woodson) - 10.32 euro
Someday (Author: David Levithan) - 16.99 euro
------------------------------
TOTAL PRICE = 068,19 €


handled by Your Name
```

4. Saving instructions

- Make a folder named 1DAE***xx***_PROGFA**2**_**08**_***name_firstname***
  (e.g. 1DAE03_PROGFA**2**_**08**_Van_der_Veken_Jan)
- Add all the folders with the corresponding Lab08 pde-files.