AUTOMATION OF LAYOUT AND PREPARATION FOR PRINTING GRAPHIC MATERIALS FOR BOARD GAMES

Zhelenok D. A., Savenko A. G.
Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus
E-mail: savenko@bsuir.by

The paper presents a software tool designed to automate the layout and preparation for printing of graphic materials of board games. The software has a graphical and console interface, a well-thought-out architecture that allows you to avoid duplicating the source code of algorithms for different interfaces. The main functions of the software tool are: preparing a specification file with a wide selection of parameters (various types of printing materials, sheet formats (including roll printing), options for source files of board game materials and ways to compare them, choice of packaging strategy); generation of documents for printing in accordance with specifications. Software algorithms solve problems of two-dimensional packaging and guillotine cutting. The use of software will reduce the costs of preparing and producing board games.

Introduction

Today, board games are a common hobby that forms an entire industry. They have a wide variety both in design and theme, as well as in genres, ranging from abstract and logic games to strategy and role-playing games. There are many communities, clubs, and online resources dedicated to board games. The process of developing new games or additions to existing ones is most often iterative. This means that before creating the final version of the product, a prototype is developed, the ideas and mechanics are tested on it through the game itself, then edits are made, the ideas are tested again, etc. Each prototyping and testing cycle is a separate iteration of game development. After developing a polished version of the product, the author can contact publishers to present them with his work and, possibly, publish a game ready for sale through them. He can also try to publish it himself, working directly with printing houses. Or, as an alternative, release its materials in digital form, either free or for a fee through specialized resources. In addition, the next re-release of board games is often accompanied by changes to the rules and mechanics of the game. Changes in the rules are also often accompanied by changes in both the information contained on the game components and the components themselves. An iterative approach and the release or re-release of a board game in any form implies the layout of components for their further printing. Simplifying and increasing the quality of layout allows you to reduce the cost of publishing board games. This, in turn, reduces the cost of individual copies, which increases accessibility to reach a larger audience. Different materials are used to make different types of components. Different components require different preparation procedures for further manufacturing. The preparation of components for production differs between the development and publication stages. So, due to the iterative nature of the development process, changes are made frequently, but they are aimed primarily at the rules and game mechanics, and not at the design of components (although this can be thought through and changed at this stage). At the publishing stage, the design must acquire its final form, which will be convenient both for production and for the end consumer. In the context of board games, the layout process involves arranging game elements so that they are easier to print and cut. At the same time, the paper format and the layout of objects depend on where the layout objects are subsequently produced.

I. Proposed solution

The developed software allows you to automate the process of layout and preparation of board game materials for printing. The PS supports a hybrid operating mode, which means support for both graphical and text (console) interface options. The graphical operating mode provides the user with a convenient interface for creating a specification file and its use for generating documents. The specification file is in YAML format, which in declarative form describes the parameters that are expected from the generated documents.

The specification file describes the following parameters:

- the structure of folders in which the results of the program will be saved;
- the number and name of documents that the user expects to receive;
- paper format and indents that must be observed in each document;
- source of images of game components and their expected size;
- strategy for packing objects on each page;
- source of cover images, if the component requires them;
- a method for forming cutting marks, if necessary.

Each of the points above allows you to customize the expected result. The program, reading the specification and other necessary files, can provide the following functions:

– arrangement of images on a two-dimensional limited area, taking into account indents, expected sizes of components and the packaging strategy chosen by the user;
– scaling input images, as well as forming a frame around them if it is necessary to avoid problems associated with paper displacement during printing;
– slicing input images if they do not fit into the expected area;
– placing marks for cutting;
– генерация и расстановка рубашек для тех компонентов, которым она необходима;
– the ability to distribute components both within one document and in different ones.

At the output, the program generates a structure of folders containing generated files in the Portable Document Format (PDF). The folder structure is also described in the original specification. Packing is a class of combinatorial geometry problems that involve placing figures of a given size or shape inside another given figure with the greatest economy or with some other restrictions [1]. Packaging strategy refers to the algorithm by which game components will be placed on document pages. In the developed software, packaging algorithms are divided into two main groups: optimizing the space used and simplifying post-printing processing. The first are algorithms that solve the two-dimensional packing problem (2DBP). They must be able to handle components of different sizes and their result must, if possible, minimize the space taken up on the sheets and the number of sheets of paper used. The latter must minimize the number of guillotine cuts that need to be made to cut out all the components. This problem is also known as the guillotine cutting problem. The problem solved by two-dimensional packaging algorithms is formulated as follows: given rectangles of a certain size (width and height are specified), which must be placed on larger rectangles (containers). The main goal is to find such an arrangement of all rectangles so that they do not intersect and occupy the least number of containers [2].The two-dimensional packing problem has two options for input data: when the set of objects to be packed is known in advance (offline problem) and when the data arrives in portions (online problem). The developed software uses algorithms that solve the offline problem, since the size of all game components is known in advance. Another variation of the two-dimensional packaging problem, which is algorithmically implemented in this software, is the 2-Dimensional Strip Packing (2DSP) problem. This option differs from 2DBP in that packaging is carried out in only one container, which is limited in width, but not limited in height. Instead of minimizing the number of containers, we minimize the filling height of one [3]. The guillotine cutting problem is a problem of combinatorial geometry, close to the cutting problem and container packing problems. The main goal is to obtain the maximum number of rectangular-sized sheets from a larger sheet by making only guillotine cuts, that is, straight cuts from edge to edge. The functions of the software tool can be conditionally divided into two categories: one allows you to create a specification that describes how to prepare the source data and convert it into a document ready for printing, the second allows you to convert the input materials into the final document. To transform a document, primitives such as point, size, and indentation are used. These in turn are combined into more complex entities: plain paper, roll paper and components. Packing objects in a two-dimensional limited space is carried out by packing strategies. To structure the model, the strategy design pattern is used. Strategy is a behavioral design pattern; it defines a family of algorithms, encapsulates each of them, and makes them interchangeable. The strategy allows you to change algorithms regardless of the clients who use them. Packing strategies return a packed document. It describes where which primitives should be located. This approach allows you not to be tied to a specific format for export, implementing it separately. The paper, component set, and packaging strategy enable the document specification to be represented. After creating a packaged document, you need to convert it to a file or display it. This again uses the strategy design pattern, but this time it describes classes for different rendering methods. Before packing the components, preparation is carried out. It involves reading and converting input images of the front and back of the components.

II. Conclusion

A software tool was developed to automate the layout and preparation for printing of materials for board games, solving the problem of optimal packaging in two-dimensional space and the subsequent problem of guillotine cutting. The developed software can be used both in production by board game publishers and by private users to prepare and print board game components using the PNP (print and play) method or develop their own games. The use of the developed software will reduce the costs of preparing and publishing or reprinting board games and significantly simplify the process of preparation and production.

Bibliography

1. Britanica Packing Definition. [Electronic resource]. – Access mode: `https://www.britannica.com/science/packing-combinatorics`. Access date 10.10.2023.
2. Nikhil Bansal, Maxim Sviridenko. New Approximability and Inapproximability Results for 2-Dimensional Bin Packing / Maxim Sviridenko Nikhil Bansal. – 2003.
3. Habr: Про двумерную упаковку: offline алгоритмы. [Electronic resource]. – Access mode: `https://habr.com/en/post/136225`. Access date 10.10.2023.