|  |
| --- |
| Win-interactive |
| Puzzle game |
| Integration Specifications |
|  |
| **VWMT team** |
| **29-Nov-13** |

**Revision History:**

|  |  |  |
| --- | --- | --- |
| Date | Author | Comments |
| Nov/29/13 | O. Tarandevich  O. Gibaev | Document created |

Contents

[Game Flow 3](#_Toc373609820)

[Game flow diagram 4](#_Toc373609821)

[Fringes 4](#_Toc373609825)

[Technical Integration Information 6](#_Toc373609826)

# Game Flow

Andrew Removsky has proposed the following scenario (rus):

Игра Puzzle – конструирование картины

Имеется множество “мест”, в которые помещаются фрагменты некоторой картины (рисунка).

Имеется также множество фрагментов, которые должны быть помещены в эти “места”.

Иными словами, из фрагментов конструируется картина. (Изображение этой картины также обозревает тот, кто её “конструирует”). Количество фрагментов равно количеству “мест”. Фрагменты представлены игроку в случайном порядке. Игрок выбирает фрагмент и указывает место, в которое, по его мнению, следует поместить фрагмент.

Если игрок указал место правильно, то фрагмент помещается на “своё” место. И этот фрагмент удаляется из совокупности предъявляемых далее фрагментов. Если же нет – регистрируемое количество ошибок увеличивается на 1. При этом игрок может указать новое “место” для этого ж фрагмента, выбрать другой фрагмент либо отменить свой выбор (например, указав область не являющуюся ни фрагментом, ни “местом”).

Игра оканчивается, когда картина (рисунок) полностью сконструирована либо игрок сдался.

Комментарий: VWML описание игры не должно зависеть от количества фрагментов. Количество фрагментов картины описывается в интеграционной части.

## Game flow diagram

## 

Is the place correct for the selected image?

Select Place

Select Image

Initialization

Notify External World

No

Error++

## 

## 

Yes

# Fringes

No

Yes

Notify External World: Complete

All the places filled?

Remove the image from the active images stack

There are 3 “fringes” in the system:

1. Config initialization

- Conf: Load images and places configuration

This is input-type of fringe: loading of the list of images IDs, places IDs and relations between images and places (see the table below).

1. User action fringe (UserAction):

* SelectImage: select image.
* SelectPlace: select place for the selected image.

This is input-type of fringe: notifies model about image selection and place selection.

1. Turn result (GameNotify):

* Success: the place is correct for the image.
* Failure: the place is incorrect for the image.
* GameComplete: all the places are filled.

This is output-type of fringe: notifies external world about the turn result and the end of the game.

The following table specifies the fringes:

|  |  |  |  |
| --- | --- | --- | --- |
| Fringe name | Method name | Parameters | Description |
| Conf | GetImagesList | ImagesList | Input. Initialization request. Receives the list of images IDs. |
| Conf | GetPicturesList | PlacesList | Input. Initialization request. Receives the list of places IDs. |
| Conf | Get Correspondences | GetCouples | Input. Initialization request. Receives the list of correspondences table (image->place). |
| UserAction | SelectImage | ImageID | Input. Receives the selected image ID by the user. |
| UserAction | SelectPlace | PlaceID | Input. Receives the selected place ID by the user. |
| GameNotify | TurnResult | Success, ErrorsCount | Output. Sends the result of turn: True, errors count for the moment. |
| GameNotify | TurnResult | Failure, ErrorsCount | Output. Sends the result of turn: False, errors count for the moment. |
| GameNotify | GameComplete | ErrorsCount | Output. Sends: overall errors count for the game session. |

# Technical Integration Information

This section should contain the following information:

1. Modules which are passing to the «external world».
2. How the model should be handled by java-code.
3. Instructions about initialization and handling the jar-module.