

Проекти по RIAAFLEX

летен семестър 2008

Съдържание

Clickomania	2
Не се сърди човече	3
SandGarden	4
Flex-фотогалерия	5
Карта от изображения	8
Електронен магазин	10
Тетрис	13
Табла	14
Дама	15
Белот	16
Sokoban	17
Snake	18

проект	максимална големина на екипа по курсове			
	1 курс	2 курс	3 курс	4 курс
Clickomania	4	3	2	2
Не се сърди човече	4	4	3	3
SandGarden	4	3	2	2
Flex-фотогалерия	6	5	5	5
Карта от изображения	5	5	5	5
Електронен магазин	5	5	4	4
Тетрис	5	4	3	3
Табла	5	4	3	3
Дама	4	3	2	2
Белот	5	4	3	3
Sokoban	5	4	3	3
Snake	3	3	2	2

Clickomania

курс	макс. големина екип
1	4
2	3
3	2
4	2

Играта представлява правоъгълна мрежа ($m \times n$), чиито елементи са оцветени с един от k на брой цвята или са „празни“. Първоначално всички елементи са оцветени.

Целта на играча е да сведе броя на квадратчетата до минимум. Квадратчета могат да се „унищожават“, ако имат поне един съсед от същия цвят. Ако играчът реши да унищожи дадено квадратче, се премахват всички съседни по вертикала и хоризонтала квадратчета на посоченото, които са осветени със същия цвят. Новополучените „празни места“ се запълват от по-горните пълни квадратчета, а ако имаме празна колона - от дясно стоящите до нея. Стартовата конфигурация да се зарежда от локалния компютър, ако не е достъпен (или не съществува), да се създаде с произволно начално оцветяване.

Да се съхраняват големината на табло и най добрите резултати с имена на съответните играчи.

За повече информация:

<http://www.clickomania.ch/click/index.html>

<http://www.clickomania.ch/click/pix/Clickoani.gif>

Не се сърди човече

курс	макс. големина екип
1	4
2	4
3	3
4	3

Да се реализира добре познатата игра. Да се даде възможност на потребителя за съхраняване/зареждане на игра. Да се направи възможна играта срещу компютър, като не се набляга на интелекта му.

SandGarden

курс	макс. големина екип
1	4
2	3
3	2
4	2

Дадена е правоъгълна карта с размер $m \times n$, Играчът стартира с позиция извън картата, като целта му е до премине през всяко квадратче от правоъгълния терен точно по 1 път. Квадратчетата могат да бъдат проходими и непроходими, като началната конфигурация за зарежда от файл.

Придвижването се извършва по следния начин. При влизане в карта се използва направлението на самото влизане и движението продължава до излизане от картата или до достигане на вече непроходимо квадратче. При такова квадратче се прави избор за промяна на посоката като избора ни определя новата посока, и извършваме движението по вече известните правила. Всяко посетено квадратче се маркира като непроходимо. Да се предостави възможността за UNDO. Също така и възможност за отваряне на различни градини от XML файлове.

За повече информация може да се погледнете:

<http://www.gametunnel.com/gamespace.php?id=225&tab=3>

Flex-фотогалерия

курс	макс. големина екип
1	6
2	5
3	5
4	5

Да се реализира FLEX или AIR приложение, което да консумира публична услуга (напр. Flickr), предоставяща изображения и информация за тези изображения (напр. автор, дата, коментари, тагове и т.н.).

Изискванията към приложението:

1. Да изобразява списък от изображения в някоя от списъчните контроли на Flex или в такава custom data-driven контрола.
2. Да може да визуализира детайлна информация за избраното изображение от списъчната контрола.
3. Да се извършва филтриране върху показаните изображения по зададен от потребителя критерий.
4. Смяната на изгледите на приложението трябва да става без потребителят да загубва текущият потребителски контекст, т.е. смяната на изгледите трябва да бъде извършена чрез смислени трансформации.

Изисквания към реализацията:

Приложението трябва да е съставено от композитни MXML-компоненти, чиито данни се зареждат посредством data binding, където употребата на data binding е смислена. Компоненти трябва да използват states и transitions, където е необходимо. Изисква се поне една употреба на states и transitions. Една добра реализация на това приложение ще използва смислено разделение между модел, изгледи и контролна логика (MVC). Допълнителен бонус носи ActionScript 3 реализация на custom контрола, наследяваща UIComponent или някой негов наследник от Flex SDK - имплементацията на такъв компонент, трябва да покаже разбиране на инвалидационният механизъм, който ползват стандартните контроли във Flex. Допълнителен бонус носи модифицирането на външният вид на контролите чрез skinning и styling. Допълнителен бонус носи имплементирането на потребителски интеракции, характерни за desktop-приложенията (напр. drag-and-drop, mouse wheel zoom и т.н.).

Допълнителна информация:

1. Имплементация на приложение със сходни изисквания, може да намерите тук: <http://flex.riabg.org/?p=4>

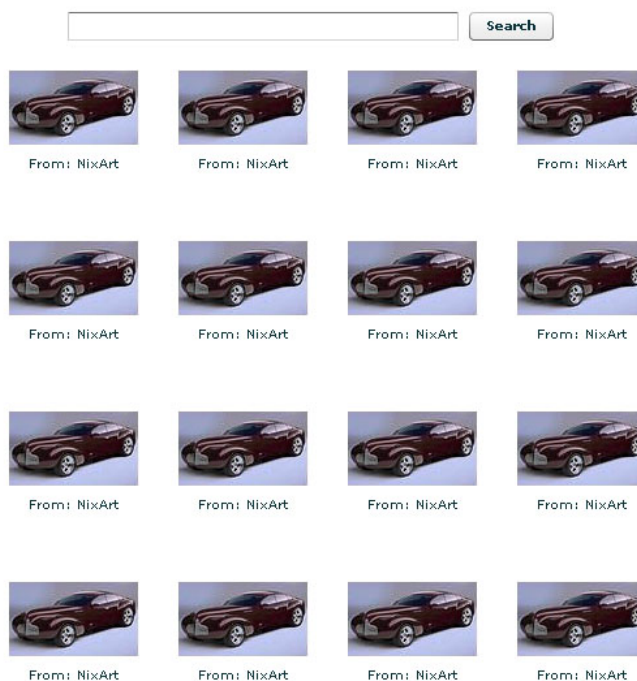
2. За въпроси към заданието пишете на npacemo@gmail.com, или на някой от адресите на преподавателите.

3. За въпроси, свързани с Flex, използвайте: <http://groups.google.com/group/flexbulgaria>

Примерни изгледи:

Предоставените изгледи са напълно ориентировъчни.

1. Списъчен изглед.



2. Изглед върху детайлите на избрано изображение.

Volvo 3CC Concept Car



Tags

NAIAS
Detroit Auto Show
Volvo
Volvo 3CC
Volvo Concept Car
Concept Car
Automotive Design
Automotive Styling
Volvo Concepts

The Volvo 3CC Concept Car as seen at the North American International Auto Show in Detroit Michigan in 2005. Electric powered 3 seater.

Comments



lisa721 [deleted] says:

Cars are getting to be so cool.



lisa721 [deleted] says:

Cars are getting to be so cool.

Карта от изображения

курс	макс. големина екип
1	5
2	5
3	4
4	4

Да се направи Flex или AIR приложение, което да консумира публична услуга (напр. Flickr), предоставяща изображения, разполага ги в една равнина и реализира pan и zoom функционалности за навигация, подобно на интерактивна географска карта.

Изискванията към приложението:

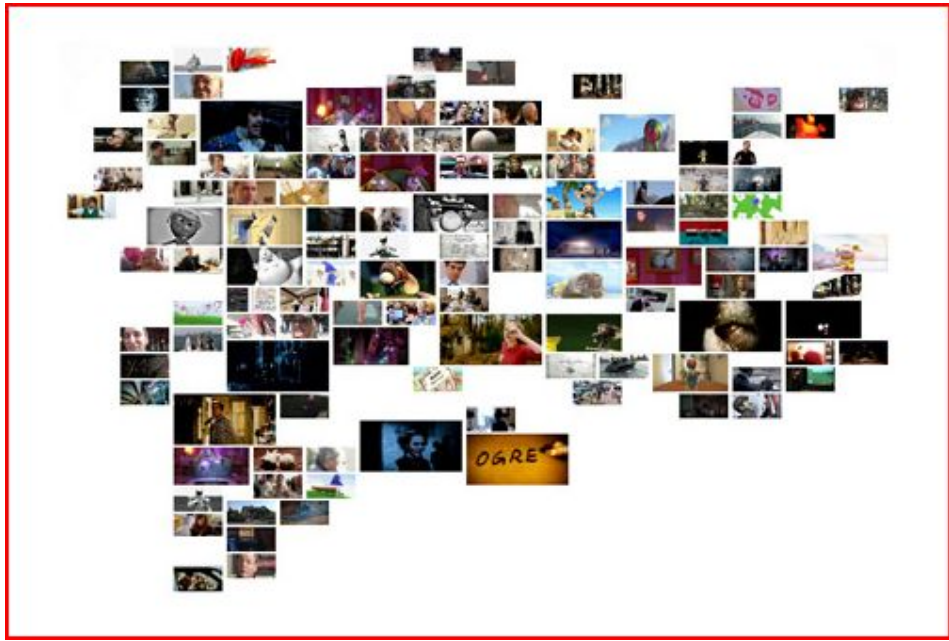
1. Да зарежда динамично изображения от публична услуга. API-та за работа с Flickr може да намерите на адрес: <http://www.flickr.com/services/api/> в секцията за ActionScript.
2. Да разполага заредените изображения по случаен начин в равнина. Задължително е изображенията да не се застъпват. За предпочитане е да се разполагат около центъра, но не плътно, т.е. да се оставят празни места за по добър визуален ефект (виж т. 3 от допълнителната информация). Да се използват thumbnail версиите на изображенията, защото се предполага броя им е да сравнително голям (100 - 500).
3. Да се реализира pan и zoom функционалност. Потребителят трябва да може да навигира по XY чрез panning, както и по Z чрез zoom in и zoom out. За предпочитане е и двете действия да е възможно да се извършват както с клавиатурата, така и с мишката.
4. Бонус носи реализацията на контрола, в която потребителя да задава ключови думи, по които да се търсят изображенията, както и максималния брой на резултатите.
5. Бонус носи реализация на детайлен изглед на изображение. Например при click върху изображение да се отваря прозорец с информация за изображението (име, автор и др.), както и неговата оригинална (hi-resolution) версия.

Бонус се дава за реализация на всяка от следните оптимизации на производителността:

1. По време на panning картата да се заменя с bitmap копие.
2. Да се ограничи броя на паралелно зареждащите се изображения. Например да се разделят на порции от по 5 и всяка порция да се пуска след като е приключила предишната.

Допълнителна информация:

Примерен изглед на картата:



Електронен магазин

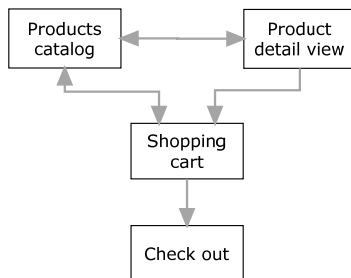
курс	макс. големина екип
1	5
2	5
3	4
4	4

Общо функционално описание

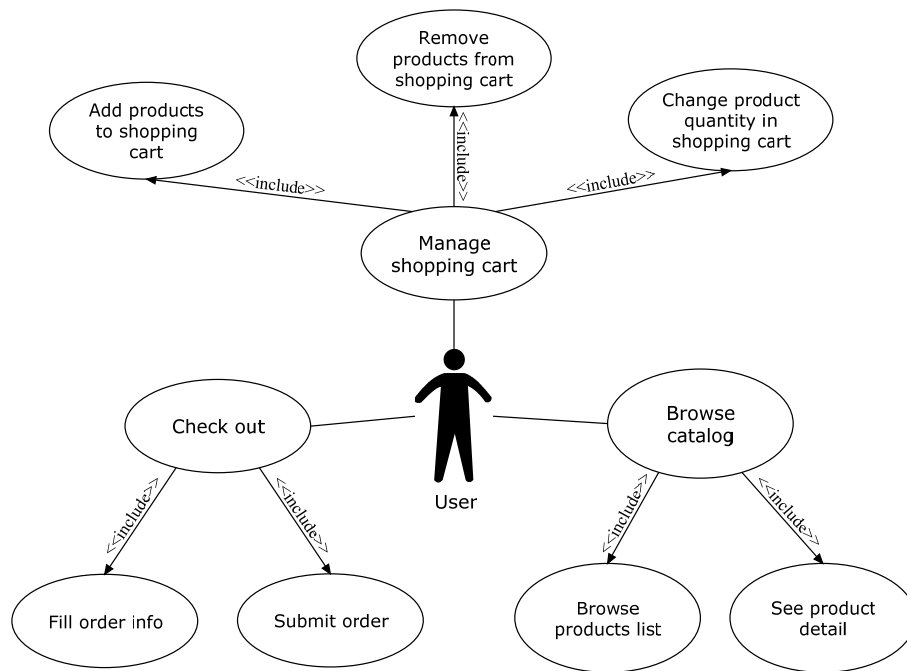
Приложението предоставя възможност за разглеждане на каталог с продукти, избор на продукт(и) от потребителя и закупуването им чрез попълване на информация за разплащането и адреса на доставка.

Описание на навигацията

Системата ще има 4 основни състояния (екрани). Навигацията между тях е показана на схемата:



Сценарии на работа



Данни в системата

Продуктивият каталог е подреден по категории. Всяка категория съдържа неограничен брой продукти. Всяка категория трябва да има заглавие.

Когато се разглежда каталога, се визуализират следните свойства на продуктите:

- Име
- Кратко описание
- Цена
- Снимка

В детайлният изглед се визуализират още:

- Детайлно описание

При финализирането на покупката клиентът попълва следната информация (всички полета са задължителни):

- Име
- Град на доставка (Избор от списък с предварително дефинирани градове)
- Адрес на доставка
- Телефон за връзка
- E-mail адрес за връзка
-

Функционални изисквания

- Да се валидират данните попълвани от клиента при покупка. E-mail адреса да се валидира за правилно форматиране.

-

Технически изисквания

- Да се реализират плавни преходи между състоянията (екраните) в системата. Да се използват states и transitions.
- Данните за продуктите, категориите и градовете за доставка да се зареждат от подходящо форматиран XML файлове.
- Данните за покупката на клиента да се пращат на метода SubmitOrder() на Web услугата <http://dev.gugga.com/FlexCourseShoppingBag/service.asmx?WSDL> . Тестовата конзола за услугата се намира на <http://dev.gugga.com/FlexCourseShoppingBag/service.asmx> .
- Да се използват механизмите за Data Binding във Flex, където това е подходящо.
- Да се използват стандартните контроли на Flex. За реализирането на количката (списъка с закъпени продукти) да се използва DataGrid.

Тетрис

курс	макс. големина екип
1	5
2	4
3	3
4	3

Табла

курс	макс. големина екип
1	5
2	4
3	3
4	3

Дама

курс	макс. големина екип
1	
2	
3	
4	

Белот

курс	макс. големина екип
1	5
2	4
3	3
4	3

Sokoban

курс	макс. големина екип
1	5
2	4
3	3
4	3

Snake

курс	макс. големина екип
1	3
2	3
3	2
4	2