



Министерство образования Республики Беларусь

**Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»**

Кафедра «Информационные технологии»

Д. В. Соболев, Т. С. Семенченя

ИНДУСТРИЯ КОМПЬЮТЕРНЫХ ИГР

ПРАКТИКУМ

**по одноименной дисциплине для студентов
специальности 1-40 05 01 «Информационные
системы и технологии (по направлениям)»
дневной формы обучения**

Гомель 2021

УДК 004.92:004.42(075.8)
ББК 32.973я73
С54

*Рекомендовано научно-методическим советом
факультета автоматизированных и информационных систем
ГГТУ им. П. О. Сухого
(протокол № 7 от 02.03.2020 г.)*

Рецензент: доц. каф. «Промышленная электроника» ГГТУ им. П. О. Сухого
канд. техн. наук *А. В. Ковалев*

Соболев, Д. В.
С54 Индустрия компьютерных игр : практикум по одной дисциплине для студентов специальности 1-40 05 01 «Информационные системы и технологии (по направлениям)» днев. и заоч. форм обучения / Д. В. Соболев, Т. С. Семенченя. – Гомель : ГГТУ им. П. О. Сухого, 2021. – 58 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.

Практикум знакомит студентов с историей развития компьютерных игр, учит описывать, анализировать и совершенствовать компьютерную игру по определенным параметрам, а также содержит материал для выполнения практических занятий.

Для студентов специальности 1-40 05 01 «Информационные системы и технологии (по направлениям)» дневной формы обучения.

УДК 004.92:004.42(075.8)
ББК 32.973я73

© Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», 2021

СОДЕРЖАНИЕ

Практическая работа № 1	4
Практическая работа № 2	13
Практическая работа № 3	24
Практическая работа № 4	34
Практическая работа № 5	41
Литература	58

ПРАКТИЧЕСКАЯ РАБОТА № 1

ЗНАКОМСТВО С *XMIND*. СОСТАВЛЕНИЕ ИНТЕЛЛЕКТ КАРТ С ОПИСАНИЕМ КОМПЬЮТЕРНОЙ ИГРЫ

1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Интеллект-карты

Интеллект-карты (в оригинале *Mind maps*) – это инструмент визуального отображения информации, позволяющий эффективно структурировать и обрабатывать ее.

Англоязычное словосочетание *Mind map* дословно переводится как «карта ума» («*map*» – карта, «*mind*» – ум). Встречаются и другие варианты перевода этого понятия: ментальная карта, карта мыслей, диаграмма связей или ассоциативная карта.

Интеллект-карта представляет собой древовидную схему, на которой изображены слова, идеи, задачи или другие понятия, связанные ветвями, отходящими от центрального понятия или идеи. Ветви, имеющие форму плавных линий, обозначаются и поясняются ключевыми словами или образами.

Интеллект-карты являются разработкой Тони Бьюзена, британского писателя, лектора и консультанта по вопросам интеллекта, психологии обучения и проблем мышления. По мнению разработчика, интеллект-карта – это мощный графический метод, предоставляющий универсальный ключ к высвобождению потенциала, скрытого в мозге. Благодаря этому метод интеллект-карт может найти применение в любой сфере жизни, где требуется совершенствование интеллектуального потенциала личности или решение разнообразных интеллектуальных задач (рисунок 1.1).

Mind Maps или интеллект-карты – это кустообразные картинки, изобретенные Тони Бьюзеном как способ представления и связывания мыслей. Рисуются они очень просто:

– берется лист бумаги и в центре картинкой или одним-двумя словами обозначается основное понятие или концепция, либо анализируемая проблема. Я обычно использую формат А4 «стоя», но часто рекомендуется брать лист бумаги формата А3 «лежа».

– Обводим это понятие в кружок – из него все будет произрастать.

– Рисуем ветки – основные связанные понятия, свойства, и т.п. Подписываем их одним-двумя словами каждую (получается далеко не всегда, тем более русские слова достаточно длинные);



Рисунок 1.1 – Применение интеллект-карт

– Дальше – больше. Каждая ветка делится на несколько более тонких веточек – развитие этих понятий, детализация свойств, направлений работы.

– Лист постепенно заполняется и проблема из страшной–непонятной делается вполне ручной – прищиплена тоненькими веточками как булавками к листу бумаги, уже не лаает–кусают, а тихо поскуливает.

– Вот теперь берем цветные ручки и внимательно смотрим на получившуюся картинку. И видим, что прищиплена проблема к листу кривовато, надо бы исправить. Обычно рекомендуется рисовать карандашами и чуть что – сразу стирать исправлять. По моему опыту – не рекомендую. Лучше не стирать – пусть на бумаге останется история мысли – а цветными ручками обвести некоторые веточки. Каждый цвет – своя связь. При некоторой тренировке очень легко на одной картинке видеть две схемы – на одной связи обозначены иерархией веток, на другой – одинаковой раскраской.

– Теперь, если картинка получилась красивой и достаточно запутанной, повесьте ее на стену. И проблема вся как на ладони, и гостям есть тема для вопросов.

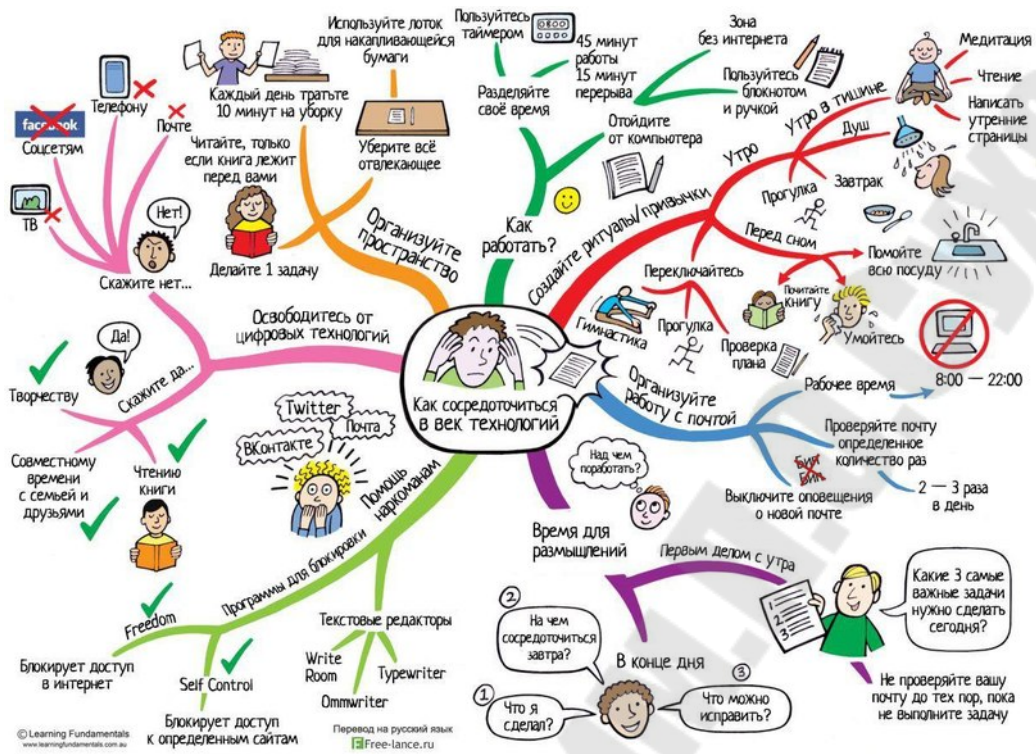


Рисунок 1.4 – Интеллект-карта «Как сосредоточиться в век технологий»

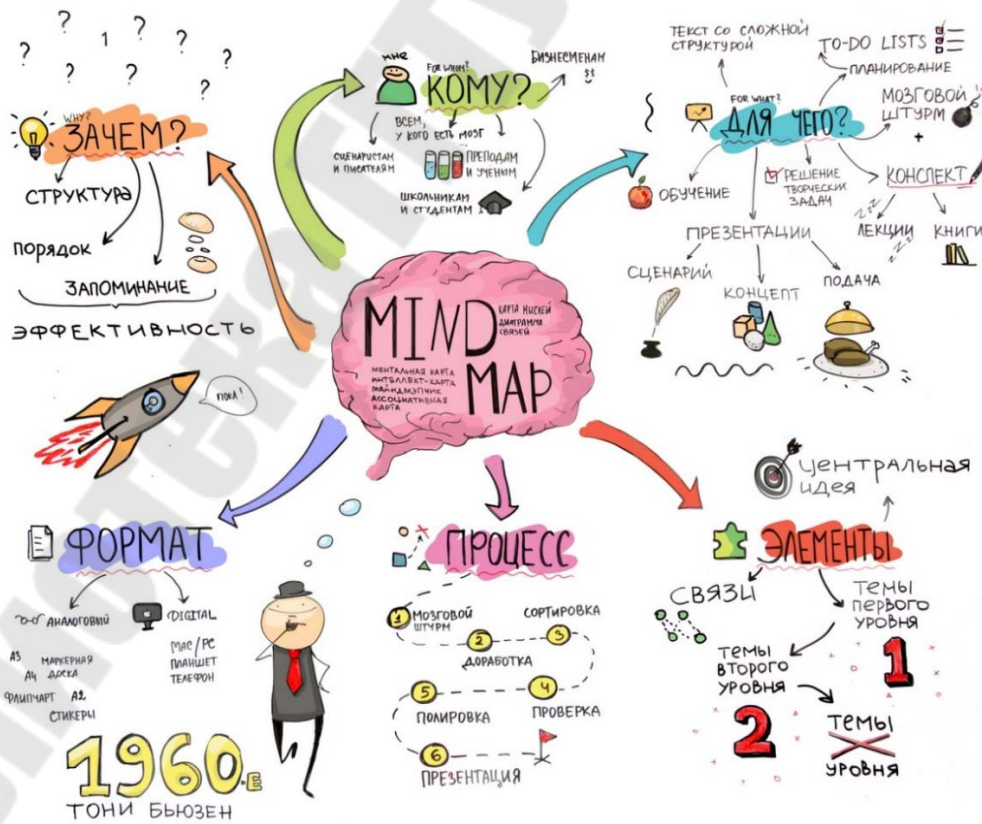


Рисунок 1.5 – Интеллект-карта «Mind map»

1.2 Знакомство с *XMind*

XMind – программа для составления так называемых интеллект-карт (*mind maps*). Такие карты помогают организовать информацию в наглядном ассоциативном виде. Каждый элемент карты может представлять собой мысль или идею, связанную с другими идеями при помощи иерархических связей. Вы можете использовать *XMind* для поиска оптимальных решений, мозгового штурма, решения бизнес-задач и многого другого. *XMind* позволяет организовывать карты идей в логическом, древовидном и других представлениях. Внешний вид элементов карты настраивается – вы можете устанавливать цвет фона, параметры шрифта, добавлять текстовые заметки и файлы, таблицы, маркеры, гиперссылки, аннотации и связывать одни идеи с другими.

Программа позволяет фокусироваться на определенных участках интеллект-карт, создавать голосовые заметки, проверять орфографию текста, использовать поиск элементов карт, экспортировать данные и опубликовывать созданные карты в интернете. В это приложение можно импортировать карты, созданные в других программах подобного рода, например – *FreeMind* и *MindManager*. *XMind* поддерживает шаблоны и темы для карт, а также позволяет закрывать карты от несанкционированного доступа паролем. Помимо бесплатной, существует и несколько платных версий *XMind*, позволяющих сохранять карты в форматы *Word*, *Power Point* и *PDF*, работать в режиме мозгового штурма и создавать презентации.



Рисунок 1.6 – Интеллект карта в *XMind*

На рисунке 1.6 представлен пример создания интеллект-карты в *XMind*.

В последней версии *XMind* много нововведений:

– новые иконки (рисунок 1.7). Интеллект-карты в *XMind* теперь можно украшать. Доступна библиотека цветных иконок, которые помогут визуализировать мысли (придать им образ). Добавлено 109 новых клип-артов: все сгруппированы по темам.

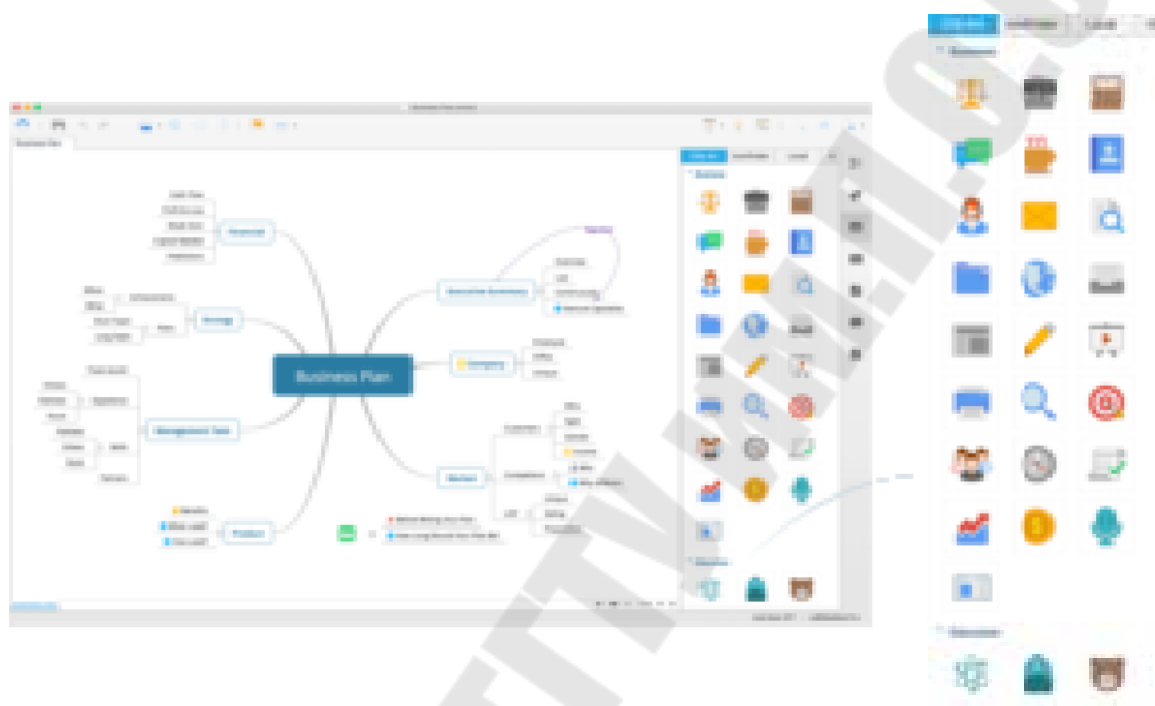


Рисунок 1.7 – Новые иконки

– Новые шрифты и темы блоков. До 10 удивительных шрифтов для ваших карт (открытый код, кроссплатформенность гарантируют привлекательный вид на всех операционных системах). И столько же новых тем. А также редактор *Advanced Theme*, для создания собственной темы.

– Новые шаблоны тем для экономии вашего времени (рисунок 1.8). В программу встроены готовые шаблоны карт. Благодаря заготовкам любой пользователь сможет быстро создать красивую ассоциативную диаграмму.

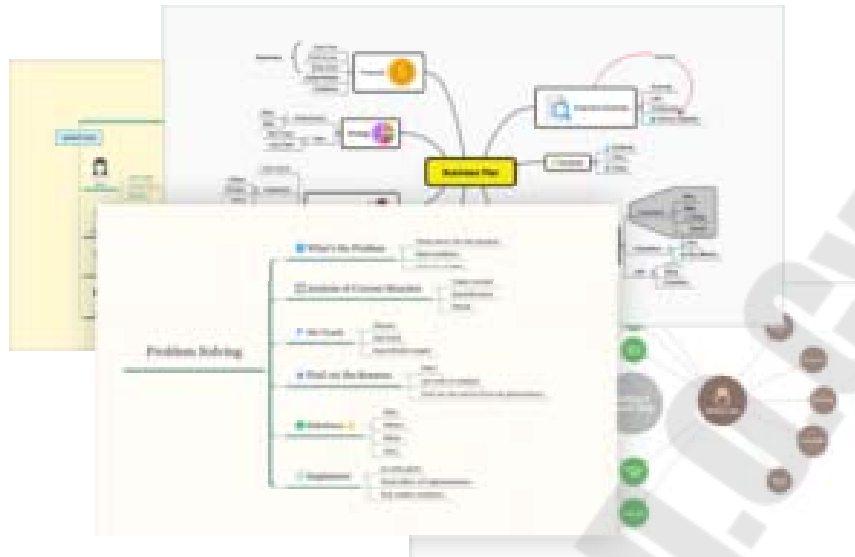


Рисунок 1.8 – Новые шаблоны тем

– Работа с *Office / PDF*. Для подготовки отчета или доклада мы прибегаем к офисным программам. Разработчики *XMind* посчитает, что будет полезно иметь возможность одним щелчком мыши экспортировать нужные диаграммы из *XMind* в любую из офисных программ. И исполнили задуманное: теперь из утилиты можно экспортировать интеллект-карты, схемы, таблицы, диаграммы в: текстовый редактор (*doc, rtf*), графический (в форматах *PNG, JPEG, GIF* и др.), PowerPoint.

2 ЗАДАНИЕ

Составить интеллект карту на заданную тему (варианты тем приведены в таблице 1.1).

Таблица 1.1

Темы заданий	
Номер варианта	Тема
1	Игра «Шахматы»
2	Игра «Кости»
3	Игра «Лото»
4	Игра «Поле чудес»
5	Игра «Камень–ножницы–бумага»
6	Игра «Города»
7	Игра «Уно»
8	Игра «1000»
9	Игра «Уголки»
10	Игра «Домино»
11	Игра «Крестики–нолики»
12	Игра «Дурак»
13	Игра «Шашки»
14	Игра «Морской бой»
15	Игра «Танчики»
16	Игра «Гонки»
17	Игра «Пакман»
18	Игра «Тетрис»
19	Игра «Zuma»
20	Игра «Mario»
21	Игра «Мафия»
22	Игра «Крокодил»
23	Игра «Прятки»
24	Игра «Твистер»
25	Игра «Догонялки»
26	Игра «Кич»
27	Игра «Баланс»
28	Игра «Башня (Дженга)»
29	Игра «Дубль»
30	Игра «Точки»

3 ТРЕБОВАНИЯ К ОТЧЕТУ

В отчете должны быть отображены следующие пункты:

1. Задание.
2. Краткие теоретические сведения.
3. Результат выполнения задания для своего варианта.
4. Заключение по результатам работы.

ПРАКТИЧЕСКАЯ РАБОТА № 2 СОСТАВЛЕНИЕ АНАЛИТИЧЕСКОГО ОБЗОРА КОМПЬЮТЕРНОЙ ИГРЫ.

1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Классификация компьютерных игр по жанрам

На начальном этапе развития игровой индустрии каждая игра была уникальна и неповторима. Но со временем игр становилось всё больше и больше. Для того, чтобы ориентироваться во всём многообразии игр, люди стали разделять игры на различные категории.

Критерии деления на категории были самые разные: количество игроков, вид игровой камеры, сюжет. Но большую популярность получил способ деления на игровые жанры (таблица 2.1).

Игровой жанр – группа игр, которые имеют схожую игровую механику и похожие правила игры. Жанр игры определяет то, какие действия чаще всего необходимо выполнять игроку для победы. Примеры: в шутерах нужно хорошо стрелять, в экшенах – иметь хорошую реакцию и применять в бою разнообразные способности, в платформерах – преодолевать препятствия, в гонках – обгонять соперников, в стратегиях – развиваться и действовать умнее, чем соперники.

Указание жанра игры само по себе уже даёт общее примерное представление о содержании игры. Основная масса игроков выбирает новые игры, ориентируясь на свои жанровые предпочтения.

На рисунке 2.1 представлена схема жанров компьютерных игр.

В схеме собраны все общеупотребимые игровые жанры, которые известны большинству игроков, только расположены они так, что образуют чёткую структуру.

Вся схема разделена на 15 вертикальных полос. 15 полос разгруппированы на 3 большие группы игр – «Игры информации», «Игры действий», «Игры контроля».

В нижней части каждой полосы указана её сущность – базовый (неделимый) жанр, который можно назвать одним словом. Все названия сущностей: обучение, загадки, общение, герой, изучение, собирание, уклонение, уничтожение, соревнование, вождение, забота, развитие, контроль, тактика, планирование.

Таблица 2.1

Жанры компьютерных игр

Категория	Игры информации	Игры действий	Игры контроля
Гибридные жанры	Action–RPG (боевая ролевая игра) Roguelike (рогалик)	ММОFPS (сетевой экшн) Survival (выживание)	RTS (стратегия–боевик) МОБА (Дота–подобная)
5 элементов	Open RPG (открытая ролевая игра)	Open Action (открытый боевик)	Global Strategy (глобальная стратегия)
3 элемента	RPG (ролевая игра) MUD (текстовая онлайн-игра) MMORPG (онлайн-ролевая игра)	Action (боевик) Slasher (сражение) Battle Racing (гонки–сражения)	Strategy (локальная стратегия) Sim Strategy (непрямого контроля) Global Wargame (глобальная военная)
2 элемента	Puzzle (головоломка) Quest (квест) Browser RPG (браузерные ролевые) Adventure (приключения)	Platformer (платформер) Stealth–Action (Шпионский боевик) Fighting (поединок) Racing (гонки)	Economical (экономическая стратегия) Tower Defence (защитные башни) Wargame (военная игра) Cardgame (карточная ролевая игра)
1 элемент	Education (обучающая игра) Test (вопросы, загадки) Contact (общение) Hero (геройская игра) Toure (путешествие)	Arcade (аркада) Horror (ужасы, выживание) Shooter (стрелялка) Sport (спорт) Simulator (симулятор)	Logic (Логические игры) Tactic (Тактическая игра) MicroControl (микроконтроль) Building (строительство) Life Sim (сим)
Элементарные жанры	1.1. Обучение 1.2. Загадки 1.3. Общение 1.4. Роль 1.5. Изучение	2.1. Собрание 2.2. Уклонение 2.3. Уничтожение 2.4. Соревнование 2.5. Вождение	3.1. Забота 3.2. Создание 3.3. Контроль 3.4. Тактика 3.5. Планирование



Рисунок 2.1 – Схема жанров компьютерных игр

Сверху от названия сущностей (базовых игровых механик) расположены сами жанры в виде прямоугольников. Самые простые жанры находятся внизу, они занимают всего одну полосу. Чем выше расположен жанр, тем на большем количестве полос он располагается. Вхождение блока жанра сразу в несколько полос означает, что в нём присутствует сразу несколько базовых игровых механик, объединенных в единое целое. (Например: жанр «Квест» это объединение двух базовых классов «Загадки», «Общение»).

1.2 Сеттинг. Классификация игр по сеттингу

Разделение компьютерных игр на жанры весьма специфично и не похоже на систему жанров фильмов и книг. Игровые жанры определяют лишь основные действия, которые будут совершать игроки в процессе игры, тем самым они отвечают только на вопрос «ЧТО?». На вопросы «ГДЕ?» и «КОГДА?» отвечает другая основная характеристика игры – сеттинг.

Сеттинг – это принадлежность игры к какой-то сюжетной теме или к определённому виртуальному миру. В среде компьютерных игр сформировалось несколько наиболее популярных сеттингов: фэнтези, научная фантастика (*sci-fi*), вторая мировая война, средневековье, стимпанк, постядерный мир, аниме, комиксы.

Создание игры в популярном сеттинге обеспечивает её собственную популярность, да и игроки чувствуют себя уютно и

комфортно в уже знакомом мире. Некоторые игры создаются в своих уникальных сеттингах или в необычных сочетаниях стандартных тем. Такие игры менее популярны, но, тем не менее, они имеют свою аудиторию особых игроков, которые терпеть не могут шаблонность и однообразие.

Классификация игр по сеттингу.

По месту действия (тип вымышленного мира):

1. Реальный мир (игровой мир мало чем отличается от нашего мира).
2. Параллельные миры (реальный мир и переходы в искаженные миры).
3. Альтернативная история (другие варианты исторических событий).
4. Фэнтези (сказочный мир с наличием магии).
5. Геройская мифология (сверхсущества: супергерои, мутанты, полубоги).
6. Христианская мифология (ангелы, демоны, рай, ад).
7. Современная мифология (нашествие зомби, нашествие инопланетян, постапокалипсис).

По времени действия (историческая эпоха):

1. Зарождение жизни (простейшие организмы, игры на клеточном уровне).
2. Доисторические времена (эра динозавров, пещерные люди).
3. Средневековье (рыцарские походы, междоусобицы, инквизиция).
4. Эпоха колонизации (морские путешествия, новые земли).
5. Эпоха индустриализации (XVIII – XIX века, стимпанк).
6. Прошедшие войны (Вторая Мировая война, локальные конфликты).
7. Наше время (привычный мир).
8. Информационная эпоха (киберпанк, антиутопия будущего).
9. Освоение космоса (научная фантастика, *Sci-Fi*, космические путешествия).
10. Эволюция (наличие нескольких эпох в одной игре).

Чаще всего игроков разделяют на казуальных, мидкорных и хардкорных (таблица 2.2).

Таблица 2.2

Разделение игроков

Параметр	Казуальный		Мидкорный		Хардкорный	
	М	Ж	М	Ж	М	Ж
Соотношение	40%	60%	55%	45%	72%	28%
Возраст	≈18	≈35	≈25	≈28	≈20	≈25
Устройства	Смартфоны/приставки					
Ежедневно играют	20–60 минут		30–120 минут		120–300 минут	
Цель игры	Убить время		Удовольствие и убийство времени		Удовольствие и прокачка навыков	
Жанры	Азартные Настольные	Приключения Головоломки	РПГ Экшн	Стратегии Фермы	ММО Шутеры Гонки	РПГ Экшн ММО

С появлением внутриигровых покупок игроков стали разделять и по уровню финансовых возможностей (рисунки 2.2–2.4).

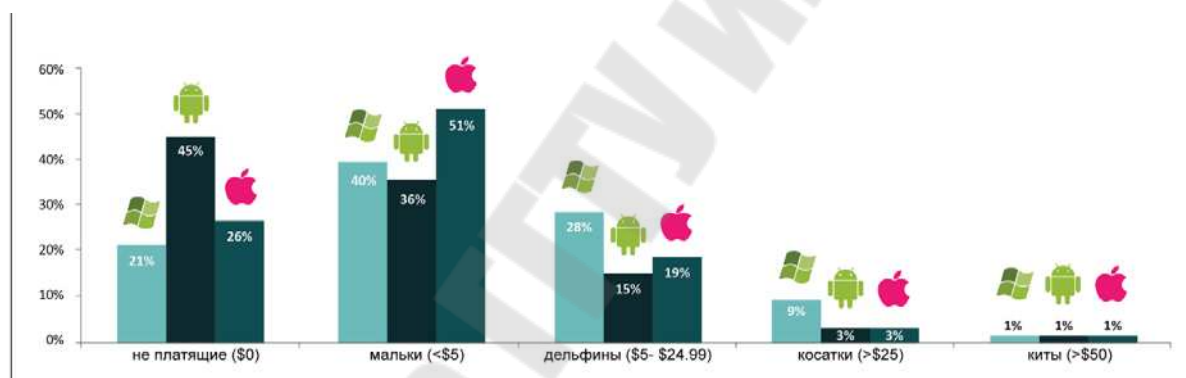


Рисунок 2.2 – Сегментация платящих пользователей в играх по платформам

	 Мужчины	 Женщины	 Средний возраст	 Недельная вовлеченность мобильными играми	 Недельная вовлеченность всеми играми
не платят	35%	65%	36.6	4.1 часа	7.2 часа
платят мало	53%	47%	32.5	4.9 часа	9.9 часа
платят много	66%	34%	30.2	11.8 часа	26.5 часа
в среднем	44%	56%	34.6	-	-

Рисунок 2.3 – Сегментация платящих пользователей в играх по полу и возрасту

ИГРОВОЕ ВРЕМЯ НА ВСЕХ ТИПАХ УСТРОЙСТВ

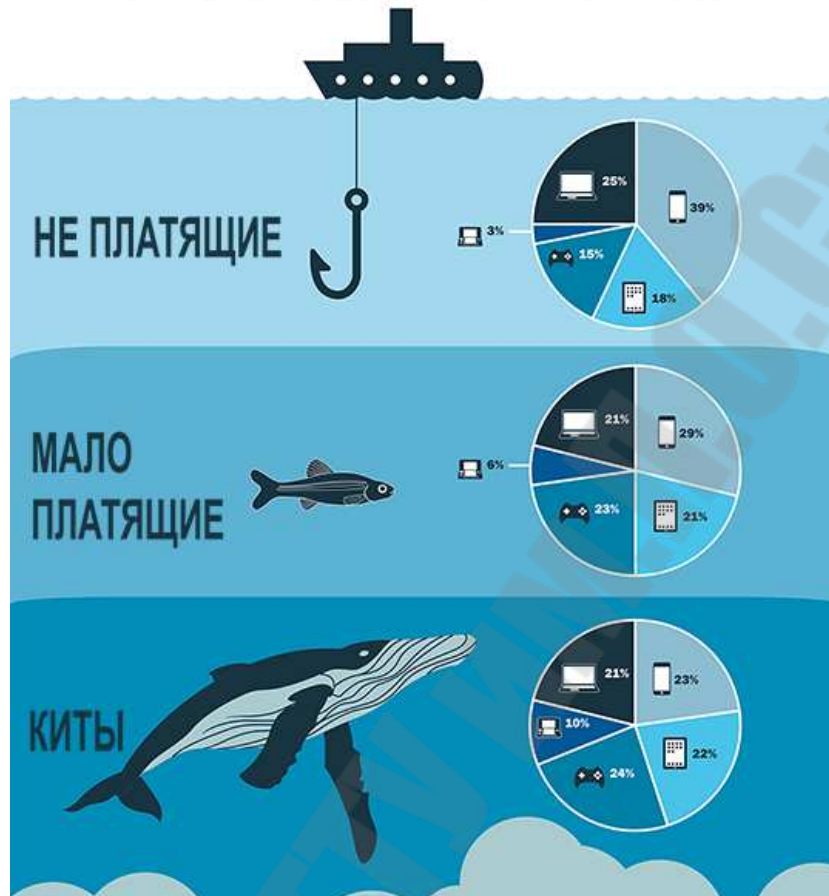


Рисунок 2.4 – Игровое время на всех типах устройств

1.3 Игровой цикл

Игра – она как луковица. У нее есть геймплейная сердцевинка, к которой по слоям крепятся все остальные механики. Так вот эта сердцевинка, «одетая во сто шуб» – это и есть *core gameplay*. Одним словом, это то, что делает вашу игру такой, это то, без чего она просто перестанет работать. Шутер – очень сложно представить без стрельбы, тетрис – без деталей, которые нужно складывать вместе и так далее. Таким образом, можно сказать, что *core gameplay* – это небольшой набор ключевых механик, которые являются самыми важными для вашей игры, без которых представить себе ее существование просто невозможно. Дальше, давайте рассуждать логически: если механик у нас всего несколько, значит они должны как-то повторяться на протяжении всей игры, чтобы игроку было что делать. Здесь нам на помощь и приходит слово из названия этого урока, а именно *loop*, что переводится с английского, как «петля».

Игровой цикл (англ. *gameplay loop*, также англ. *core gameplay loop*) – принцип, согласно которому геймдизайнеры задают главный элемент игровой механики, который определяет фундаментальный опыт игрока. Один игровой цикл представляет собой действие игрока, результат этого действия в игровом мире, реакцию игрока на результат и запрос игры на повторение нового действия.

1.4 Метагейм

Метагейм (от греч. *meta* – за пределами и англ. *game* – игра), в широком понимании – всё, что относится к пользователю, а не игре и её персонажам. Рассуждения и мысли игрока, а не его персонажа.

Метагейм может принимать следующие формы:

Использование знания игромеханики (нередко в ущерб отыгрышу) – например, использование боевой тактики, направленной на слабости монстра, известные игроку по прошлым играм или почерпнутые напрямую из монстрятника. (В «*Advanced Bestiary*» *GRP* рекомендуют менять описание монстра с оставлением его оцифровки, фактически обманывая игроков).

Использование сеттинговых сведений, неизвестных персонажу из-за его образования, воспитания, принадлежности к той или иной группе населения, но известных игроку опять-таки по прошлым играм или почерпнутых из чтения сеттинговых дополнений. Такие сведения, строго говоря, метаигровыми не являются, в отличие от способа их получения.

Знание модуля: некоторые модули содержат сюжетные повороты, ловушки или загадки, требующие нетривиального подхода к их распутыванию, но легко проходимые теми, кто про них знает. Многие модули содержат предупредительные надписи о том, что некоторые разделы читать категорически не следует тем, кто собирается по этому модулю играть – и дело тут не в том, чтобы не дать игрокам «победить» мастера, а в том, что получаемое от прохождения модуля удовольствие идёт от самого процесса преодоления препятствий, а не от нажатия нужных кнопок.

«Подглядывание» как реакция персонажа на события, свидетелем которых он(а) не был(а). Из-за сложности «развидеть» виденное и забыть услышанное мастера в настольных играх удаляют игроков в некоторых сценах «на кухню» или закрывают чужие ветки

на форумной игре. Если объёмы конфиденциальной информации невелики, можно обойтись записками или личными сообщениями.

1.5 Пример аналитического обзора игры

Аналитический обзор игры – обзор основных элементов и циклов игры, оценка ее целевой аудитории, стилистики, определение жанра и сеттинга.

Пример аналитического обзора игры *Battle phase*

Игра в жанрах экономической стратегии, в которой игрок будет выступать в роли министра военной промышленности. Он возьмёт под своё управление предприятия различных отраслей и должен будет настраивать производственные цепочки по всей стране таким образом, что бы вовремя выполнять заказы и удовлетворять потребности армии в различных видах вооружения, боеприпасов и техники.

Основной фишкой игры, станет способ отображения успехов и неудач игрока в производственной сфере в виде успехов армии и флота, т.е. если игрок вовремя поставляет армии всё необходимое что бы иметь численное и качественное преимущество над противником, то та сама наступает и побеждает, захватывая новые предприятия и ресурсы.

Сеттинг: реалистичный.

Стилистика арта: схематическая (рисунки 2.5, 2.6).



Рисунок 2.5 – Арт игры Battle phase



Рисунок 2.6 – Вид игры в мобильном устройстве

Жанр: Макроэкономическая стратегия.

Описание игры. Игрок выступает в роли министра мобилизованной по военному времени промышленности. Он управляет вверенными ему предприятиями: добывает, перерабатывает ресурсы, производит необходимые армии изделия, модернизирует предприятия. В основном игрок занимается высокоуровневым управлением целыми предприятиями, управляет их загрузкой, номенклатурой и размерами партий производимых изделий, выделяет время и ресурсы на модернизацию производств, запускают производство новых образцов, занимается настройкой производственных цепочек для своевременной поставки ресурсов и комплектующих. Игра от имени армии/правительства ставит перед игроком задачи в указанные сроки поставлять определенные изделия в нужных фронту количествах.

Основная особенность. Основной особенностью игры является “непрямое” управление армией. Игрок может обеспечить победу армии своей страны обеспечивая ей количественное и качественное превосходство над армией противника. При этом фронты самостоятельно двигаются по карте захватывая чужие предприятия и ресурсы либо теряя свои. На боевые действия расходуются техника, боеприпасы и живая сила. Такой способ отображения успехов игрока так же создаёт платформу для реализации интересного мультиплеера, как РУР так и кооперативного.

Целевая аудитория. Порядка 1 млн человек могут заинтересоваться игрой в жанре экономической стратегии.

2 ЗАДАНИЕ

Согласно теме игры, представленной в таблице 2.3, составить аналитический обзор компьютерной игры по следующему плану:

- Классификация игры;
- Жанр;
- Сеттинг;
- Целевая аудитория;
- Список элементов игры;
- Список циклов в игре;
- Метагейм.

Таблица 2.3

Темы заданий

Номер варианта	Тема для обзора игры
1	Квест
2	Головоломки
3	Стратегия
4	РПГ
5	Экшен
6	Симулятор
7	Аркада
8	Настольные игры
9	Музыкальные игры
10	Азартные игры
11	Спортивные игры
12	Приключения
13	Файтинг
14	Авиасимулятор
15	Управленческо–экономические игры
16	Игра для игрового автомата
17	Экономический симулятор
18	Тактический шутер
19	Инкрементальная игра
20	Стелс

3 ТРЕБОВАНИЯ К ОТЧЕТУ

В отчете должны быть отображены следующие пункты:

1. Задание.
2. Краткие теоретические сведения.
3. Результат выполнения задания для своего варианта.
4. Заключение по результатам работы.

ПРАКТИЧЕСКАЯ РАБОТА № 3

СОСТАВЛЕНИЕ КОНЦЕПЦИИ КОМПЬЮТЕРНОЙ ИГРЫ ПО ЗАДАНЫМ ТРЕБОВАНИЯМ

1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Концепт-документ

Концепт-документ (концепт) – это краткое и ёмкое описание концепции (идеи) игры, то есть, максимально сжатый документ, в котором рассказывается о том, какой будет игра, чем она будет интересна и как она должна выглядеть после разработки.

Концепт-документ необходим как для дизайнера игры, так и для команды разработчиков и издателя. Первым двум он нужен для того, чтобы удерживать основной ход своих мыслей и не потерять «видение игры» (так называемый *Vision*), положительно ограничить рамки разработки. Издателю концепт–документ интересен из–за того, что в нём коротко и ясно рассказано о будущей игре, и потому издатель способен оценить, стоит ли ему браться за этот проект или нет.

Стандартные пункты концепт-документа таковы:

1) Жанр и аудитория – указывается жанр, предполагаемая аудитория игры, наличие лицензируемого материала или ограничивающего контента.

2) Основные особенности игры (*USP*) – список основных особенностей игры, которые игроку будут интересны и за которые он выложит деньги. Лучше всего, если в игре будет Одна Главная Особенность.

3) Геймплей – наиболее полное и в то же время обобщённое (без лишних деталей) описание всего игрового процесса. Нужно ответить на вопросы о том, что должен делать игрок, как он это должен делать и, наконец, ответить лично для себя на вопрос: «Чем игроку будет интересно каждое из этих действий?».

4) Сюжет – в этом пункте стоит коснуться сюжета, но не просто предыстории событий, а именно сюжетной линии игры. Огромное количество литературного материала в данном пункте никого не заинтересует.

5) Системные требования – здесь нужно оценить, какие требования будет предъявлять игра к «железу» и выписать их минимальные и максимальные (рекомендуемые) границы.

б) Сроки и бюджет разработки – в этом пункте нужно подсчитать, сколько будут стоить работы по проекту и какое время на них будет затрачено, а затем выписать полученные значения.

Рассмотрим пример концепт–документа на основе игры «*Ryaba's Revenge*».

Жанр и аудитория.

Игра «*Ryaba's Revenge*» относится к жанру стратегий в реальном времени (*RTS*) в изометрической проекции со связанным сюжетом и элементами *RPG*, разрабатывается только в версии для *PC*.

Игра ориентирована на широкую аудиторию, не содержит ограничивающего контента, минимальный возраст игрока – 8 лет. Дополнительную привлекательность игра имеет для владельцев не самой современной конфигурации *PC* и людей, ищущих новые подходы в жанре *RTS*, а также для интересующихся русскими народными сказками. Для младшей части этой категории характерна потребность в простых играх.

Игра не использует торговые марки или другую собственность, подлежащую лицензированию.

Сеттинг: фэнтезийный вымышленный мир, наше время.

Ключевые особенности игры (USP):

– Неповторимый дух русских сказок. Возможность «примерить на себя» известные с детства истории.

– Выделенный юнит с особыми свойствами, которым игрок управляет отдельно – Курочка Ряба. Почувствуйте себя героем и командиром!

– Нетрадиционный подход к использованию ресурсов. Теперь ресурсы нужны не для постройки юнитов, а только для их поддержания.

– Множество загадок и мини–миссий, требующих тактических решений.

– Невысокие требования к технике при отличной яркой графике и богатом звуке.

– *Violent-free*. Убийство выполняется в лучших традициях детских игр – превращение в бабочек, цветки и подобное.

Игра пригодна для издания на западном рынке.

Сюжет рассчитан на 5~10 часов прохождения.

Основные элементы игры:

1. Карта – неисследованный лабиринт, игрок должен с осторожностью открывать и осваивать новые области.

2. Вражеские юниты – Осы, Крысы, Коты – нападают на отряды игрока.

3. Специальные объекты – Черный кот, Медведь-шатун, Лис Прохвост – обладают особыми характеристиками, могут взаимодействовать с отрядами игрока и с вражескими юнитами. Могут сотрудничать с игроком либо могут быть побеждены игроком. В бою используют приемы, недоступные обыкновенным юнитам.

4. Специальные квесты – квесты, не относящиеся к основному сюжету, но наполняющие игру контентом (открытие ворот, нахождение золотых когтей и клюва, освобождение Полкана).

5. Строения – сооружения, которые используются для производства юнитов, хранения, производства и переработки ресурсов. Некоторые строения игрок может строить самостоятельно, используя своих юнитов и затрачивая на это определенные ресурсы.

6. Курятники – строения, которые игрок строить не может. Используются для найма юнитов – Цыплят. Курятники также используются для апгрейда Цыплят путем обучения их чарам, если в них присутствует игрок юнит – Наседка. Если Наседки нет, игрок не сможет обучать юнитов в этом Курятнике.

7. Колодцы – тип строений, которые может строить игрок. Количество воды в колодце непостоянно и возможные уровни добычи воды неравномерно распределены по местности картой «уровня вод», задаваемой при дизайне уровней.

8. Кормушки – тип строений, которые игрок строить может, располагаются (могут и должны, располагаться) недалеко от курятников, используются для апгрейдов Цыплят – их превращения в Петушков и Жар-птиц.

9. Команды – используются игроком для управления отрядами. Стандартные команды, отвечающие за передвижение и бой юнитов, одинаковы для всех; некоторые юниты имеют дополнительные возможности и соответственно дополнительные команды.

10. Вода – единственно необходимый и доступный ресурс. Поиск воды является, одним из видов деятельности, которым занимается игрок. Вода находится в бидоне и расходуется со скоростью, зависящей от размера войска и активности боевых действий. Для разработки этого ресурса используются юниты – Цыплята, которые носят воду из водоемов или колодцев.

Игровой цикл: игрок формирует отряд, добывает ресурсы, сражается, выполняет квесты → тратит опыт и ресурсы на улучшение боевых отрядов → повторяет свои действия.

При достижении игроком конечной цели – прохождение все трех уровней и получение Золотого яйца – игра заканчивается. Продолжать играть можно сначала или с любого сохраненного места.

Пример геймплея.

Вид игры – изометрия в $\frac{3}{4}$, тайловый ландшафт с дискретным изменением высоты, плоский интерфейс пользователя, управление полностью мышью, с возможностью назначения быстрых вызовов на клавиатуре (рисунок 3.1).

Интерфейс функционально привычен и стандартен для *RTS*, за исключением бидона как визуального индикатора наличия самого важного ресурса – воды.

Стиль юнитов и построек мультяшно–сказочный, красочный, озвучена игра звонко, весело, понятно даже самым маленьким игрокам.



Рисунок 3.1 – Арт игрового приложения

Общий стиль графики лубочный славянский, вписывающийся в сеттинг и подчеркивающий атмосферу сказочности игры (рисунок 3.2).

Общая цель миссии – вырастить 20 цыплят.

Начальные условия: стартовая позиция возле колодца, одна Ряба, одна Наседка.



Рисунок 3.2 – Пример графики приложения

Задание: вырастить пять цыплят. Разведать территорию вокруг хутора.

Игрок выбирает простое гнездовище (Панель «Строить» → Гнездовище простое) и строит его неподалеку от колодца (*LClick* на точке строительства).

Выбирает Наседку (*LClick* на ней) и привязывает к гнездовищу (ВСЁ на гнездовище). Наседка подходит к гнездовищу и начинает высидывать яйца. В статусной панели игрок видит истекающую полосу, символизирующую таймер откладывания яиц.

Игрок выделяет Рябу и начинает разведывать территорию вокруг хутора, направляя движение курицы (*LClick* на ландшафте или *LClick* на мини-карте).

Игрок находит ручей и стог сена рядом, возвращает камеру к колодцу (*LClick* на мини-карте или скроллинг). Там уже выращены пять цыплят, и наседка освободилась. Они объединяются в группу (рамка, *Ctrl+I*) и отправляются в район ручья (*LClick* на мини-карте). У ручья игрок строит новое гнездовище (Панель «Строить» → Гнездовище из сена).

Задание: обеспечить семью водой.

Игрок выделяет пять цыплят (рамка), которые назначаются на добычу воды (*RClick* на ручье). Цыплята начинают таскать воду в гнездовище. Бидон в интерфейсе медленно наполняется.

Игрок тем временем разведывает территорию дальше с помощью Рябы. Он замечает Кота, крадущегося в направлении ручья и нового гнездовища.

Задание: убережь цыплят от Кота.

Кот подбегает к цыплятам, носящим воду, и хватает одного. На месте цыпленка появляется желтенький цветочек. Игрок же отдает приказ Рябе атаковать (*RClick* на Коте), она кудахчет и клюет Кота. Тот убегает туман войны.

Задание: высидеть больше цыплят.

Воды достаточно, и игрок строит кормушку (Панель «Строить» → Кормушка простая), после чего отправляет Наседку в гнездовище (*LClick* на ней, *RClick* на гнездовище). Наседка продолжает производить цыплят. По мере появления цыплята направляются на добычу воды коручью (рамка, *RClick* на ручье).

Задание: исследовать территорию вдоль ручья.

Ряба под управлением игрока продолжает исследовать местность вдоль ручья. Вскоре находятся залежи орешков, вокруг которых летают Осы. Игрок оставляет Рябу на безопасном расстоянии в один экран и вызывает цыплят (*Shift+I*, *LClick* на ландшафте). Рябе отдается приказ распугивать врагов (команда *Guard*) – с кудахтаньем и размахивая крыльями она бросается на Ос. Осы разлетаются со злобным жужжанием; но в целом продолжают крутиться над залежами ресурсов. Цыплята приходят; и игрок приказывает строить новую кормушку (Панель «Строить» → Кормушка большая). Цыплята строят ее, а Ряба тем временем продолжает отгонять Ос. Но одной из Ос удается ужалить цыпленка, он превращается в бабочку и улетает.

Задание: построить отряд Петушков и исследовать другой берег.

Кормушка готова, Осы разогнаны, и игрок начинает улучшать цыплят, посылая их поочередно в кормушку (выделение цыпленка, *RClick* на большой кормушке). Полученные молодые петушки умеют форсировать ручей, и игрок направляет их вместе с Рябой на другой берег. Там быстро находится большая залежь ресурсов, охраняемая отрядом Навозных жуков. Девять Петушков под контролем игрока легко их разгоняют (рамка, *RClick* на любом враге) одного за другим.

Задание: высидеть больше цыплят.

Игрок строит еще одну кормушку (Панель «Строить» → Кормушка большая), гнездовище (Панель «Строить» → Гнездовище простое), Наседка. форсирует ручей и высиживает яйца. Вылупляется 20-й цыпленок и миссия заканчивается.

Сравнение и особенности позиционирования

Игра «*Ryaba's Revenge*» имеет, с одной стороны, некоторые оригинальные решения в жанре, но в то же время концепция игры использует следующие лучшие свойства выбранных образцов:

1. *Warcraft* – классика жанра *RTS*: исследование карты, управление юнитами, обеспечение ресурсами. Также использован подход жесткого сюжета, который выступает как центральная ведущая линия и дает минимум свободы в выборе целей.

2. Князь – использование славянского сеттинга.

3. *Wargames* – минимальный менеджмент ресурсов, ,акцент на планировании действий и ведении сражения.

4. Казаки – применен подход, в котором ресурсы тратятся постоянно, в зависимости от размера войска.

5. *C&C*, *Warcraft III* – выделенный персонаж, игровое взаимодействие с которым ближе к *RPG*, чем *RTS*.

Данный подбор свойств игры ориентировался на такие современные тенденции в *RTS*, как использование разнообразных сеттингов, «эксперименты» с ресурсами, использование *RPG* – элементов для юнитов.

Основные отличия от конкурентов:

1. Полное отсутствие жестокости, крови, убийств, ориентация на детскую аудиторию

2. Игра короче и проще, более, насыщена роликами и внутриигровыми событиями, больший упор на повествование, а не стратегическое развитие. Игра ориентирована на максимально казуальную аудиторию.

3. Необычный сеттинг, визуально и по звуковому ряду больше похожий на картинки из детских книжек, мультфильмы.

Платформа

Разработка игры для иных платформ кроме *PC Windows* не планируется.

Необходимые требования к компьютеру представлены в таблице 3.1.

Таблица 3.1

Требования к компьютеру

Требования	Минимальные	Рекомендуемые
Процессор	<i>Celeron 266 МГц</i>	<i>Pentium 3400 МГц</i>
ОЗУ	<i>128 МБ</i>	<i>256МБ</i>
Видеокарта	<i>DirectX 5 совместимая, класса ATI RAGE</i>	

2 ЗАДАНИЕ

Составить концепт компьютерной игры по заданным требованиям. Темы игры представлена в таблице 3.2.

План описания концепта:

1. Видение.
2. Сеттинг. Целевая аудитория.
3. Описание того, что делает игрок.
4. Детализация того, что делает игрок (описание наиболее важных элементов и главного героя).
5. Список USP – уникальные, неповторимые элементы в игре, которые привлекут пользователя.

Таблица 3.2

Варианты заданий

Номер варианта	Тема для обзора игры
1	Квест
2	Головоломки
3	Стратегия
4	РПГ
5	Экшен
6	Симулятор
7	Аркада
8	Настольные игры
9	Музыкальные игры
10	Азартные игры
11	Спортивные игры
12	Приключения
13	Файтинг
14	Авиасимулятор
15	Управленческо–экономические игры
16	Игра для игрового автомата
17	Экономический симулятор
18	Тактический шутер
19	Инкрементальная игра
20	Стелс

3 ТРЕБОВАНИЯ К ОТЧЕТУ

В отчете должны быть отображены следующие пункты:

1. Задание.
2. Краткие теоретические сведения.
3. Результат выполнения задания для своего варианта.
4. Заключение по результатам работы.

ПРАКТИЧЕСКАЯ РАБОТА № 4 ЗНАКОМСТВО С МЕТОДИКОЙ «МОЗГОВОГО ШТУРМА» (BRAINSTORM)

1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Понятия и определения

«**Мозговой штурм**» (англ. *brainstorming*) — один из наиболее популярных методов стимулирования творческой активности. Позволяет найти решение сложных проблем путем применения специальных правил обсуждения. Широко используется во многих организациях для поиска нетрадиционных решений самых разнообразных задач.

Метод мозгового штурма был разработан Алексом Осборном в 1953 году. Метод основан на допущении, что одним из основных препятствий для рождения новых идей является «боязнь оценки»: люди часто не высказывают вслух интересные неординарные идеи из-за опасения встретиться со скептическим либо даже враждебным к ним отношением со стороны руководителей и коллег. Целью его применения является исключение оценочного компонента на начальных стадиях создания идей.

Классическая техника мозгового штурма, предложенная Осборном, основывается на двух основных принципах – «отсрочка вынесения приговора идее» и «из количества рождается качество». Этот подход предполагает применение нескольких правил.

Критика исключается: на стадии генерации идей высказывание любой критики в адрес авторов идей (как своих, так и чужих) не допускается. Работающие в интерактивных группах должны быть свободны от опасений, что их будут оценивать по предлагаемым ими идеям.

Приветствуется свободный полет фантазии: люди должны попытаться максимально раскрепостить свое воображение. Разрешено высказывать любые, даже самые абсурдные или фантастические идеи. Не существует идей настолько несуразных либо непрактичных, чтобы их нельзя было высказать вслух.

Идей должно быть много: каждого участника сессии просят представить максимально возможное количество идей.

Комбинирование и совершенствование предложенных идей: на следующем этапе участников просят развивать идеи, предложенные другими, например комбинируя элементы двух или трех предложенных идей.

На завершающем этапе производится отбор лучшего решения, исходя из экспертных оценок.

Были проведены многочисленные экспериментальные исследования, с целью сравнения количества и качества идей, созданных группами в процессе мозгового штурма и людьми, работающими индивидуально. Результаты свидетельствуют о том, что при условии правильного применения данной техники интерактивные группы нередко генерируют большее количество значимых идей, чем отдельные индивиды. Однако на сегодняшний день не существует доказательств в пользу более высокого качества идей, генерируемых группами.

В последние годы широкое распространение получил «электронный мозговой штурм» (*online brainstorming*), использующий интернет-технологии. Он позволяет почти полностью устранить «боязнь оценки», т.к. обеспечивает анонимность участников в сети, а также дает возможность решить ряд проблем традиционного мозгового штурма. К последним, в частности, относится т.н. «блокирование продуктивности»: поскольку участники группы представляют идеи поочередно, то люди в ожидании своей очереди могут передумать или испугаться публично высказывать свою идею, либо просто ее забывают.

Мозговой штурм дает возможность объединить в процессе поиска решений очень разных людей; а если группе удастся найти решение, то ее участники обычно становятся стойкими приверженцами его реализации. В настоящее время данный метод может быть эффективно использован организациями для улучшения качества работы в командах.

1.2 Этапы и правила мозгового штурма

Правильно организованный мозговой штурм включает три обязательных этапа. Этапы отличаются организацией и правилами их проведения:

1. Постановка проблемы. Предварительный этап. В начале этого этапа проблема должна быть четко сформулирована.

Происходит отбор участников штурма, определение ведущего и распределение прочих ролей участников в зависимости от поставленной проблемы и выбранного способа проведения штурма.

2. Генерация идей. Основной этап, от которого во многом зависит успех (см. ниже) всего мозгового штурма. Поэтому очень важно соблюдать правила для этого этапа:

- Главное – количество идей. Не делайте никаких ограничений.
- Полный запрет на критику и любую (в том числе положительную) оценку высказываемых идей, так как оценка отвлекает от основной задачи и сбивает творческий настрой.
- Необычные и даже абсурдные идеи приветствуются.
- Комбинируйте и улучшайте любые идеи.

3. Группировка, отбор и оценка идей. Этот этап часто забывают, но именно он позволяет выделить наиболее ценные идеи и дать окончательный результат мозгового штурма. На этом этапе, в отличие от второго, оценка не ограничивается, а наоборот, приветствуется. Методы анализа и оценки идей могут быть очень разными. Успешность этого этапа напрямую зависит от того, насколько «одинаково» участники понимают критерии отбора и оценки идей.

Мозговой штурм должен включать в себя следующие этапы.

1. Внятная постановка цели. Участники мозгового штурма должны понимать, зачем они здесь собрались. Поставь задачу команде. Или несколько задач, если это требуется. Но не переусердствуй. Посмотри, как правильно ставить цели. Без внятных целей **мозговой штурм** может оказаться безрезультатным.

2. Подбор людей, которые будут участвовать в мозговом штурме. Желательно подобрать разношерстный состав участников (с разным опытом работы). Ключевую роль в проведении мозгового штурма отводится руководителю (модератору) процесса. А самих участников мозгового штурма не должно быть слишком много, а их положение внутри компании должно быть приблизительно равноправным. Это нужно, чтобы все вели себя более раскованно. Не помешает пара маркерных досок. А также нужны будут люди, которые будут вести протокол. Но как вариант можно воспользоваться диктофонами и видеокамерами. Но для надежности желательно, чтобы ключевые идеи кто-то записывал, а также следил за реакцией участников мозгового штурма.

3. **Непосредственно мозговой штурм.** Мозговой штурм можно начать вступлением, которое должно быть коротким. Назовем это брифингом. 10 минут должно быть достаточно, чтобы поставить задачу и сделать вступительное слово. Дальше идет сам процесс обсуждения и фиксации идей на бумаге. В конце мозгового штурма можно классифицировать и структурировать полученные идеи наподобие того как материалы структурируются в научных статьях. Похожие идеи объединяются в кластеры. В итоге можно оценить идеи, но нужно воздерживаться от критики. Особенно это касается неконструктивной критики. Почитайте материал про правила конструктивной критики.

4. В итоге участники мозгового штурма выбирают идеи, которые предстоит воплощать в жизнь команде в самое ближайшее время. Идет обсуждение идей. Возможно, что ни одна из идей так и не будет принята. Или же может быть принято сразу несколько идей. Это уже зависит от того, насколько команда хорошо поработала.

1.3 Мозговые атаки

Для проведения мозговой атаки обычно создают две группы:

- участники, предлагающие новые варианты решения задачи;
- члены комиссии, обрабатывающие предложенные решения.

Различают индивидуальные и коллективные мозговые атаки.

В мозговом штурме участвует коллектив из нескольких специалистов и ведущих. Перед самым сеансом мозгового штурма ведущий производит четкую постановку задачи, подлежащей решению. В ходе мозгового штурма участники высказывают свои идеи, направленные на решение поставленной задачи, причём как логичные, так и абсурдные. Если в мозговом штурме принимают участие люди различных чинов или рангов, то рекомендуется заслушивать идеи в порядке возрастания ранжира, что позволяет исключить психологический фактор «соглашения с начальством».

В процессе мозгового штурма, как правило, вначале решения не отличаются высокой оригинальностью, но по прошествии некоторого времени типовые, шаблонные решения исчерпываются, и у участников начинают возникать необычные идеи. Ведущий записывает или как-то иначе регистрирует все идеи, возникшие в ходе мозгового штурма.

Затем, когда все идеи высказаны, производится их анализ, развитие и отбор. В итоге находится максимально эффективное и часто нетривиальное решение задачи.

Успех мозгового штурма сильно зависит от психологической атмосферы и активности обсуждения, поэтому роль ведущего в мозговом штурме очень важна. Именно он может «вывести из тупика» и вдохнуть свежие силы в процесс.

1.4 Преимущества мозгового штурма

Методика мозгового штурма полагает ряд следующих преимуществ по сравнению с другими методами анализа и генерации идей:

– посредством совместной деятельности специалистов, который отличаются друг от друга опытом, знаниями, видению ближайшего будущего, создаются необходимые условия для синергетического эффекта → качественного умножения знания (целое есть больше, чем набор частей); также новые подходы, перспективы видения и интересные аналогии возникают «на стыках различных дисциплин, областей человеческой практики» в ходе обсуждения поставленных проблем качественно отличающимися специалистами;

– сам ход обсуждения во время поиска путей решения поставленной задачи наделяется «особой энергетикой творчества, духом карнавального коллективизма, игры и праздника»;

– доброжелательная обстановка позволяет участникам усвоить навыки критики по существу, научиться импровизировать, а также усиливает положительный настрой и доверие.

Требования к участникам

1. Требования к ведущему всего мозгового штурма:

Ведущий регулирует ход мозгового штурма.

1. Отвечает за процедуру и регламент работы (соблюдение правил). Прерывает неконструктивное поведение участников – переговоры между группами во время работы в группе.

2. Обеспечивает психологическую поддержку участников, атмосферу активности и доброжелательности.

2. Требования к ведущему группы:

Ведущий обеспечивает качество регулирования хода мозгового штурма в группе.

1. Отвечает за процедуру и регламент работы (соблюдение правил). Прерывает неконструктивное поведение участников – «перетягивание каната», «павлиний хвост» и др.

2. Обеспечивает психологическую поддержку участников, атмосферу активности и доброжелательности (должен уметь мгновенно оценить любую идею).

3. Управляет процессом поиска идей (необходимо постоянно уточнять формулировки задачи, расширять поле поиска, выделять новые направления и аспекты решений, задавать новые области поиска решений для участников).

3. Требования к генератору идей:

Задача генераторов идей: непрерывно выдвигать идеи по поставленной проблеме или задаче. Требования к генераторам: выдвижение по широкому спектру тем большого количества идей, основанных на новых принципах, перенос идей из различных областей, использование ярких неожиданных аналогий.

Использование ранее выдвинутых идей другим генератором. Должна быть присуща вера в то, что лучшие идеи ещё впереди, оптимизм. Недопустимо включать в группу генераторов прирождённых скептиков и критиканов.

4. Требования к критику-аналитику:

Задача критика–аналитика: выявление рационального зерна в каждой предложенной к анализу идее на основе глубокого знания теории и практики, понимания специфики проблемы (задачи), способности к обобщению. Ему должна быть присуща вера в то, что лучшая идея – это та, которая анализируется в данный момент, оптимизм.

Хорошие аналитики могут выявить новые принципы решения задачи после классификации принципов, выдвинутых генераторами. Очень часто самое ценное в мозговом штурме – новое направление поиска, а не конкретное решение.

2 ЗАДАНИЕ

Разделиться на группы по 4 – 5 человек и получить задание согласно варианта в таблице 4.1.

Таблица 4.1

Темы заданий	
Номер варианта	Тема для мозгового штурма
1	Фантастический мир
2	Новый суперкар
3	Выдуманный герой
4	Для чего может пригодиться гвоздь (15 вариантов)
5	Где и как заработать первый миллион
6	Как заставить животное пить воду
7	Организм требует пищи
8	Кто обитает на дне океана
9	Человек человеку ...
10	Как быть и кто виноват?

3 ТРЕБОВАНИЯ К ОТЧЕТУ

В отчете должны быть отображены следующие пункты:

1. Задание.
2. Краткие теоретические сведения.
3. Результат выполнения задания для своего варианта.
4. Заключение по результатам работы.

ПРАКТИЧЕСКАЯ РАБОТА № 5 РАБОТА В SCRATCH. ПРОТОТИПИРОВАНИЕ КОМПЬЮТЕРНОЙ ИГРЫ.

1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Установка и регистрация

Scratch – это новая бесплатная среда программирования, которая позволяет пользователю создавать собственные истории, мультфильмы, игры, презентации и другие произведения. *Scratch* специально разрабатывался как учебная среда для обучения пользователей начальному уровню программирования. Программы создаются из разноцветных блоков точно так же, как собираются из разноцветных кирпичиков конструкторы Лего.

Scratch разработан в 2006 г. под руководством Митчела Резника группой *Lifelong Kindergarten* в лаборатории *Media Lab* Массачусетского технологического института.

Существует два способа работы в *Scratch*. Можно работать в браузере, а можно установить оффлайновый редактор и делать программы не подключаясь к интернет. Если Вы планируете работать используя интернет, то прочитайте первую часть введения. Если Вы планируете работать без подключения к интернету, то можете перейти сразу ко второй части введения.

Для получения возможности создавать свои проекты на *Scratch*, Вам необходимо зарегистрироваться на сайте *scratch.mit.edu*. Откроется окно регистрации, в котором в верхнее окно надо ввести Ваше имя для сайта *Scratch* (иногда это имя называют «логин»). Разработчики советуют не вводить настоящее имя.

После регистрации наверху справа Вы увидите Ваш логин и некий конвертик с цифрой один, которая означает, что в почтовом ящике уже есть одно письмо. Кликнув на конвертик, Вы можете прочитать сообщение.

Чтобы скачать и установить оффлайновый редактор *scratch*, нужно перейти по адресу <http://scratch.mit.edu/scratch2download/>. Далее выбрать операционную систему совпадающую с Вашей и следовать стандартным инструкциям установки.

Запустите *Scratch 2* через ярлык на рабочем столе.

Откроется окно программы. Разверните его на весь экран, нажав, как обычно, на квадратик в правом верхнем углу.

После установки на рабочем столе появится симпатичный ярлык.

1.2 Запуск *Scratch*. Начало работы

Запустите *Scratch 2* через ярлык на рабочем столе.

Откроется окно программы (рисунок 5.1). Разверните его на весь экран, нажав, как обычно, на квадратик в правом верхнем углу.

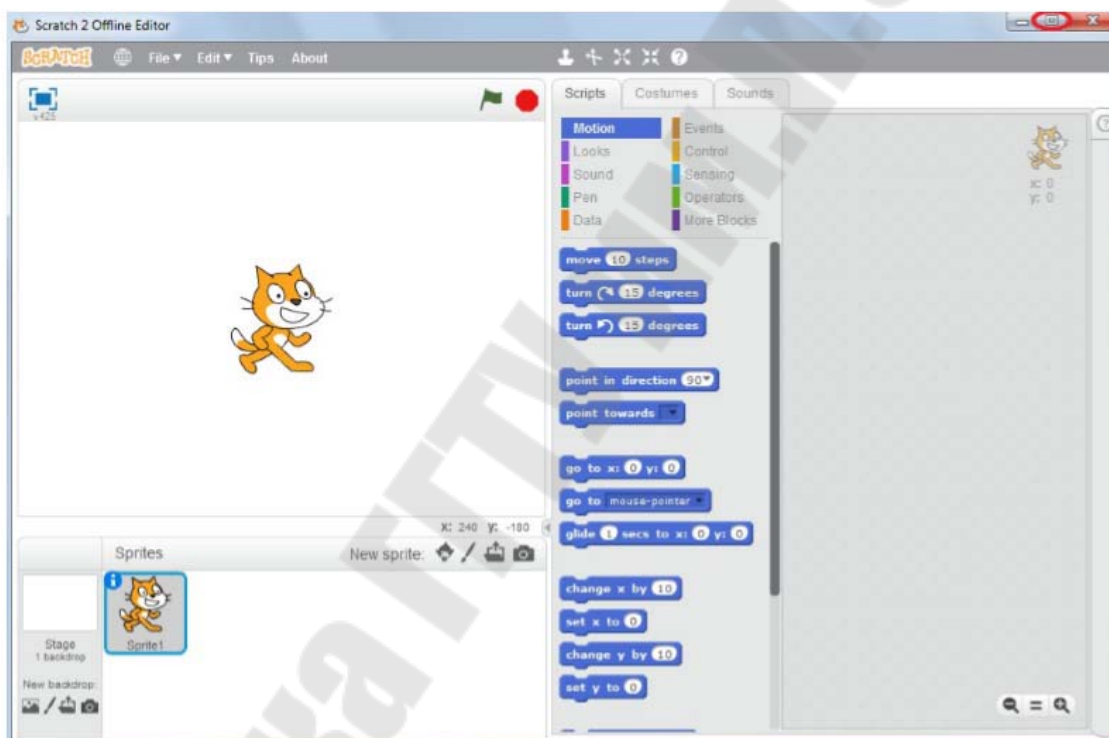


Рисунок 5.1 – Рабочее окно программы

После запуска приложения видно, что *Scratch 2* состоит из трёх частей.

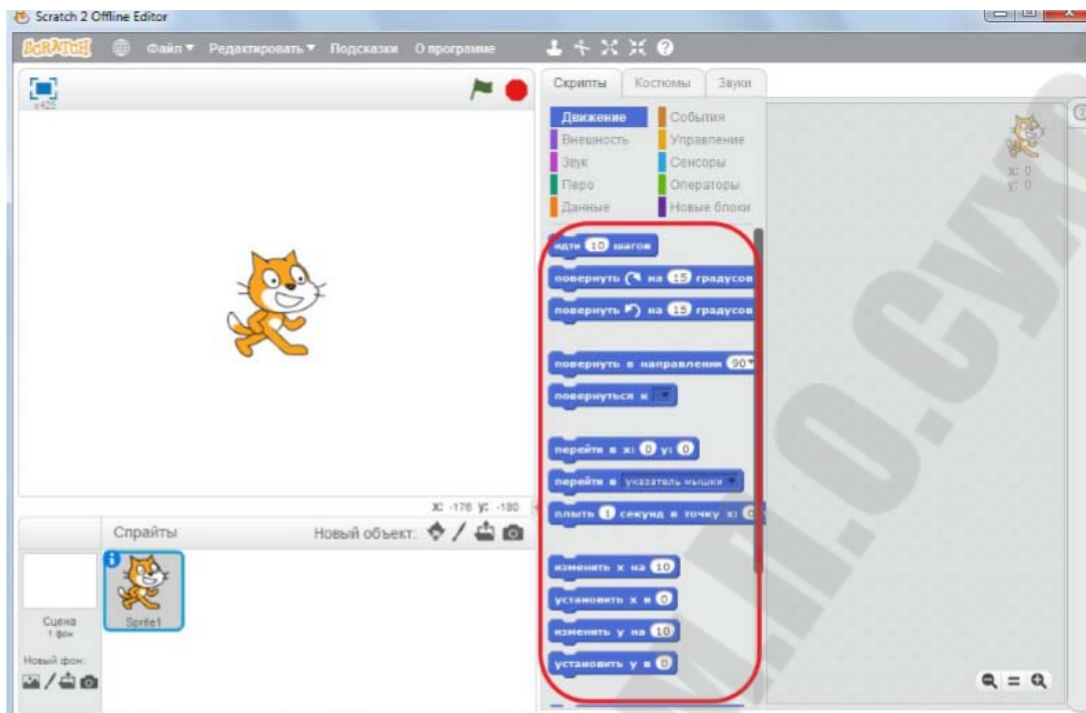


Рисунок 5.2 – Область доступных команд программы

Слева находится Сцена (рисунок 5.3), на которой будет работать программа. На Сцене, как в театре, будут происходить все приключения персонажей.

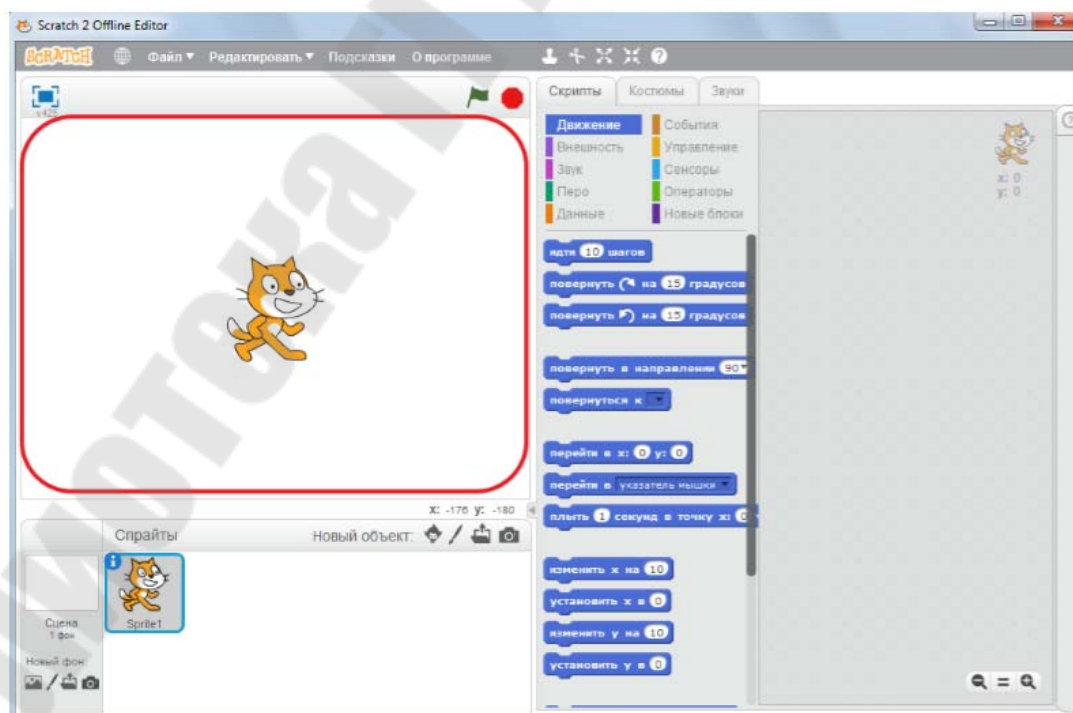


Рисунок 5.3 – Сцена программы

Область спрайтов доступных конкретному проекту спрайтов (рисунок 5.4).

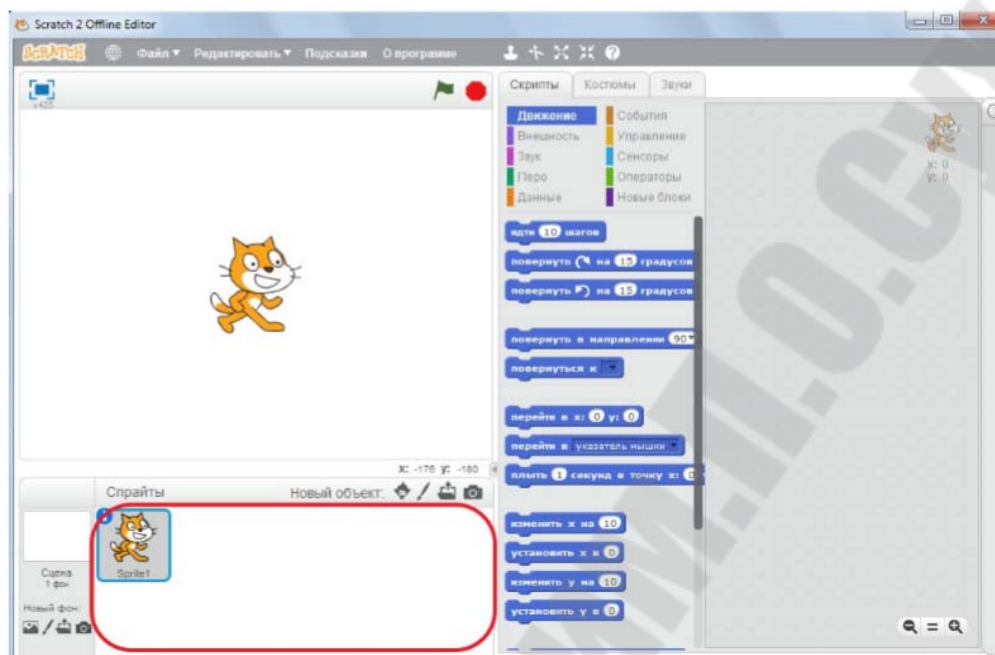


Рисунок 5.3 – Область спрайтов

Справа большое серое пустое место – область, где пользователь будет складывать программу из блоков (рисунок 5.4), это рабочая область или область скриптов.

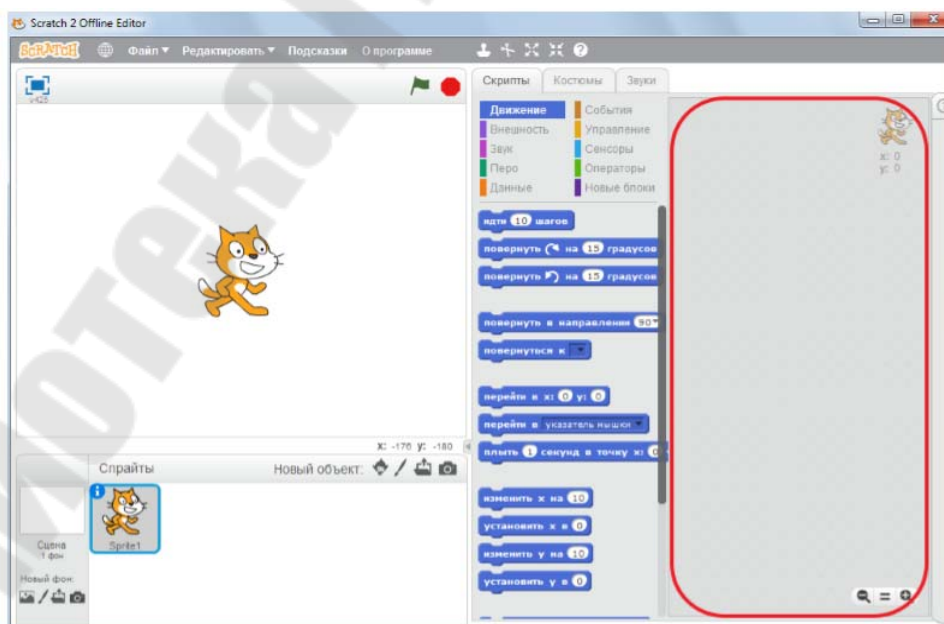


Рисунок 5.4 – Рабочая область скриптов

Основа *Scratch* – разноцветные блоки обозначающие соответствующую группу действий (движение, внешность, звук и т.д.). В основном используются синие, жёлтые, фиолетовые и голубые (рисунок 5.5). При включении одной кнопки все остальные выключаются. Включенная кнопка вся окрашивается в соответствующий ей цвет. При этом в нижней ячейке первого столбца появляются команды, связанные с включенной кнопкой. У всех кнопок разные привязанные к ним команды.

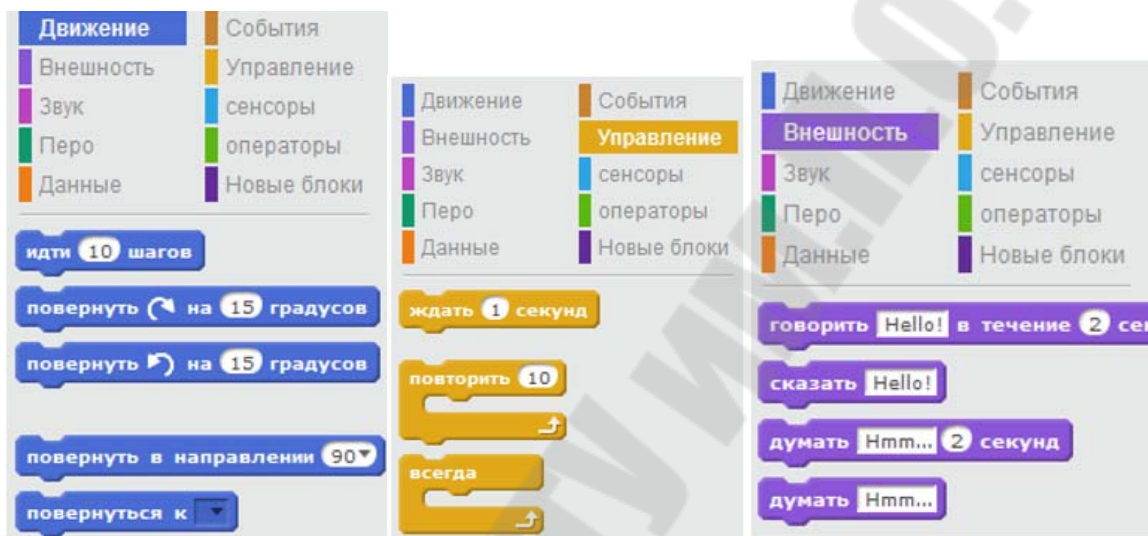


Рисунок 5.5 – Разноцветные группы блоков действий

Спрайтами в *Scratch* называются все объекты по умолчанию, отличаются лишь их номера. Обычно предполагается, что мы сами их будем переименовывать и называть более адекватными именами.

Над панелью выбора группы действий находятся три кнопки–вкладки – скрипты, костюмы и звуки. От того какая из них нажата, зависит ячейка внизу. Если нажата кнопка скрипты, то нижняя ячейка второго столбца покажет программы (скрипты) для объекта, который отображен в верхней ячейки. Кнопки костюмы и звуки позволяют настраивать и менять соответственно внешний вид объекта и издаваемые им звуки. Оставьте включенной кнопку скрипты.

1.3 Одновременное выполнение скриптов (программ)

Когда несколько объектов в программе двигаются вместе, одновременно (при нажатии на кнопку запуска) можно говорить об одновременном (параллельном) выполнении разных блоков команд (скриптов). В серьезном программировании используются другие,

более умные слова (многопоточность и т.п.). При этом в Scratch можно сделать так, что два (или больше) скрипта одновременно будет выполнять вообще один объект. Допустим, кот будет шагать и при этом менять свои размеры, форму и другие свойства. При этом ходьба и изменение свойств друг с другом никак не будут связаны. Сделать это можно, например, разместив два разных скрипта в ячейке скрипты для кота (рисунок 5.6).

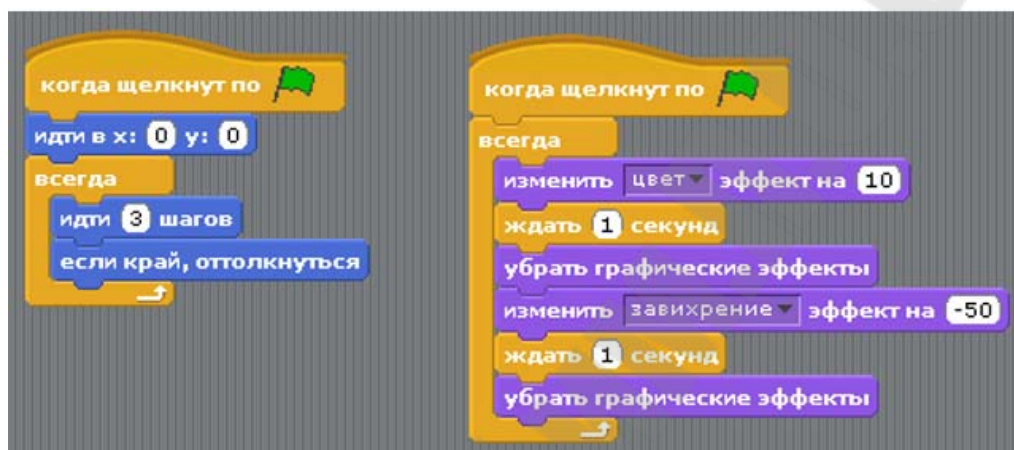


Рисунок 5.6 – Одновременное выполнение скриптов

При щелчке на кнопке запуска начнут работать оба скрипта сразу. Первый будет заставлять кота ходить слева направо. Это нам уже знакомо. Второй скрипт будет менять внешний вид кота. Фиолетовые команды связаны с кнопкой внешность. Во втором скрипте выше используется команда «изменить ... эффект на ...» После слова «изменить» в раскрывающемся списке можно выбрать понравившийся нам эффект, а в поле с числом прописать, на сколько единиц его изменять. Команда убрать графические эффекты возвращает объект к его исходному внешнему виду. Итак, второй скрипт в цикле всегда выполняет следующие действия: «изменяет цвет объекта» → оставляет объект в таком состоянии на 1 секунду → возвращает объект к прежнему цвету → искривляет объект с помощью завихрения → оставляет в таком состоянии на 1 секунду → возвращает к исходной форме.

1.4 Последовательное выполнение скриптов (программ)

При последовательном выполнении скриптов сначала все действия должен совершать один объект, затем второй и т.д. Как же это можно сделать в *Scratch*?

Например, можно просто использовать команду ждать в начале скрипта второго объекта. Но на самом деле это будет никакое не последовательное выполнение скриптов, т.к. оба они начнут работать одновременно, просто у второго объекта сначала будет долго работать команда ждать. Хотя на сцене можно таким образом получить желаемый эффект: сначала действия будет совершать один объект, а через некоторое время – другой.

Рассмотрим, более грамотный способ организации последовательного выполнения скриптов. Когда один объект завершает выполнение ряда своих команд, он должен подать какой-нибудь сигнал-сообщение второму объекту, а тот, в свою очередь, должен его принять. В *Scratch* для этого есть две специальные команды, связанные с кнопкой контроль: «передать ...» и «когда я получу» .



Рисунок 5.7 – Последовательное выполнение скриптов (команды для первого спрайта)

Команда «передать ...» дается объекту, который уже закончил все или некоторые свои действия, а «когда я получу ...» применяется к объекту, который начинает работать вторым. Вместо трех точек в

этих командах вписывается сообщение, которые мы сами назначаем. Организуем для двух объектов последовательное выполнение скриптов. Допустим, по нашему сценарию планируется такая анимация: кот сначала ходит туда-сюда, после чего исчезает, и по экрану начинает метаться сыр (например). Если у вас сейчас в ячейке скриптов кота две программы, то удалите ту, которая меняет внешний вид объекта, чтобы она нас никак не вводила в заблуждение. Первый скрипт кота надо изменить (рисунок 5.7).

В команде «передать ...» надо вставить сообщение. Для этого открывается специальное окно, куда можно его вписать. Команда «спрятаться» используется для того, чтобы скрыть кота, после того, как он пошлет сообщение.

Команда «показаться» в начале скрипта требуется, чтобы снова отобразить объект, если он был скрыт до этого. Понятно, что при первом выполнении скрипта она не играет никакой роли. Цикл всегда был заменен на цикл «повторить ...», т.к. нам больше не надо, чтобы объект двигался всегда. Соберите такой скрипт. Теперь посмотрим, что будет делать второй объект. Он должен получить сообщение, после чего совершить какие-нибудь действия (рисунок 5.8).

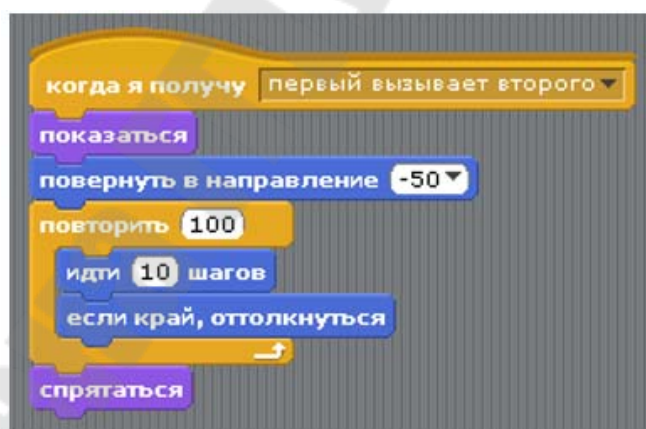


Рисунок 5.8 – Последовательное выполнение скриптов (команды для 2го спрайта)

Таким образом, второй скрипт не начнет выполняться, пока первый не передаст ему сообщение «первый вызывает второго». Обратите внимание, что команда «спрятаться» первого объекта находится ниже, чем «передать ...». Это значит, что у нас не совсем последовательное выполнение скриптов. Ведь первый еще не завершается, когда начинается второй. В принципе, чаще именно так и бывает.

2 ЗАДАНИЕ

2.1 Напишем программу реализующую бегущего Котёнка.

Выбираем вверху блоки «События» и тянем в центр экрана блок «когда клавиша пробел нажата» и там его бросаем (рисунок 5.9). Чтоб тянуть блок, надо нажать на него мышкой и тянуть, а не просто щёлкнуть.

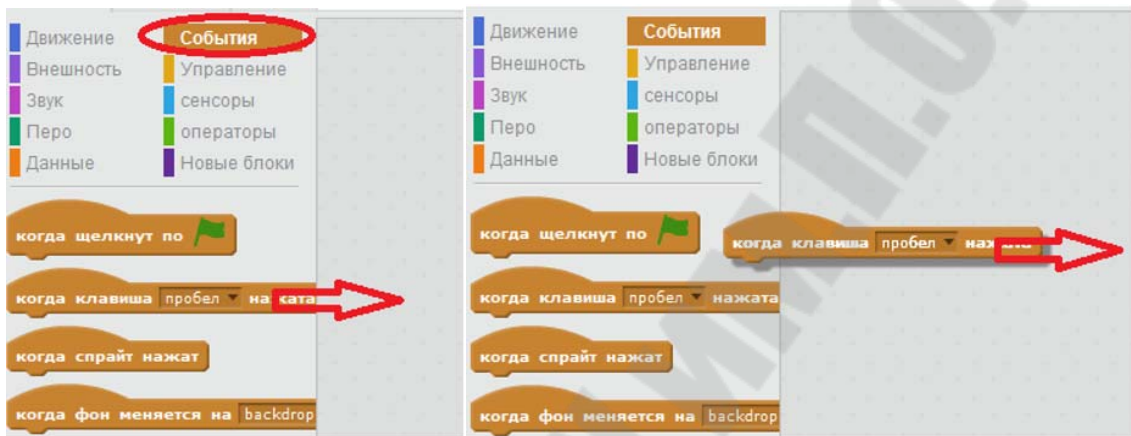


Рисунок 5.9 – Начало написания программы

Вытащите блок вправо и бросьте его, отпустив кнопку мышки. Теперь выбираем синие блоки «движение», тянем направо блок «идти 10 шагов» (рисунок 5.10).

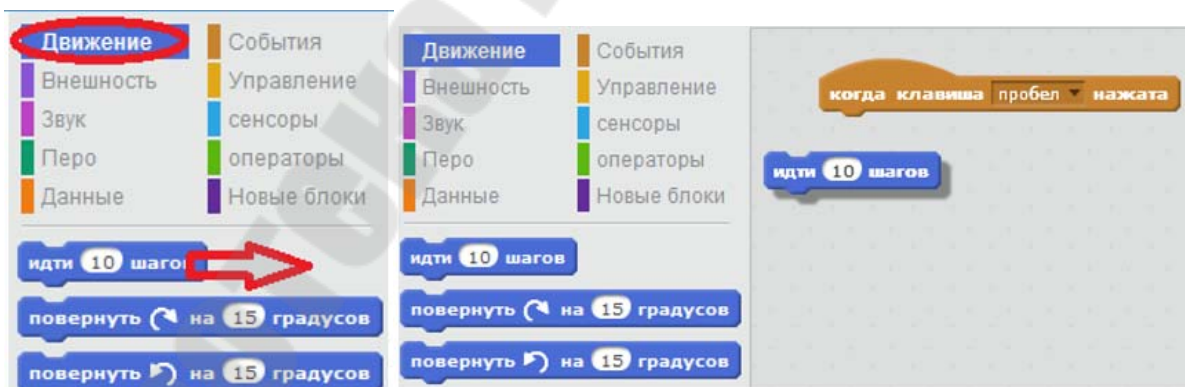


Рисунок 5.10 – Добавление «движения»

И там его прицепляем под коричневый блок «когда клавиша пробел нажата». Когда синий блок хочет прицепиться к коричневому

– появляется белая полоса. Отпускаем кнопку мышки, и он прицепится.

Получилась первая программа или скрипт, как её иногда называют (рисунок 5.11). Нажимайте пробел и посмотрите, как двигается котик.

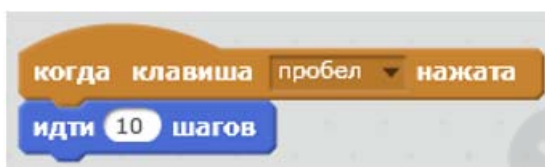


Рисунок 5.11 – Первый скрипт

Нажимайте пробел и посмотрите, как двигается котик. Если нажимать долго, то он дойдет до края экрана, и почти весь спрячется. Торчит только хвост.

Теперь добавим в программу звук «мяу»(рисунок 5.12). Нажмите на малиновые блоки «звук» и тащите блок «играть звук *meow*» под синий блок «идти 10 шагов». «*Meow*» по-английски означает «мяу». Прицепите малиновый блок снизу.

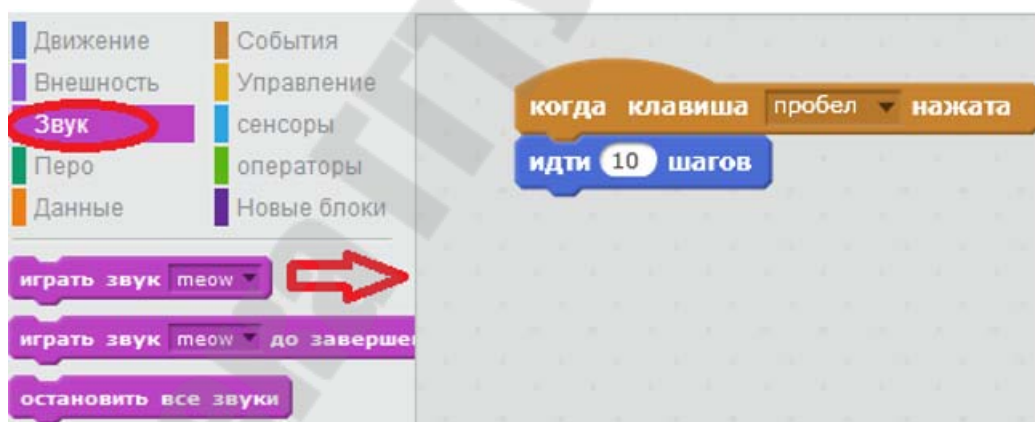


Рисунок 5.12 – Добавление звуков

Теперь при нажатии пробела Кот идет и мяукает. Если он мяукает слишком громко, то отрегулируйте громкость, например, нажав на значок громкости в правом нижнем углу *Windows*, и подвинув регулятор громкости пониже.

2.2 Игра лабиринт

Сделаем игру в которой Гига должна будет пройти по лабиринту и найти Нано, который заблудился в пещере. Сначала сделаем простой вариант игры, а потом усложним его, добавив анимацию. Создайте новый проект. Удалите Кота, и добавьте из библиотеки Гигу и Нано (рисунок 5.13). Посмотрите, какие костюмы есть у Гиги. Выберите её и нажмите на вкладку Костюмы.



Рисунок 5.13 – Персонажи (спрайты) игры

У Гиги четыре костюма, три весёлых и один грустный. У Нано тоже четыре костюма, три весёлых и один грустный (рисунок 5.14).

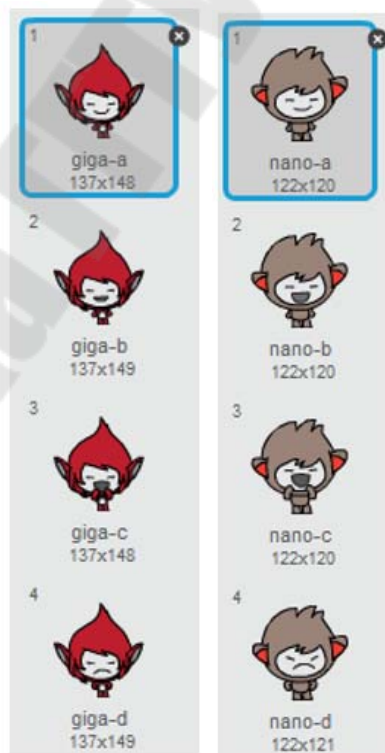


Рисунок 5.14 – Костюмы персонажей игры

Запомните названия первых весёлых костюмов и названия грустных, они нам пригодятся. Затем нарисуйте лабиринт. Выберите

сцену. Перейдите на вкладку Фоны. Выберите инструмент Линия (рисунок 5.15).

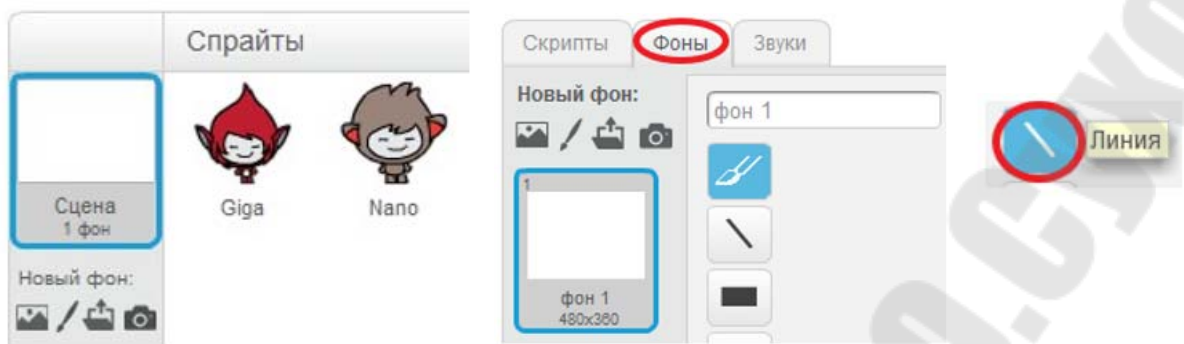


Рисунок 5.15 – Инструменты для создания лабиринта

Выберите цвет на соответствующей панели. Увеличьте толщину линии. Нарисуйте лабиринт (рисунок 5.16).



Рисунок 5.16 – Создание лабиринта

Сделайте для Гиги такую программу (рисунок 5.17). Здесь использованы знакомые блоки «перейти в указатель мышки», «показать», «скрыть», а также новый голубой сенсорный блок «касается цвета». Он используется совместно с блоком «если» и срабатывает в момент касания Гигой стенки лабиринта.



Рисунок 5.17 – Скрипт для Гиги

Для того чтобы поместить в голубой блок коричневый цвет используется очень простой способ. Кликните мышкой в квадратное окошко с цветом. Курсор измениться, и станет похож на руку. Кликните им на стену лабиринта. Выбранный цвет автоматически попадёт в голубой блок.

Теперь сделайте программу для Нано (рисунок 5.18).

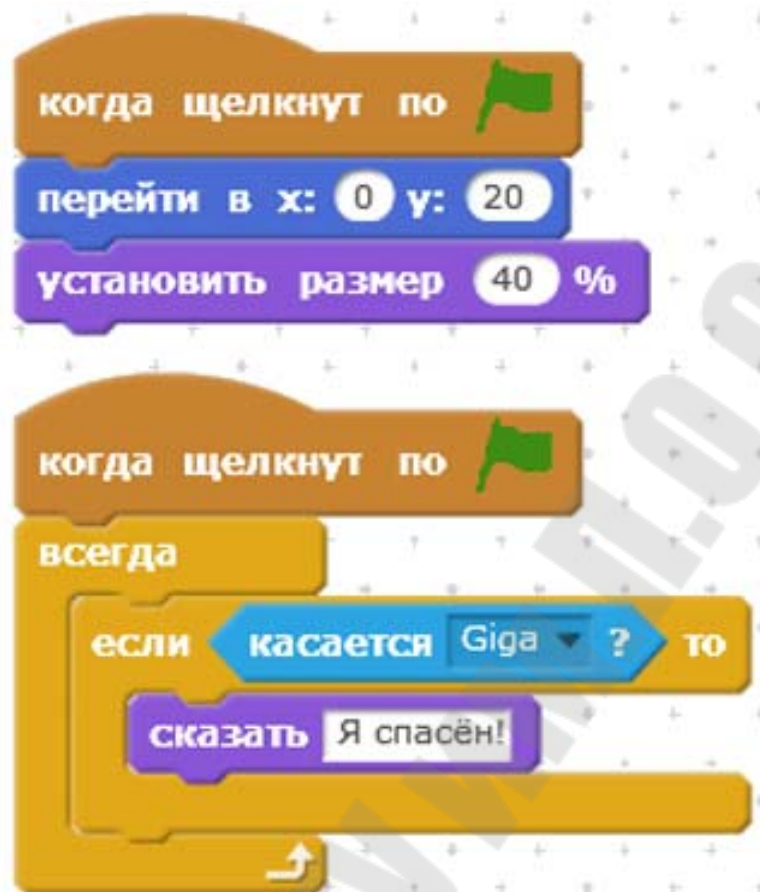


Рисунок 5.18 – Скрипт для Нано

Запустите программу зелёным флажком, аккуратно двигайте Гигу по лабиринту. Старайтесь не касаться стен, а то проиграете. Когда вы дойдёте до Нано, то он обрадуется и скажет: «Я спасён!». Теперь давайте добавим в нашу игру эмоций. Пусть в начале игры у Гиги и у Нано будут грустные выражения лиц, а в момент долгожданной встречи их озарит улыбка. А ещё пусть они запрыгают от радости.

Для того, чтобы сделать Нано и Гигу грустными, надо добавить блок «сменить костюм на». В начале программы, когда нажат зелёный флажок, они должны сменить костюмы на грустные, а когда встретятся – на весёлые. Радостные прыжки персонажей можно сделать с помощью следующих блоков (рисунок 5.19).



Риснок 5.19 – Радостные прыжки персонажей

Этот кусочек скрипта работает следующим образом, при изменении Y на 10 персонаж подпрыгнет, подождёт одну пятую часть секунды и опустится вниз, изменив Y на -10. Опять подождёт 0.2 секунды, и снова подпрыгнет. Всего будет совершено 10 прыжков. В итоге скрипты программы Гиги станут такими (риснок 5.20).

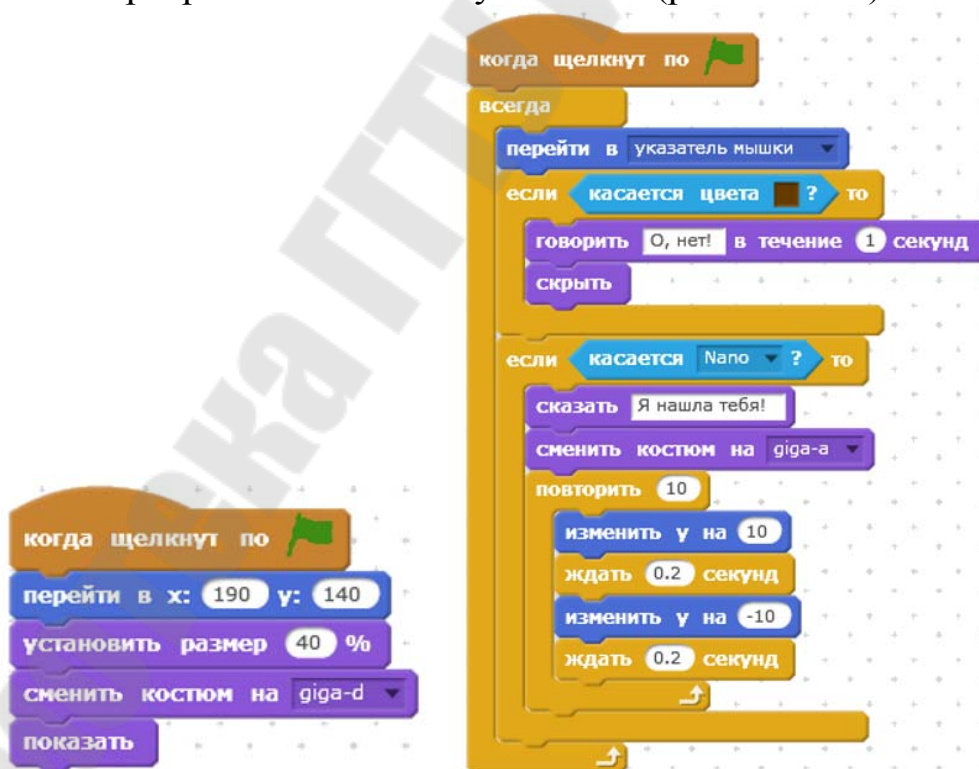


Рисунок 5.20 – Скрипты (первый и второй) программы Гиги

Скрипты программы Нано станут такими (рисунок 5.21).

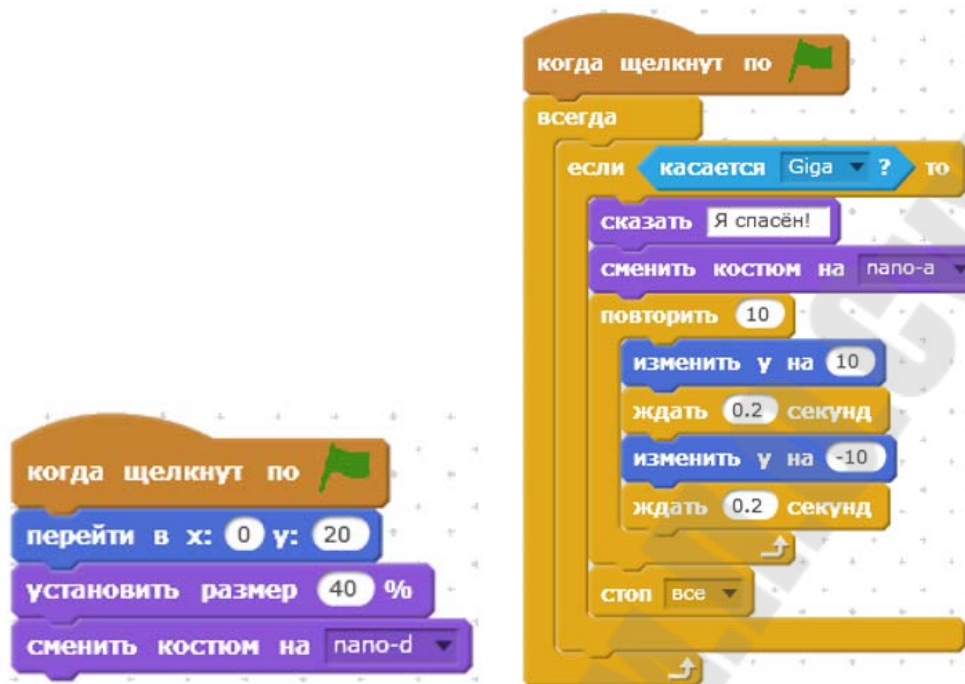


Рисунок 5.21 – Скрипты (первый и второй) программы Нано

Проект готов. Нажмите на зелёный флажок и попробуйте пройти Гигой до центра лабиринта.

2.3 Задания для самостоятельной работы

На основании выбранного концепта (таблица 5.1) продумать часть игры и реализовать ее прототипирование в соответствии с вариантом.

Таблица 5.1

Темы для создания прототипа компьютерной игры

Номер варианта	Тематика анимации
1	2
1	Полет летающей тарелки
2	Прыжки на батуте
3	Разъезд на перекрёстке (регулируемом)
4	Разъезд на перекрёстке (нерегулируемом)
5	Танец
6	Отскок мячика
7	Рикошет пули
9	Том и Джерри (кошки-мышки)
10	Хоровод

Продолжение табл. 5.1

1	2
11	Краш-тест автомобиля
12	Рост цветка с момента посадки
13	Замедление вращения юлы
14	Цветовое сопровождение музыки
15	Работа турникета
16	Бег с препятствиями
17	Игра кота с лазерной указкой
18	Игра на гитаре
19	Игра на барабане
20	Поход в кино

3 ТРЕБОВАНИЯ К ОТЧЕТУ

В отчете должны быть отображены следующие пункты:

1. Задание.
2. Краткие теоретические сведения.
3. Результат выполнения задания для своего варианта (продемонстрировать результаты в *scratch*).
4. Заключение по результатам работы.

ЛИТЕРАТУРА

1. Финни К. 3D-игры: Все о разработке + приложение / Финни К. – БИНОМ. Лаборатория знаний, 2014. – 976 с.
2. Кукин Д.П. Индустрия компьютерных игр. Лабораторный практикум: пособие/ Д.П. Кукин, Т.А. Рак, О.О. Шатилова – Минск, БГУИР 2019. – 60 с.
3. Schell J. The Art of Game Design: A book of lenses / Schell J. – CRC Press, 2008. – 435 с.
4. Кукин Д.П. Индустрия компьютерных игр. Лабораторный практикум: пособие/ Д.П. Кукин, Т.А. Рак, О.О. Шатилова – Минск, БГУИР 2019. – 60 с.
5. Голиков Д., Голиков А. Программирование на Scratch 2. Подробное пошаговое руководство для самостоятельного изучения ребёнком. – Scratch4russia.com, 2014. – 283 с.
6. Rogers S. Level Up! The Guide to Great Video Game Design / Rogers S. – Wiley, 2014. – 520 с.
7. Csikszentmihalyi M. Flow: The Psychology of Optimal Experience / Csikszentmihalyi M. – Harper Perennial Modern Classics, 2008. – 319 с.
8. Хокинг, Д. Unity в действии: мультиплатформенная разработка на C# / Джозеф Хокинг ; [пер. с англ. И. Рузмайкина]. – Спб. [и др.]: Питер, 2018.– 334 с. – (Для профессионалов).

**Соболев Денис Викторович
Семенченя Татьяна Сергеевна**

ИНДУСТРИЯ КОМПЬЮТЕРНЫХ ИГР

**Практикум
по одноименной дисциплине для студентов
специальности 1-40 05 01 «Информационные
системы и технологии (по направлениям)»
дневной формы обучения**

Подписано к размещению в электронную библиотеку
ГГТУ им. П. О. Сухого в качестве электронного
учебно-методического документа 31.03.21.

Рег. № 23Е.
<http://www.gstu.by>