Intelligent System of Visualization and Creation of Choreographic Setting

Andrii Stefanchuk[1], Olga Lozynska[0000-0002-5079-0544][2], Taras Basyuk[0000-0003-0813-0785]3, Oksana Oborska[4], Anatolii Vysotskyi[5], Khrystyna Mykich[0000-0002-4324-2080][6], Iryna Zavuschak[0000-0002-5371-8775][7]

Lviv Polytechnic National University, Lviv, Ukraine

Taras.M.Basyuk@lpnu.ua

Abstract. The purpose of this tusk is creation of an intelligent system for visualization and creation of choreographic performances, which would enable the user to perform his tasks related to dance performances quickly and conveniently. To achieve this goal, the following tasks were set: to carry out researches of the literary sources concerning a subject of work; to carry out the analysis of similar systems and to prove the development of the studied system; to carry out systematic analysis and substantiate the problem; to make the choice of methods and means for solving the problem; implement software implementation of the mobile application to solve the problem; - to test the developed system. Object of study is the process of visualizing and creating choreographic productions. Subject of study is methods and tools for building an intelligent visualization system and creating choreographic productions. Research methods is application of system analysis methods. The obtained results enable the choreographers to visualize their productions quickly and effectively. With the help of a developed system, they can overlay music on their production and arranged pictures. This information will be accessible to them from any device as the data will be stored on the server.

Keywords. Visualization, Dance, Choreographic Scene, Information System, Mobile Application, Choreographic productions.

1 Introduction

The history of dance begins with the history of human civilization. Initially, people tried to communicate with the deities in this way. Prayers for good hunting and abundant crops were expressed in the dance. As soon as sounds began to form tunes and music, so immediately people began to move into a rhythm, creating a dance [1-7]. Gradually the chaotic movements became clearer and over time the dance became a full-fledged kind of art with its history and its various kinds. Dance was created in many forms and was used in many places. These could be ritual tribal dances, wedding dances, choreographic performances in the theater, demonstration dances before gladiator battles, dances to support football teams, dance music by dancers or purely dance shows. Hundreds of years of development of dance art have formed many of its manifestations - both official and widely recognized, as well as narrow and little known. However, the notion of choreography is a common concept for describing dance on stage. In Greek, "choreography" means the recording of movements. But the meaning of this word has become much broader and the term “choreography” today includes everything that pertains to the art of dance: professional classical ballet, folk and ballroom dancing, modern dance, variety dance and contemporary dance - all this is called choreography. Usually, people perceive dance as entertainment or a classic art that is shown in theaters. However, in recent years, dance and choreography have gained a lot of popularity on the Internet and are rapidly going beyond the classical representation. Modern dances are often found in the following cases: massive flash mobs in support of anything; choreography for music videos and commercials; production of a demonstration performance of the dance team / group; statement of support groups; performances of show-ballets together with popular artists; dance videos that can garner a huge amount of views through the development of social networks.

The core of the development of the modern dance industry can be considered Los Angeles. More than one hundred vocational training studios have been opened there. Dancers from these schools take part in world-class concerts such as Beyoncé or Drake, staging a long-running show. They also shoot videos that are watched by many users of video services such as YouTube and Instagram [8-12].

Due to this rapid development of the dance sphere, the dancers have moved from the status of a dancer on the background of the star to the status of stars who perform on a level with the singer or the central figure on the stage. Modern technologies allow you to organize an incredible show and capture the attention of the viewer very firmly. This involves the organization of the stage, the lighting at every moment of the show, the transformation of the stage, the movement of dancers across the stage and the direct choreography. Due to the size of the stage, the viewer often does not pay attention to the movements of individual dancers, but pays more attention to the overall picture of all the dancers on stage. The choreographer, who creates and arranges the choreographic drawings, arranges this painting on stage.

Good dance performance is not always an easy task and requires considerable hands-on experience, musical listening and creative skills, including the ability to improvise, think artistically, and be able to show your intent through dance. Initially, the choreographer comes up with the idea of staging. Then he finds and composes the music that best describes the idea of the number. To create the most stunning performance that will be voluminous and use the entire stage area, you need to correctly and competently arrange the dancers at every moment of time.

Creative and punchy drawings can turn music and choreographic movements into an unforgettable number that is unmatched by others. Dynamic and spectacular changes in the arrangements of the dancers on the stage are a tool that keeps the viewer energized and draws attention to the required areas of the stage.

2 The Problem of Rendering Productions and Choreographic Drawings

Typically, the duration of choreographic performances varies from 3-4 minutes to 12-20 minutes. Not always enough is enough to keep in touch with the viewer of an interesting idea of a room. An important role is played by the emphasis on what is happening in a particular part of the scene. The classic way of accentuating among choreographers is interesting tricks and dynamic changes in the arrangements on stage (change of pictures). In order to organize and memorize the entire choreographer's performance, certain tools are clearly needed. Likewise, dancers need to know and remember all their positions throughout the room. In the case of a short number with the participation of several people, you can do the standard entries and sketches on a piece of paper. In this approach, the participants of the number are represented by bold dots, and their movements on the stage - by arrows. The additional designation of the dancers' names creates a clutter of drawings.

However, when the number of participants in a production increases or the duration of a number exceeds the mark in a few minutes, problems arise. It becomes difficult to memorize or draw it all on paper. It becomes difficult to make some changes to the created drawings. There are many questions and misunderstandings between the dancers themselves. The very process of arranging the dancers on stage is long and exhausting. Initial developments of the choreographer require changes and modifications. It is especially difficult to navigate when there are many drawings and they are unusual and require the dancer’s maximum attention. Typically, the image of the choreographic drawings looks as it does in Fig. 1.

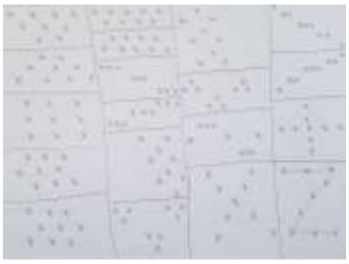


Fig. 1. Image of choreographic drawings on paper

Some problems arise due to the dynamic changes in the number of dancers in rehearsals. In the absence of some people in training, there is a need to replace them with handy tools, such as shoes, to mark their placement on stage. Increasing or decreasing the number of dancers necessitates a change in all developments and can cause major problems due to the violation of symmetry and beauty of the figures. Music is a very important factor in creating productions. Transitions are created using mood and music dynamics. In the process of rehearsals and arrangements of dancers on stage, the imposition of music on the created drawings and arrangements takes almost the most time. This is because the dots on the paper are in no way correlated with the music track. The choreographer can usually record at what point the dancers must move to other positions, but this process will take a long time. Therefore, using standard methods for rendering productions, the production choreographer is likely to run into the following problems: much time used for large productions; the need to change all the achievements, even with a minimal change of the group of dancers; communication of the choreographer with the dancers to breed all the drawings, which will take a long time to achieve the desired result; the gravity of the names of the dancers; work with complex forms; lack of means to synchronize music; the difficulty of spreading the drawing to a group of people and the incomprehensibility of the mark to others; the problem of making changes and redrawing the drawings. The most obvious and convenient solution for choreographers is to create a mobile application that combines all possible features to simplify the production process.

3 Analysis of Known Systems for Visualization of Choreographies

Analyzing the market for mobile applications, web applications and software, not many systems were found to solve the problem of rendering choreographic performances. There are many different solutions for team player placement such as football, basketball or volleyball. There are also many programs for learning about certain dance moves or working with music tracks. However, several solutions for the iOS mobile operating system have been found and analyzed. The main criteria for evaluating found applications were the following indicators: speed of performance of operations on productions; user interface convenience; functionality; user reviews. Among the applications found, the following was analyzed: PlayBook Dance, ChoreoRoom, StageKeep.

- *PlayBook Dance.* The application lets you create a sequence of choreographic drawings and see animations of transitions between them. It is possible to name dancers and perform basic operations on them (Fig.2). There is an opportunity to share the work, so the task of rendering the application performs. However, this application has the following disadvantages: the application is not updated to the current version of the iOS operating system, which does not allow users to use the application; a large number of defects not taken into account details; outdated user interface; basic functionality only.

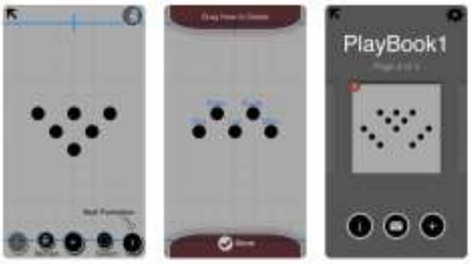


Fig. 2. Application interface Playbook.Dance

- *ChoreoRoom.* The functional application repeats the first example, but contains a number of improvements and modifications. It allows you to create new layouts better and faster. Added the ability to assign different colors to dancers (Fig.3). The graphics of the application are optimized and clear. However, the application does not add anything radically new and one of the drawbacks is the large size of the scene, which does not change and is not always necessary.

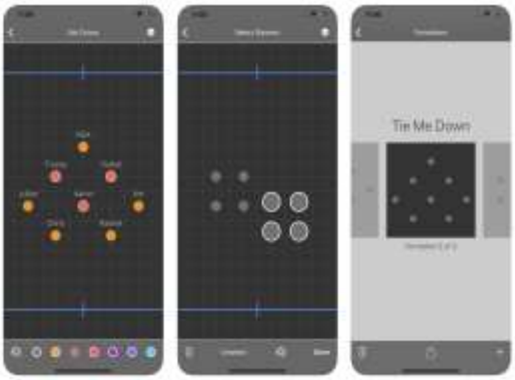


Fig. 3. Application interface ChoreoRoom

- *StageKeep.* Among all found applications, StageKeep has the greatest functionality and solves a number of additional tasks that are not solved by other applications. This app allows you to synchronize your play with music and supports tablets. The disadvantages are the high cost of the application and the complexity of use (Fig.4).

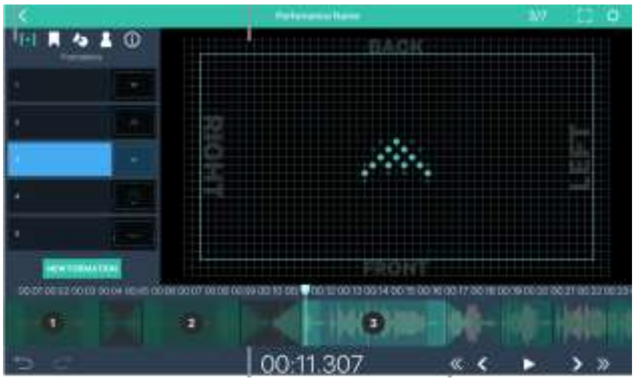


Fig. 4. Application interface StageKeep

None of the products under consideration offers the user work with forms and does not contain data dissemination within the system. A potential enhancement to existing systems is to offer the most optimal movement between images.

Among the considered analogues were highlighted the main functions that the systems that solve the problem of visualization of choreographies have. The most important indicators are performance and usability. Another important factor is the small features that help the user and eliminate the need to use other tools in the process of rendering the choreography. These include changing the color of dancers or adding notes to drawings. The analysis has shown that the urgent task is to design an intelligent system for creating choreographic productions using a mobile device so that valuable information is available to the user at any time. The new system will be able to correct any shortcomings of existing solutions and help the user to visualize their productions as efficiently as possible.

4 System Analysis of the Object of Study and Subject Area

There are several basic principles of a systems approach. Among them were chosen the best fit into the analysis of intelligent visualization system of choreographic productions:

- *The principle of the ultimate goal.* This principle emphasizes that the ultimate goal is the highest priority [13-17]. The formulation of the ultimate goal is critical for the systematic approach, since all further studies will be based on it. Any change should have an impact analysis on the final goal. The main purpose of the intellectual system is to create the arrangement of dancers on stage, their transitions between drawings and synchronization with music. The system should also analyze the arrangements of the dancers and formulate the best subsequent placements.
- *The principle of unity and connectedness*. Guided by this principle, we consider systems collectively as one unit, consisting of many separate parts. Each subsystem performs a set of functions [18-21]. It is possible to identify which components are completely independent and which are dependent on others. This analysis will also help to break down the designed system into submodules, which will improve the system's functioning and facilitate component reuse. The system will contain such basic modules - work with input, work with dancers on stage, work with a list of works, work with music synchronization, work with a profile of the user, work on data and generation of offers.
- *The principle of modularity.* When decomposing the system, modules of varying degrees of generality are highlighted [22-26]. In particular, the template generation module can be divided into two sub-modules: identifying the most popular placements and generating the most optimal routes. Also worth mentioning is a submodule for analyzing user operations on stage and a module for working with a music track.
- *The principle of hierarchy.* This principle involves building a hierarchy of elements in the system and ranking them [27-31]. This process makes it easy to view and build the system components as a whole. By building a hierarchy, the ordering and detailing of the components of the system by im-

portance were implemented (the component of moving a dancer to a new position includes the process of analyzing the movement of a dancer from previous positions).

- *The principle of functionality.* Each function of the system has its own priority and its place in the overall structure [32-34]. When new functionality is added, the entire structure is revised and modified as needed. The main functions of the system are the functions of choreography and work with music, and they have the highest priority. The editing functions of the dancers themselves, their names, and the addition of notes have lower priorities as they are additional functions of the system and do not form the basic functionality.
- *The principle of development.* The developed system is capable of expansion with new functionality and accumulation of information [35-37]. You can add music sync and conversion time depending on the dynamics of the music, etc. to the main functionality. It is also worth considering the user feedback factor that can generate a variety of ideas for improving the system.
- *The principle of decentralization.* In managing a system, the relationship between centralization and decentralization is determined by the purpose and purpose of the system. The system is centralized because it is used by the user, who uses its capabilities.
- *Modeling general requirements.* In order to describe the general system requirements [38-41], a detailed description of precedents should be made according to the RUP, as it is the most informative and useful compared to other methods. Rational Unified Process (RUP) is one of the most common flexible software development methodologies developed by Rational Software. According to this technique, the product is updated about once every six months, but it is possible more often. Unified Modeling Language (UML) is used to model the common knowledge base. RUP provides iterative product development. Each iteration is defined by a set of goals that must be completed by the end of the iteration.

Therefore, the general requirements for the system include the following:

1. *Interested persons of the precedent and their requirements*:
   - *Choreographer:* The main user of the system that will create the productions he wants to visualize. The main indicator for the user will be the speed and convenience of transactions. The user has no trouble doing all the manipulation of the scene, dancers and music. After creating the production, the choreographer should be able to expand his work with others.
   - *Dancer:* This system user will be able to view the created choreography by another user and track their movements on stage during the number. He will also be able to comment on a particular post.
2. *The user of the software system.* This program can be used by any user of the device running the iOS operating system, but it will be applicable only for choreographers, dancers and people who create different kinds of staging, whether performing on stage or performing at a stadium support team.

3. *Prerequisites for precedent:* User login; User registration; Successful retrieval of stored user data.
4. *Successful scenario:*
   - the user logs in;
   - a user creates a new account or uses an existing one;
   - a user requests a list of their saved choreographs;
   - the system processes the request;
   - the user views, edits and creates new data in the system;
   - the system displays all changes through the user interface.
5. *Scenario Extensions or Alternative Streams*. The system should issue errors to the user if there is any error on the server side or show warnings if certain conditions have been violated and the user cannot take some action on the system.
6. *Post-conditions:*
   - output a list of choreographs and templates at the request of the user;
   - display animations of transitions between different images;
   - information is stored locally and remotely;
7. *Special system conditions:*
   - validation of user actions;
   - ability to undo the most recent actions taken;
   - real time work;
   - clear graphic elements.

Conceptual model of the system. You can imagine the system using a conceptual model. A number of concepts are used to model the system in detail, to make the system more explicit. Some models may be physical objects representing the concept. Conceptual models are often abstractions of real-world objects.

Building a diagram of use cases. The precedent chart represents all possible user interactions with the system and the types of users interacting with the system. The use case is one of the main techniques used in system analysis, which is useful for defining, detailing and organizing system requirements. In the context of the use case diagram, the system is called what is being developed and will be operated. In our case, it will be a mobile application for visualizing choreographic performances. In Unified Modeling Language (UML), there are standards for marking certain real objects using precedent diagrams. The system provides basic functions for creating choreographic productions that can be viewed, listened to with music and distributed to a team of dancers. The precedent diagram is shown in Fig.5. The diagram shows the main functions of the system, including displaying a list of choreographs, creating and editing choreographs, manipulation of dancers on stage. Three types of users are also visible:

- a choreographer who has access to all the basic functions of the system;
- a premium user who will have paid premium functionality, including music sync support;
- a dancer who will have the right to review and comment on their productions.

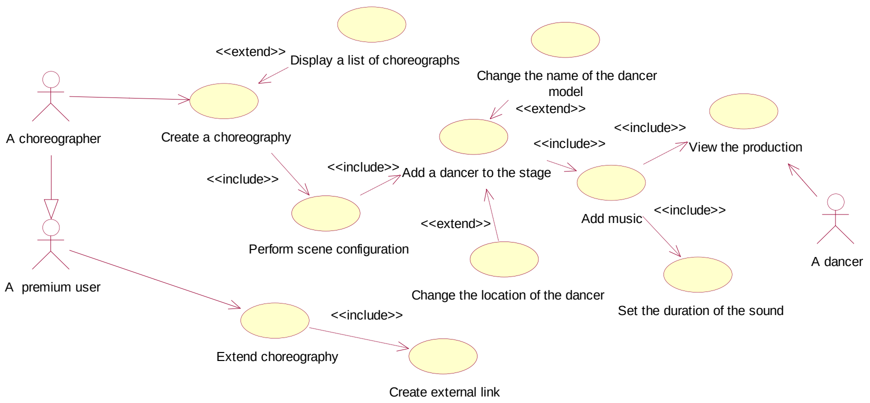


Fig.5. Use case diagram

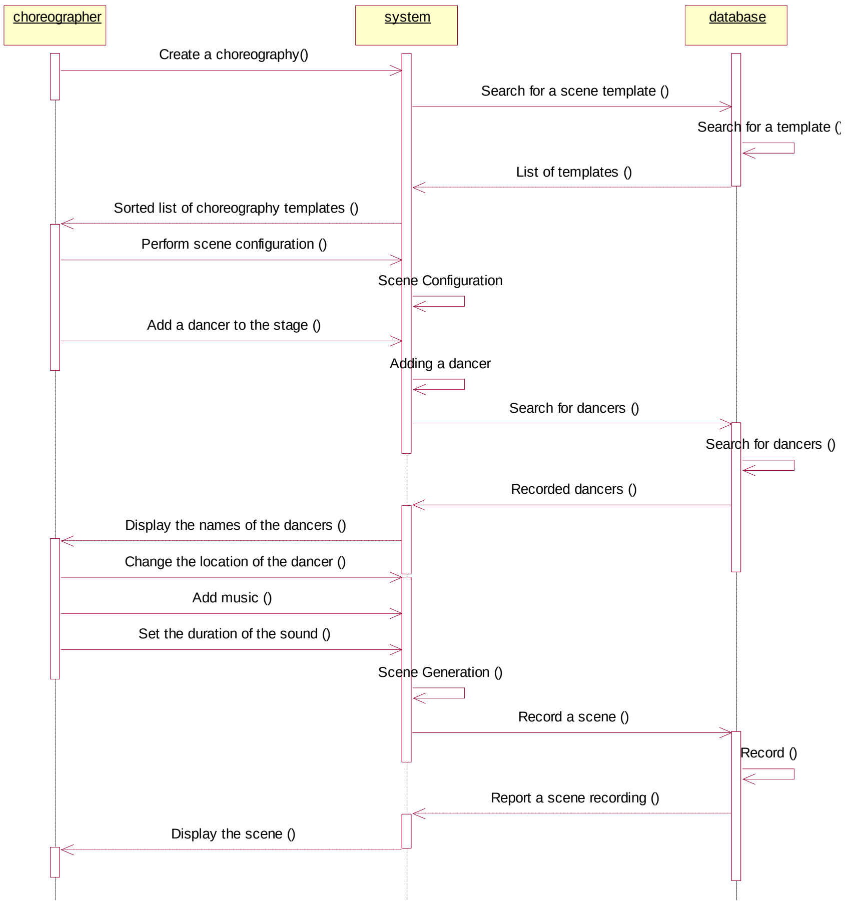


Fig.6 Sequence diagram

Building a Sequence Diagram. The sequence diagram is used to describe interactions between different elements of the model. This chart shows the interaction of time-ordered objects. The diagram depicts the premium user, the system (the application itself) and the database.
Formulation and justification of the problem. The purpose of the master's qualification work is to design and build an intelligent system that will enable to create and visualize choreographic productions. The main task is to provide the user with a set of features that will allow him to save the layout of the drawings in his choreographic setting as efficiently and quickly as possible. The system will also generate placement offers based on an analysis of the data collected. The offer system should increase the user's performance several times over its counterparts. The system will also allow the user to work with the music track and synchronize it with the pictures in the production. The main functions of the system are:

1. *Working with a list of choreographies:* adding new choreography; deletion of existing choreography; editing the name of the choreography; dissemination of choreography; quick preview of choreography drawings; filtering the list by name and date of last edit.
2. *Work with the list of blanks:* adding a new workpiece; removal of existing workpiece; filtering workpieces by the number of dancers in it.
3. *Logging the user into the system:* login with email and password; logging in using social networks Facebook or Instagram.
4. *Editing a single choreography image (working with a stage):* scene zooming and zooming; adding a new dancer to the scene grid; movement of the dancer within the stage; removal of the dancer from the production; taking the dancer outside the stage; display of dancers' names; operations on a group of dancers (movement, color change, deletion); cancel the last user action; add comments to the picture; creating a workpiece from selected dancers.
5. *Creating a transition between images*: creation of new positions of dancers; generation of optimal positions for dancers; suggesting the transformation of the current image into the next; create several variants of the following figure; set the transition interval relative to the music track.
6. *Revision of the whole production:* animated movements of dancers from previous positions to current ones; displays the current second of the music track.
7. *Editing production options:* resizing the scene; change the visual presentation of the scene; permutation of drawings in places.
8. *Working with the dancers in the production:* change of dancer's name; change the color of the dancer; display a list of dancers from previously created performances; tracking the movement of the selected dancer.
9. *Working with music*: choosing a music track; editing the interval for a particular image; choosing the best moments to change the drawings in the production.

Also, the non-basic functionality of the system may include localization in Ukrainian and English and stylization of the system. The proposed system provides efficient visualization of choreographic productions, which is guaranteed by the convenient placement of functions in the user interface and the workpiece function, which generates potential arrangements of dancers. None of the existing systems offer auto placement generation features, making the projected system a favorite among others. Moreover, the system offers synchronization with music, which is also a very popular user experience. Expected effects of system implementation:

- *labor productivity* - the system offers its layout options, which in most cases will accelerate the work of the choreographer and reduce the time spent on rendering productions. Also, the convenience and security of the system will provide the user with a pleasant user experience. The history of past choreographs can inspire the user for the following works;
- *organizational effect* - the choreographer can easily extend the created production with his team, which will allow all participants of the room to review their positions and be ready for rehearsal. It will also save time for the choreographer to find the optimal placement of the dancers on stage without even trying it out in practice. It will also allow dancers absent from rehearsal to be aware of events;
- *economic effect* - purchasing a premium account in the system to use additional features will be able to profit from purchases.

5 Choosing and Justifying the Methods for Solving the Problem

As a result of the analysis of the problem of visualization of choreographic productions in the previous sections, the relevance of the problem was proved, the existing systems for the solution of the problem were considered and the requirements of the intelligent system were formed, which will be the most effective for the user in comparison with other solutions. The user needs a system that can quickly and conveniently visualize his or her choreography. The best solution for this would be the mobile application, as it has already been discussed in the previous sections, it is a tool that is common to most users. To implement some of the functionality, you need a certain shared data repository that will process and store a large amount of data. This will introduce a server to collect data from devices and write them to a database. As the amount of data is expected to be large, it is decided to make the database distributed. The mobile application will contact the REST server, which is the most common solution among all. Specificity of the application requires that at every stage of development, the already working version of the system has been thoroughly considered by the end user. This is because the system requires maximum ease of use. Accordingly, each decision can affect the final convenience and impression of the user. To implement this method of solving the problem, it was decided to split the development of the system into several iterations, each of which will repeat certain processes. Before starting the first iteration, all system requirements will be collected and a specification created as specified in the second section. And in the process of iteration, the following processes will be implemented: formulation of requirements for the current version of the application; design creation; technical implementation of the system in accordance with established requirements and design; testing of a product with the involvement of real users who will potentially use it in the future. After the iteration, the shortcomings will be analyzed, the new functionalities will be selected and a new iteration will be started. This will maximize the efficiency of system development.

Also, at a stage when the system will have approximately similar functionality to analog products, it will be possible to boldly publish an application to the app store, as competitors' shortcomings will be taken into account and improved. This, even without additional functionality, will already attract the user.

Need for method of recommendations and suggestions. The system should include the functions of recommending and suggesting the arrangements of the dancers. Basically, a user can use blanks to generate ideas and quickly change layouts. The system will provide a large list of basic blanks, however, these blanks will not be contextual in certain choreographies, as each time the user can create something unique and not every time they come up with standard templates. That is why it is suggested to find a method that will generate its proposals based on the data that will be collected by all users of the system. It is also suggested to take into account the dynamics of the music and the wishes of the user. Thus, the user only needs to select a music track, specify the number of dancers in the production. With this input, the system will be able to generate the entire production, only confirming its decisions with the consent of the user. This approach to the production of productions is tremendously ahead of the systems that solve the problem of rendering productions, since none of them has yet implemented such an approach. Several methods of recommendation systems were considered: content-based, co-filtered, and hybrid, followed by cold start, scalability, interpretation, and exploitation/exploration.

Content-based method. At Pandora, the music distribution company, the musicians have labeled each song with over 400 attributes. Then, when the user was selecting a music station, songs that matched the station's attributes were added to his playlist. This is a content recommendation. Users or elements have profiles that describe their characteristics, and the system recommends the element to the user if the two profiles overlap. Stitch Fix Fashion Box is another example of a content recommendation. User attributes are collected (height, weight, etc.), and the corresponding fashion items are placed in a window presented to the user. Pandora requires manual effort / expense to create musical attributes, but there are many cases without such need. Stitch Fix clients provide their own benefits, LinkedIn users provide their own work experience and skills, Amazon sellers provide information about their products, all of whom are free to use content-based recommendations. A direct way of matching users and elements is to match keywords. For example, for job recommendations, you can agree on a job description with a summary of job seekers. Document Reverse Frequency is often used to give more weight to keywords unique to an element or user. A more systematic way is to build an observational model that evaluates the propensity of a user to like an invisible object. In a feature model, it is the attributes of users and elements (for example, a variable, whether a job and a job seeker in one industry), and a response variable - whether a user likes the item (for example, whether a job seeker would apply for a job). Content-based methods are computationally fast and interpretable. They can be easily adapted to new items or new users. However, some characteristics of elements / users may not be easy to accurately display or describe. Stitch Fix solved this by allowing machine learning to process structured data and human unstructured data (e.g., users' Pinterest pay).

Co-filtration method. Shared filtration systems provide recommendations based on historical users' preferences for items (click, view, purchase, like, rating, etc.). The advantage can be represented as a matrix of user elements (Fig. 7). Here is an example of a matrix describing the preference of 4 users over 5 items, where p_ {12} is the preference of user 1 over item 2.

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} \\ p_{41} & p_{42} & p_{43} & p_{44} & p_{45} \end{bmatrix}$$

Fig. 7. User Elements Matrix

Although the entries can be numeric, such as the task of predicting a Netflix movie rating (rating is 1 to 5), in most applications they are binary (eg, click, watch, purchase). In fact, the user element matrix can be more than millions * million (e.g. Amazon, Youtube), and most records are missing - the purpose of the recommendation systems is to fill in those missing entries (Fig. 8).

$$\begin{bmatrix} p_{11} & ? & p_{13} & ? & p_{15} \\ ? & ? & ? & p_{24} & ? \\ p_{31} & ? & p_{33} & ? & ? \\ ? & p_{42} & ? & p_{44} & p_{45} \end{bmatrix}$$

Fig. 8. The matrix of user elements of huge companies

There are three approaches to shared filtering: a closest neighbor and two methods for creating a new latent space: matrix factorization and deep learning.

Nearest neighbor. Neighbor-based methods are based on similarities between pairs of elements or users. The sloping similarity is often used to measure distance. The preference matrix can be represented as element vectors:

$$P = \begin{bmatrix} I_1 & I_2 & I_3 & I_4 & I_5 \end{bmatrix}$$

The similarity between I1 and I2 is calculated as cos (I1, I2). The matrix can also be represented as user vectors.

$$P = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}$$

The similarity between U1 and U2 is calculated as cos (U1, U2). Missing values in the preference matrix are usually filled with zeros. Subject-to-subject approaches are usually adapted in practice by Amazon (Amazon | 2013), Youtube (Youtube | 2012), Linkedin (Linkedin | 2018) and others. This approach can quickly process similar elements (similar elements for each element are pre-calculated and stored in the key data store). In addition, item recommendations may be more interpretable than user-to-user recommendations, for example, systems may explain why an item is recommended because "you liked X". There may be cases where the number of similar items is too small (after applying a threshold for similarity points). It would be possible to expand the list of similar items to include similar items (Youtube | 2012). Having obtained the most similar elements, it may be a useful step after processing. (Youtube | 2012) categorized similar items by video quality (e.g., measured by rating), diversity (e.g., by restricting one channel's recommendations), and user-specific (e.g., video-like video viewed more by a user, should be rated higher for the user). The three elements were combined with the linear model, providing the final rating.

Deep learning. Hidden Factor methods create a new, and usually smaller, space for the original vector of a user or object, resulting in noise reduction and faster real-time calculations. Deep learning is very flexible, including various factors in modeling and embedding. For example, deep learning was used to model sequential information using a bandwidth model originally used to calculate word similarity. (Airbnb | 2018, Zillow | 2018). Let's say the sequence of user elements is item1 -> item2 -> item 3 -> item4 ->… The intuition is to use each element in a sequence to predict adjacent elements, formulated as a classification problem, where each element is a single class. The training data includes the adjacent K elements of each element (left K and right K elements). The following figure (Fig. 9) illustrates pairs of objects with K = 1.

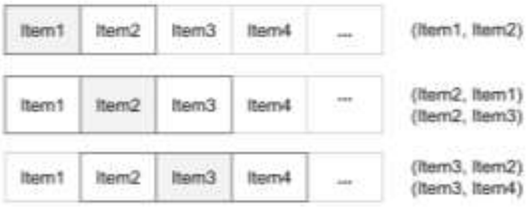


Fig. 9. The sequence of user elements

In addition, each element is represented as a hot vector having a length equal to the number of elements. The neural network takes the vector hot vector as input and outputs a vector of one of its similar elements, illustrated in the following figure, using (Item2, Item1) as a training example. A hidden layer is a new function space (or hidden space), and each element can be moved to a new function space by using the scales between the input layer and the hidden layer (essentially a linear combination of the original features). This diagram is shown in Fig. 10.

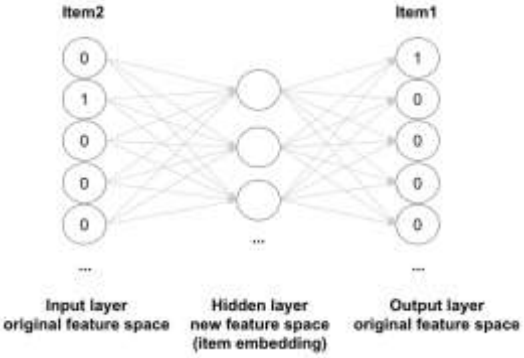


Fig.10. Scheme of the hidden layer

In fact, there can be millions of subjects, and billions of examples are used to train the network. A negative sampling idea can be used to simplify the calculation. The idea is to update only the weight of the original item (Item 1) and a small number of other items randomly selected. Next, we highlight the items and scales we need to update. This makes the calculation much faster (Fig. 11).

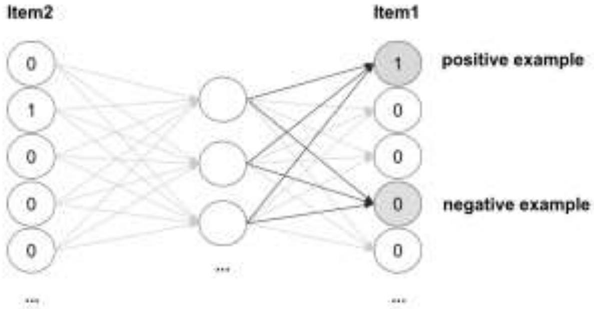


Fig. 11. Negative sampling idea

After each element is presented in a new function space, the similarity between the elements can be calculated and recommendations can be made based on similarity estimates. In some cases, users visit the funnel before converting, for example, an Amazon user makes a purchase after a pageview. An Airbnb user reserves a list after viewing several lists. This information can be included by adding the product purchased to each product training pair (Airbnb | 2018) shown in the figure below. Elements recommended in this way can improve your conversion rates.

- *Cold start.* Precise recommendations cannot be given to new users / items that have little or no information. This is called a cold start problem. This is a common problem for common filtering systems that rely on interaction between user elements. Some heuristics can be used. For the new user, you can recommend the most popular items in the user area. Some similarity criteria may be defined for a new element based on rules. For example, Airbnb used an average of 3 geographically closest lists of the same type and price range to navigate the new list (Airbnb | 2018).
- *Scalability.* Scalability is a key factor in determining what type of recommendation systems to use. More sophisticated systems need more people, potentially harder to hire, build / maintain, with greater equipment costs. This can be a long-term commitment, and therefore a business must understand the increase in business profits compared to the increase in costs. With that in mind, here are some of the basic elements of building scalable systems.

6 Choosing and Justifying the Means of Solving the Problem

There are currently two of the most common mobile app development platforms - iOS and Android. The advantages and disadvantages of both platforms were analyzed. As a result, the following categories were selected for comparison and the results of the study were obtained:

- *Quality of applications.* Apps almost always look more attractive and convenient on iPhone and iPad. Statistically, apps for the iOS platform will have more fans than those for the Android platform;
- *Quick update.* IPhone and iPad users don't have to wait for manufacturers to prepare updates after the new iOS release. Most users still haven't received Android 5.0, which Google announced last year. IOS updates are released regularly and are instantly available for all compatible devices;
- *Lasting support for old devices.* Apple mobile phone support is 48 months, which is not the case with Android smartphones;
- *The best programs are available first.* Most developers choose to first release the app on their iPhone and iPad and only after a while launch it on Android. This is due to the well-developed development tools for iOS;
- *Apple's ecosystem.* Today, when choosing a smartphone or tablet, the main thing is not different specifications such as battery capacity, camera resolutions, etc. They are more or less similar on all modern devices. The main thing for the user is the mobile ecosystem. And on iOS it is much more developed. In recent promotional material, Apple introduced its ecosystem as four key products - Apple Watch, MacBook, iPhone 6 and iPad Air 2. Apple TV and AirPort routers can still be added here.
- *Friendly interface.* The iPhone has become popular among successful people because it is convenient. People whose time is expensive make that choice because the iPhone saves time, hassle and focuses on what they want to do: connect with the right people, write messages, use social networks, and do it all seamlessly. Both the handsets and the iPhone software look great. Gadgets look appropriate in the hand of an entrepreneur, a student;
- *Reliability.* The iPhone is far more reliable as the manufacturer hones the nuances of producing one model and one operating system over the years, each upgrade making it more reliable, simpler, more powerful, and more productive. Scale sales of hundreds of millions of devices have allowed Apple to hone production technology and software to the smallest. Considering the above mentioned advantages of iOS platform and statistics of using platforms among choreographers, the iOS platform was unambiguously chosen.

To develop mobile applications for the iOS platform, there are two most popular development environments: xCode (developed by Apple) and AppCode (developed by JetBrains). Both options were analyzed and the main reasons for the advantages of each were identified:

- *Interdependence factor.* For the full functioning of the AppCode environment, some xCode tools are required. xCode contains all the tools you need to complete the development cycle for iOS mobile devices.
- *Storyboard.* The xCode integrated user interface tool is very user-friendly and allows you to quickly and without code create responsive application interfaces. AppCode does not support such tools and needs full code development.
- *Code refactoring and syntax highlighting*. AppCode provides the user with more tools to rewrite and improve code than xCode. However, the latest version of xCode has added a lot of enhancements to these tasks.
- *Price.* Xcode, unlike Appcode, is free. AppCode requires a monthly payment.
- Stability. The chances of unexpected technical problems during development in Xcode are minimal, which cannot be guaranteed in the case of AppCode.
- *User interface and settings.* The xCode design is very efficient, simple and allows the developer to accomplish their tasks quickly.

7 Choosing a Programming Language to Design

The following most common programming tools and languages are available for developing iOS applications: Swift or Objective-C. The following advantages of Swift over Objective-C can be distinguished: Swift is easier to read, easier to maintain, more secure, requires less code, faster [8]. Instead, Objective-C has some powerful Run-Time tools that have not yet been implemented in Swift. Also, many popular libraries that were written on Objective-C have not yet been copied to Swift, and without the knowledge of this language can not do in certain situations. Considering all the benefits of modern and progressive Swift, she was chosen to write this work.

- *Database selection*. The Realm mobile database was chosen to develop the application because it has such advantages as: quick familiarization, simple structure when adding to a project, simple flowchart, clear syntax, high speed and support for many platforms [9]. Apple's Core Data tool was considered as an alternative.
- *Choosing other means.* Cocoapods was selected to connect third-party libraries to the project. Fabric is a convenient tool for distributing iOS applications. Branch IO is a service for creating and generating deep links.

8 Creating an Application Architecture

Swift programming language, which is standard and native for development on iOS mobile platform, was used to develop the mobile application. The advantages of using it over other options were described in the previous section. Swift supports many programming paradigms, including imperative and declarative approaches. An imperative approach was chosen to implement the system because it is more reliable and easier to use.

MVC (Model-view-controller), which is standard and recommended by Apple, was chosen as the main architectural template. However, its standard implementation does not sufficiently separate the responsibility of the objects, which contributes to the emergence of classes responsible for many unrelated things. Because of this, the standard architecture structure was modified and adapted to the needs of the project. As a standard, this architectural approach contains 3 types of objects that divide all classes in the system into 3 groups:

- View is classes that are responsible for the user interface of the application. These classes contain information about placing graphic elements on the screen. Their main function is to display the data they receive in programmed and placed relative to each other. The view is also responsible for responding and transmitting user actions to the controller that processes them and performs the necessary actions.
- Model is classes that represent the business logic of the application. Here are all the model classes that represent the subject area of the program. Also, here is the logic of all modifications to objects in the system. Model classes are responsible for storing data on the system.
- Controller is classes, which is the interface between model and type. It handles all events and directs it where it is needed. These classes synchronize all input and output in the system. Often, these objects contain too much logic that other auxiliary objects can execute.

Communication between the view and the model using the controller can be represented in Fig. 12.

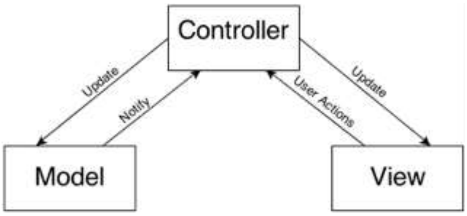


Fig. 12. Communication in the MVC template

The model notifies the controller of changes in its data, and the controller in response updates the data as. The view can then notify the controller of the actions taken by the user, either updating the model as needed, or retrieving data that the user needs.

All this separation is done to delineate the code and to reuse it. Ideally, the look should be completely isolated from the model. If it does not depend on the particular model implementation, it can be easily reused with another model to represent completely different data.

MVC on iOS. For starters, it is worth making sure that each class in the project is either a controller, a view, or a model. It is not necessary to combine the responsibilities of two different roles in one class. Next, you need to create three project groups to group the code, each for the appropriate category. One of the benefits of an MVC template is the ability to design the project independently, and then easily combine them. The diploma project will be divided into several components:

- lists that will be responsible for displaying and filtering choreographs;
- a scene that will contain all the logic of displaying dancers, their transitions and manipulations;
- audio is a separate component to separate all work with audio files.

9 Creating an Application Database

As a result of the work carried out and described in the previous section, a mobile Realm database was selected. This is a non-relational database that allows you to easily handle objects stored on disk as with normal objects in the program. Given the specifics of the software being developed, the database structure will not be complicated and will not contain a large number of links. To meet the needs of the user only combinations of basic properties will be enough to save the choreographic drawings that the user will create. The main entities will be choreography, template, dancer, drawing, stage configuration, position and background.

Choreography is stores information about the production, all the drawings it consists of, their order and transitions between them, the dancers, the music track and the scene refinement. Contains the following fields:

- Title - the name of the setting;
- date of creation;
- Date the last changes were made;
- Scene configuration;
- Id - a unique identifier for distinguishing choreography during distribution;
- Dancers - a list of all dancers participating in the production;
- Drawings - a list of all created drawings.

Template is saves information about one particular drawing without being tied to specific dancers or music. Contains the following fields:

- is the name of the template;
- Positions - a list of positions that form a picture.

Dancer is stores information about the dancer in the production. Contains the following fields:

- name - identifies the dancer; may be empty, if this is not significant for the choreographer;
- Color - the color to represent the dancer on stage.

Scene configuration is stores scene size information. Contains the following fields:

- Scene width (number of vertical lines);
- Scene length (number of horizontal lines);

- The size of the field on the sides of the stage;
- Check box to show or hide off-stage space.

All these dimensions are set in relative units, which will be displayed as one square of the scene. The stage itself is essentially a set of lines that form a rectangle and allow the dancers to be placed at the intersection of these lines.

Picture is- stores information about the arrangement of dancers on stage. Contains the following fields:

- positions - a list of all the positions where the dancers are placed;
- The duration of the transition to the next image is given in seconds;
- The name of the part of the production to which the picture relates.

Position is stores information about the placement of a particular dance on the stage.

- x and y - relative placement coordinates.

Musical accompaniment is stores the information needed to synchronize your images with music. Contains the following fields:

- a list of time frames indicating the time of a particular drawing;
- - The path in which the audio file will be stored.

All of the above entities will be represented as the descendants of the base class Object, which is basic when working with Realm and provides a link between the objects in the program and the stored data on the disk. Object allows you to define primary keys, indexed properties, ignored properties. At the same time, all of these classes form the database structure - each class is a separate table with a corresponding name and fields. The schematic of the database is presented in Fig. 13.

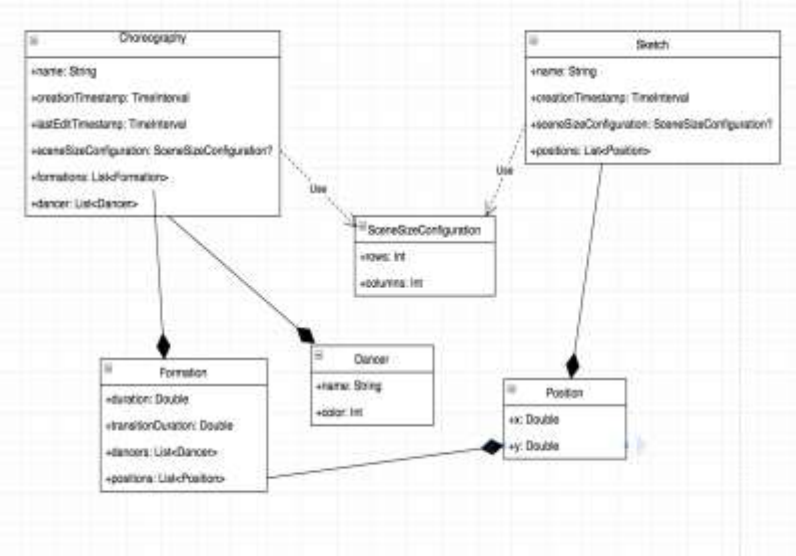

Choreography
+name: String
+creationTimestamp: TimeInterval
+lastEditTimestamp: TimeInterval
+sceneSizeConfiguration: SceneSizeConfiguration?
+formations: List<Formation>
+dancer: List<Dancer>
Sketch
+name: String
+creationTimestamp: TimeInterval
+sceneSizeConfiguration: SceneSizeConfiguration?
+positions: List<Position>
Use
Use
SceneSizeConfiguration
+rows: Int
+columns: Int
Formation
+duration: Double
+transitionDuration: Double
+dancers: List<Dancer>
+positions: List<Position>
Dancer
+name: String
+color: Int
Position
+x: Double
+y: Double

Fig. 13. Database schema

10 Implementation of the Main Classes of the System

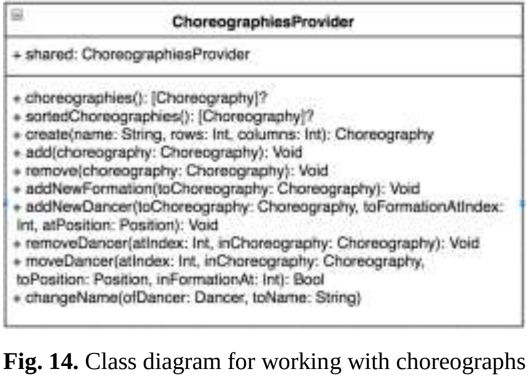
Each model class described in the design section is a descendant of the base class of the Realm - Object database. Object allows you to handle data stored on disk as simple objects. A common RealmService class manager has been created to unify all database objects. This is the wrapper by which the database is accessed. If you need to move to another database, you will simply need to override the logic of all methods in this class. Template methods allow you to apply basic operations to any types that are descendants of Object. These methods include the following:

- add() - create / update row / rows in table;
- update() - updating table row specific properties;
- Delete() - delete row;
- removeAll() - deleting table;
- write() - record any changes to the table by changing the properties of the objects.

Another responsibility of this class is the initial configuration of the database and the migration between different versions of the database schema.

11 Working with Choreographies

The specificity of the Realm database is such that it is enough to get objects of a certain type only once, which will be stored in the Results structure <object type>, after which all changes, additions or deletions will be made and this structure will be actual in any -what time. With this feature in mind, a ChoreographiesProvider class was created to access choreography manipulation from anywhere in the program. The RealmService class described above is used for manipulation. Also, this class provides an opportunity to get a list of all the choreographies and sort this list by a specific property. Among the available operations for choreography are the following methods: adding a new image, adding a new dancer, changing the position of a dancer in a particular drawing, changing the name and color of the dancer, deleting the picture, inserting a sub-picture.



```
ChoreographiesProvider
+ shared: ChoreographiesProvider
+ choreographies(): [Choreography]?
+ sortedChoreographies(): [Choreography]?
+ create(name: String, rows: Int, columns: Int): Choreography
+ add(choreography: Choreography): Void
+ remove(choreography: Choreography): Void
+ addNewFormation(toChoreography: Choreography): Void
+ addNewDancer(toChoreography: Choreography, toFormationAtIndex:
Int, atPosition: Position): Void
+ removeDancer(atIndex: Int, inChoreography: Choreography): Void
+ moveDancer(atIndex: Int, inChoreography: Choreography,
toPosition: Position, inFormationAt: Int): Bool
+ changeName(ofDancer: Dancer, toName: String)
```

Fig. 14. Class diagram for working with choreographs

12 Implementation of Scene Logic

The central element that implements the application's business logic is the stage, which is used both to display choreography, dancers and all transitions between drawings, and to create productions directly. The logic of the scene was divided into separate components, namely:

- visual display;
- component for calculating scene size;
- component for placing dancers on stage;
- a component for dragging a dancer across a stage.

A descendant of the class UIView - SceneView was created to display the scene and the logic of drawing lines was implemented. The input provides the scene size configuration and the screen size of the device. The result is a path containing a set of horizontal and vertical lines that are spaced at the same interval from each other, forming a certain number of squares corresponding to the desired scene size. To place the dancers on stage, a separate class -SceneUnitLocator was implemented, which receives a point of contact with the stage that is relative to the screen size of the device. This point is converted to the scene coordinates, and then the nearest point where the dancer can be placed is calculated. There are 2 options for calculating the nearest point on a stage: the intersection of the line or the middle of the line. If the point is valid and is within the stage, then a new dancer is added as a round point.

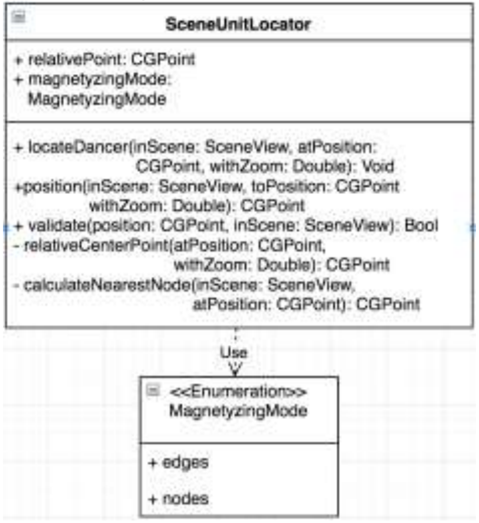
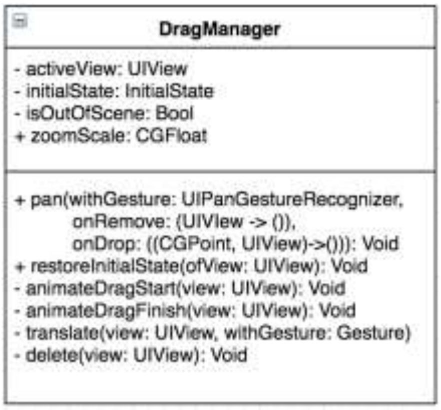

SceneUnitLocator
+ relativePoint: CGPoint
+ magnetyzingMode: MagnetyzingMode
+ locateDancer(inScene: SceneView, atPosition: CGPoint, withZoom: Double): Void
+position(inScene: SceneView, toPosition: CGPoint withZoom: Double): CGPoint
+ validate(position: CGPoint, inScene: SceneView): Bool
- relativeCenterPoint(atPosition: CGPoint, withZoom: Double): CGPoint
- calculateNearestNode(inScene: SceneView, atPosition: CGPoint): CGPoint
Use
<<Enumeration>>
MagnetyzingMode
+ edges
+ nodes

Fig. 15. Class diagram for placing dancers on stage

To change the position of the dancer on the stage was implemented DragManager class. After touching the dancer, his performance is animated. While dragging the finger position of the user's finger, the dancer also changes his position accordingly. If outside the stage, the user is asked to remove the selected dancer from the production. If the user releases the dancer within the scene, then the closest point is calculated according to the same logic as when adding the dancer. This class supports dragging many dancers at the same time.

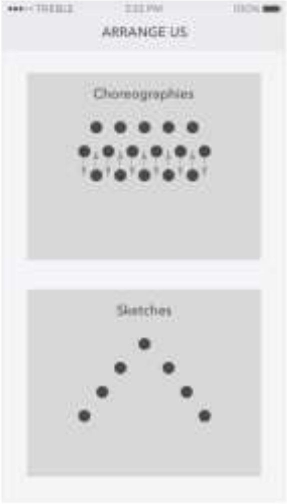


```
DragManager
- activeView: UIView
- initialState: InitialState
- isOutOfScene: Bool
+ zoomScale: CGFloat

+ pan(withGesture: UIPanGestureRecognizer,
        onRemove: (UIView -> ()),
        onDrop: ((CGPoint, UIView)->())): Void
+ restoreInitialState(ofView: UIView): Void
- animateDragStart(view: UIView): Void
- animateDragFinish(view: UIView): Void
- translate(view: UIView, withGesture: Gesture)
- delete(view: UIView): Void
```

Fig. 16. Class diagram for dragging dancers on stage

There is a class in which all the logic of drawing a scene and dancers on it takes place, but in this case the size of the grid is much smaller, since the generated image is used only as a preview.



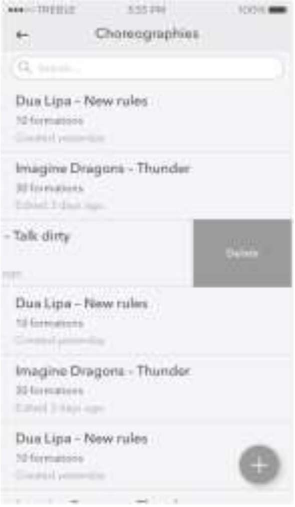


Fig. 17. Main menu **Fig. 18.** List of choreographies

13 Classes for Displaying Graphic Information

7 classes were created, 5 of which are inherited by UITableViewCell and 2 by UICollectionViewCell. The inheritance of class data allows you to create your own prototypes with the required functionality and basic functionality that is inherited from the corresponding classes. For example, the awakeFromNib () method can be overridden

by calling super.awakeFromNib () for its basic implementation, and then defining all the necessary settings that will be made when creating a cell instance.

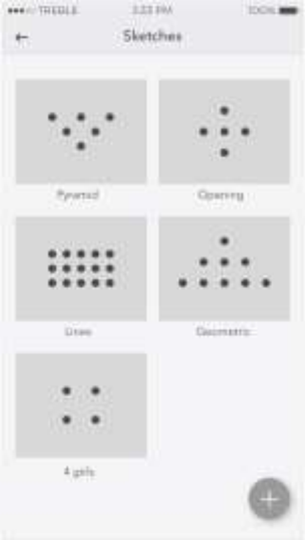


Fig. 19. List of templates

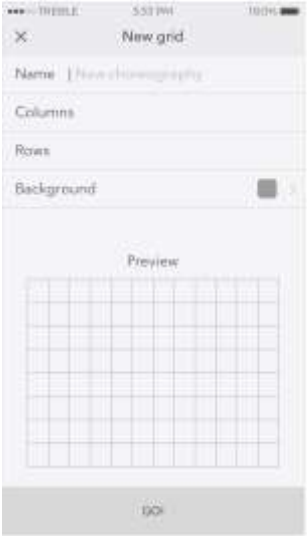


Fig. 20. Scene configuration

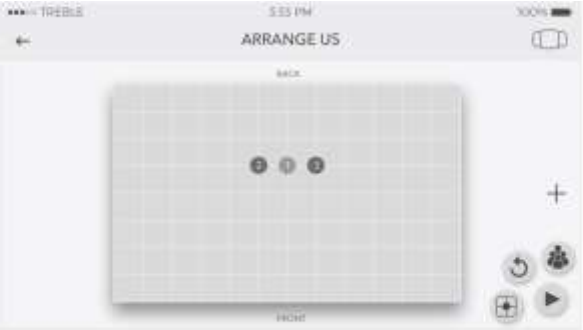


Fig. 21. Scene view

Classes inherited from UITableViewCell:

- ChoreographyTableViewCell is displays general choreography information among other choreographies in the list, such as name, number of drawings, and date of the last change.
- EmptyTableViewCell is a universal cell whose purpose is to show that the result table is still empty and you need to create something to display it.
- NameTableViewCell is displays information about the dancer in the production, such as name, sequence number, which appears inside the corresponding point on the stage, and color. Color and name can be edited right inside the cell.

- TextFieldTableViewViewCell is contains a text box that can be used anywhere where you need to enter some information.
- SelectNumberTableViewCell is contains two buttons to decrease or increase the current number. Used to configure scene size.

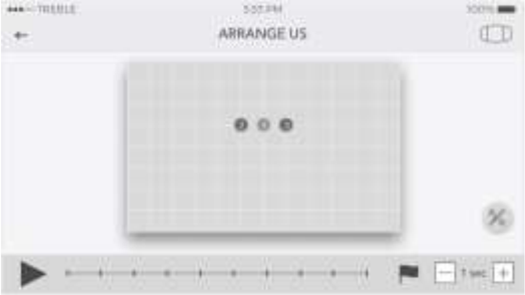


Fig. 23. View mode

Classes inherited from UICollectionViewCell:

- SketchCollectionsCell is displays a smaller version of the template and its name.
- ColorCollectionCell is displays a color variant to select and an indicator indicating the currently selected color.

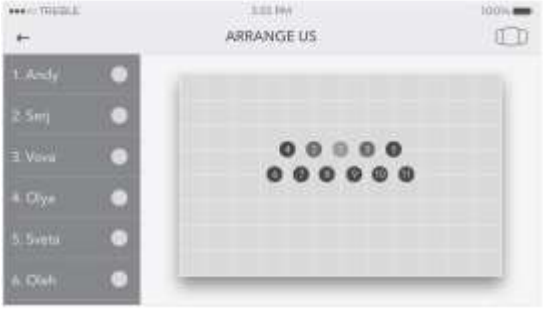


Fig. 24. Editing dancers

Classes for displaying graphic information

- *Edge Structure*. Contains the size of the side and the base points along the X and Y axes. Having these two properties generates an array of points describing all the end points of the square containing the base point. This structure makes it easy to work with the stage in determining the closest point to where you need to move the dancer.
- *StoryboardIdentifiable protocol*. The main objective of this protocol is to ensure that all visual elements of the project will have identifiers that match their names. This will reduce the number of errors and make the handling of these elements more convenient.

- *The Utils class*. Contains all support methods for displaying error messages to the user. Allows you to configure these messages differently, including the ability to transmit a code block to respond to messages.
- *StyleManager and Constants classes*. The only places in the project where all the constants and colors are stored. The advantage of these classes is that when you change a ribbon or color that is used many times, it will only need to be changed in one place in the code.

14 Conclusion

Considering that, the paper discusses the methods and means by which an intelligent system of visualization of choreographic productions will be implemented. IOS mobile system was chosen for development. The mobile application itself will be developed by the language. Swift programming in xCode environment. Additional tools for system development were considered and the best ones were chosen to solve potential problems. Methods for the functioning of the system of proposals and the generation of new placements were also analyzed. Shared filtering was chosen among all the methods considered. The developed mobile application was named “Locate us”. To run the program, you must have a device running iOS 12.0 or higher. You can also run the application on a tablet with a similar operating system.

References

1. Rzheuskyi, A., Gozhyj, A., Stefanchuk, A., Oborska, O., Chyrun, L., Lozynska, O., Mykich, K., Basyuk, T.: Development of Mobile Application for Choreographic Productions Creation and Visualization. In CEUR Workshop Proceedings, vol.2386, pp.340-358 (2019)
2. Abraham, A., Dunsky, A., Dickstein, R. The effect of motor imagery practice on elevé performance in adolescent female dance students: a randomized controlled trial. J. Imag. Res. Sport. Phys. Act. 12, (2017)
3. Chappell, K., Craft, A., Rolfe, L., Jobbins, V. Dance partners for creativity: choreographing space for co-participative research into creativity and partnership in dance education. Res. Dance Educ. 10, 177–197. (2009)
4. Collet, C., Guillot, A., Lebon, F., MacIntyre, T., Moran, A. Measuring motor imagery using psychometric, behavioural, and psychophysiological tools. Exerc. Sport Sci. Rev. 39, 85–92.(2011)
5. Cross, E. S., Acquah, D., Ramsey, R. A review and critical analysis of how cognitive neuroscientific investigations using dance can contribute to sport psychology. Int. Rev. Sport Exerc. Psychol. 7, 42–71.(2014)
6. Fink, A., Woschnjak, S. Creativity and personality in professional dancers. Pers. Indiv. Diff. 51, 754–758. (2011)
7. Pavlik, K., Nordin-Bates, S. Imagery in dance: a literature review. J. Dance Med. Sci. 20, 51–63. (2016)
8. Torrents, C., Ric,A., Hristovski, R. Creativity and emergence of specific dance movements using instructional constraints. Psychol. Aesth. Creat. Arts 9:65. (2015).

9. Rumbaugh, J., Jacobson, I., Booch, G. The Unified Modeling Language Reference Manual. Addison-Wesley Professional; 2 edition. 752p. (2004)
10. Vasilevskis, E., Dubyak, I., Basyuk, T., Pasichnyk, V., Rzheuskyi, A.: Mobile Application for Preliminary Diagnosis of Diseases. In CEUR Workshop Proceedings, vol.2255, pp.275-286 (2018)
11. Chen, J., Dosyn, D., Lytvyn, V., Sachenko, A.: Smart Data Integration by Goal Driven Ontology Learning. In: Advances in Big Data, 283-292. (2016)
12. Zdebskyi, P., Vysotska, V., Peleshchak, R., Peleshchak, I., Demchuk, A., Krylyshyn, M.: An Application Development for Recognizing of View in Order to Control the Mouse Pointer. In: CEUR Workshop Proceedings, Vol-2386: 55-74. (2019)
13. Chyrun, L., Burov, Y., Rusyn, B., Pohreliuk, L., Oleshek, O., Gozhyj, A., Bobyk, I.: Web Resource Changes Monitoring System Development. In: CEUR Workshop Proceedings, Vol-2386: 255-273. (2019)
14. Gozhyj, A., Chyrun, L., Kowalska-Styczen, A., Lozynska, O.: Uniform Method of Operative Content Management in Web Systems. In: CEUR Workshop Proceedings, Vol-2136: 62-77. (2018)
15. Veres, O., Rusyn, B., Sachenko, A., Rishnyak, I.: Choosing the Method of Finding Similar Images in the Reverse Search System. In: CEUR Workshop Proceedings, Vol-2136: 99-107. (2018)
16. Basyuk, T., Vasyliuk, A., Lytvyn, V.: Mathematical Model of Semantic Search and Search Optimization. In: CEUR Workshop Proceedings, Vol-2362: 96-105. (2019)
17. Lytvyn, V., Peleshchak, I., Vysotska, V., Peleshchak, R.: Satellite spectral information recognition based on the synthesis of modified dynamic neural networks and holographic data processing techniques. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 330-334. (2018)
18. Kazarian, A., Kunanets, N., Pasichnyk, V., Veretennikova, N., Rzheuskyi, A., Leheza, A., Kunanets, O.: Complex Information E-Science System Architecture based on Cloud Computing Model. In: CEUR Workshop Proceedings, 2362: 366-377. (2019)
19. Basyuk T.: The Popularization Problem of Websites and Analysis of Competitors. In: Advances in Intelligent Systems and Computing, 689: 54-65. (2018)
20. Basyuk, T.: Innerlinking website pages and weight of links. In: Computer science and information technologies (CSIT), 12-15. (2017)
21. Basyuk, T.: Popularization of website and without anchor promotion. In: Computer science and information technologies (CSIT), 193-195. (2016)
22. Emmerich, M., Lytvyn, V., Yevseyeva, I., Fernandes, V. B., Dosyn, D., Vysotska, V.: Preface: Modern Machine Learning Technologies and Data Science (MoMLeT&DS-2019). In: CEUR Workshop Proceedings, Vol-2386. (2019)
23. Lytvyn, V., Sharonova, N., Hamon, T., Cherednichenko, O., Grabar, N., Kowalska-Styczen, A., Vysotska, V.: Preface: Computational Linguistics and Intelligent Systems (COLINS-2019). In: CEUR Workshop Proceedings, Vol-2362. (2019)
24. Vysotska, V., Chyrun, L.: Methods of information resources processing in electronic content commerce systems. In: Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2015-February. (2015)
25. Lytvyn, V., Vysotska, V., Peleshchak, I., Rishnyak, I., Peleshchak, R.: Time Dependence of the Output Signal Morphology for Nonlinear Oscillator Neuron Based on Van der Pol Model. In: International Journal of Intelligent Systems and Applications, 10, 8-17. (2018)

26. Vysotska, V., Hasko, R., Kuchkovskiy, V.: Process analysis in electronic content commerce system. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 120-123. (2015)
27. Lytvyn, V., Vysotska, V.: Designing architecture of electronic content commerce system. In: Computer Science and Information Technologies. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 115-119. (2015)
28. Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., Rishnyak, H.: The Risk Management Modelling in Multi Project Environment.. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 32-35. (2017)
29. Kanishcheva, O., Vysotska, V., Chyrun, L., Gozhyj, A.: Method of Integration and Content Management of the Information Resources Network. In: Advances in Intelligent Systems and Computing, 689, Springer, 204-216. (2018)
30. Naum, O., Chyrun, L., Kanishcheva, O., Vysotska, V.: Intellectual System Design for Content Formation. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 131-138. (2017)
31. Vysotska, V., Fernandes, V.B., Emmerich, M.: Web content support method in electronic business systems. In: CEUR Workshop Proceedings, Vol-2136, 20-41. (2018)
32. Vysotska, V., Lytvyn, V., Burov, Y., Gozhyj, A., Makara, S.: The consolidated information web-resource about pharmacy networks in city. In: CEUR Workshop Proceedings, 239-255. (2018)
33. Gozhyj, A., Kalinina, I., Vysotska, V., Gozhyj, V.: The method of web-resources management under conditions of uncertainty based on fuzzy logic. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 343-346. (2018)
34. Gozhyj, A., Vysotska, V., Yevseyeva, I., Kalinina, I., Gozhyj, V.: Web Resources Management Method Based on Intelligent Technologies. In: Advances in Intelligent Systems and Computing, 871, 206-221. (2019)
35. Vysotska, V., Chyrun, L.: Analysis features of information resources processing. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 124-128. (2015)
36. Vysotska, V., Chyrun, L., Chyrun, L.: Information Technology of Processing Information Resources in Electronic Content Commerce Systems. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 212-222. (2016)
37. Vysotska, V., Rishnyak, I., Chyrun L.: Analysis and evaluation of risks in electronic commerce. In: CAD Systems in Microelectronics, 9th International Conference, 332-333. (2007)
38. Vysotska, V., Burov, Y., Lytvyn, V., Oleshek, O.: Automated Monitoring of Changes in Web Resources. In: Advances in Intelligent Systems and Computing, 1020, 348–363. (2020)
39. Lytvyn, V., Vysotska, V., Mykhailyshyn, V., Rzheuskyi, A., Semianchuk, S.: System Development for Video Stream Data Analyzing. In: Advances in Intelligent Systems and Computing, 1020, 315-331. (2020)
40. Lytvyn, V., Vysotska, V., Shakhovska, N., Mykhailyshyn, V., Medykovskyy, M., Peleshchak, I., Fernandes, V. B., Peleshchak, R., Shcherbak, S.: A Smart Home System Development. In: Advances in Intelligent Systems and Computing IV, 1080, 804-830. (2020)
41. Lytvyn, V., Kowalska-Styczen, A., Peleshko, D., Rak, T., Voloshyn, V., Noennig, J. R., Vysotska, V., Nykolyshyn, L., Pryshchepa, H.: Aviation Aircraft Planning System Project Development. In: Advances in Intelligent Systems and Computing IV, Springer, Cham, 1080, 315-348. (2020)