



НТО ЮНИОРЫ 2026/27 · СФЕРА

Игротех

Быстрый старт: главное для подготовки к
отборочному этапу

5–7 класс



Автор-составитель

Артём Осинowskiй, методист и разработчик детского технопарка «Кванториум»

Детский технопарк «Кванториум», Новосибирск · 2026

Игротех

Быстрый старт: главное для подготовки к отборочному этапу НТО Юниоры. 5–7 класс

Автор-составитель – **Артём Осиновский**, методист и разработчик детского технопарка «Кванториум».

Нашли ошибку или неточность? Напишите на osinovsky.aa@kvantorium54.ru (рабочая почта): сфера, версия книги (1.0) и номер страницы – исправим в следующей версии. Актуальная версия и список изменений – nto.kvantorium54.ru/knigi. Личная почта для других вопросов – artem081033@gmail.com.

Важно. Это не официальное пособие олимпиады, а книга методиста «Кванториума». Каких заданий ждать на отборе 2026/27, мы не знаем: организаторы их заранее не показывают, а задания разных лет не повторяются. Книга готовит к темам из программы сферы и приёмам решения, а не к конкретным задачам.

Книга помогает школьнику самостоятельно подготовиться к отборочному этапу сферы «Игротех» Национальной технологической олимпиады «НТО Юниоры»: теория с нуля, подробные разборы задач и задачи для самопроверки с ответами.

Большинство задач составлены автором по образовательной программе сферы, часть – по мотивам открытых заданий прошлых лет; источник указан у каждой задачи. Уровни сложности – шкала этой книги, а не баллы олимпиады. Правила, сроки и формат отбора – на сайте junior.ntcontest.ru.

Это «Быстрый старт» – сжатая версия книги: по каждой главе главное, один разбор и четыре задачи с ответами. Для подготовки к отбору рекомендуем полную версию – все разборы, задачи и полный справочник, главы те же: nto.kvantorium54.ru/knigi.

Быстрый старт · версия 1.0 от 02.10.2026, сборка 02.10.2026. © Артём Осиновский, 2026.

Содержание

Как устроена эта книга	3
Для кого эта книга	3
Как устроен отбор	3
Что пригодится на отборе и где это в книге	4
Как пользоваться книгой	5
Как вводить ответы на отборе	5
Что понадобится	6
План на месяц до отбора	16
Родителям и наставникам	16
Честно об этой книге	17
Глава 1 Игры и жанры: язык геймдизайнера	18
Глава 2 Геймдизайн: механики, баланс, документ игры	23
Глава 3 Левел-дизайн: карты и пути	29
Глава 4 Логика, выборы и случайность	34
Глава 5 Машины состояний: как думает NPC	39
Глава 6 Переменные, условия и поиск ошибок	44
Глава 7 Сюжет и диалоги: язык Ink	50
Глава 8 3D и визуал: Blender	56
Ответы и решения	61
Справочник	64
Шпаргалка: всё «Запомни» из глав	64

Как устроена эта книга

Эта книга – тренажёр для отборочного этапа сферы «Игротех» олимпиады «НТО Юниоры». Здесь есть всё, чтобы готовиться самостоятельно: теория с нуля, подробные разборы задач и похожие задачи для самопроверки с ответами в конце книги. Мощный игровой компьютер не нужен: многие задачи решаются в тетради в клетку, а остальные – в браузере и в бесплатном Blender.

Для кого эта книга

- Для школьников **5–7 классов**, которые выбрали (или думают выбрать) сферу «Игротех».
- Для тех, кто любит играть, но ни разу не делал игру сам и не программировал. Каждая тема начинается с самого начала: что такое механика, чем состояние отличается от события, как устроена развилка в сюжете.
- Для **наставников и родителей**, которые хотят помочь. Для них есть отдельный раздел в конце введения.

Как устроен отбор

Таблица В.1

Когда	Что происходит
до 9 ноября	регистрация на олимпиаду (наш совет – успеть до 20 октября, тогда будут обе попытки)
20–30 октября	отбор, попытка 1 (3 часа)
31 октября – 10 ноября	отбор, попытка 2 (засчитывают лучшую из двух)
15 декабря	финал сферы «Игротех» – в один день во всех регионах

- Отбор проходит **дома, онлайн, с компьютера**. Задания открываются в личном кабинете на my.ntcontest.ru и решаются на платформе «Орбита» (orbita.education). Каждый решает сам за себя.
- Задания – это задачи с ответом, который проверяет компьютер: ввести число, выбрать верные варианты и записать их номера подряд, сопоставить, расставить шаги по порядку. В программе сферы среди проверочных заданий прямо названы тест на знание терминов и поиск ошибок в чужой программе – к обоим в книге много задач.
- Свой игровой проект для отбора не нужен. На вебинаре для наставников разработчики сферы сказали: на отборе и в финале участник работает с готовыми шаблонами. А ещё – что многое из того, что будет происходить на олимпиаде, связано с редактором игр «Урсула». Поэтому зайти в «Урсулу» заранее и пощёлкай по ней, пока есть время.
- За тест бесплатного курса своей сферы дают **+5 баллов**. Курс «Игротеха» – на платформе «Мастерская игр «Урсула»» (games.kruzhok.org), вход через «Талант». Бонус

нужно включить самому в личном кабинете («Мои привилегии») до конца отбора. Учти: после этого сферу уже не поменять. И бонус засчитают, только если за задания ты набрал больше нуля.

- У сферы есть проектные конкурсы «Идея и визуал» и «Геймдизайн и прототип». Для отбора они не обязательны, но это хорошая тренировка: придумать персонажа, собрать маленький прототип. Сроки и условия – на games.kruzhok.org, в разделе «Лаборатория».
- В октябре идёт фестиваль НТО Юниоры (junior.ntcontest.ru/fest). На вебинаре сферы обещали, что там будут пробные задания, – не пропусти.
- Финал – **командный** и очный. Команде выдают проект в редакторе игр: там уже есть сцена и объекты, а нужно добавить персонажей, запрограммировать их и добиться, чтобы игра в жанре «квест» заработала. Команду можно собрать заранее, а можно прямо на месте – организаторы финала распределяют участников сами.

Совет. Сроки и правила могут уточняться. Главный источник – сайт олимпиады junior.ntcontest.ru и памятка для Новосибирской области nto.kvantorium54.ru.

Что пригодится на отборе и где это в книге

На странице сферы перечислено, что пригодится на отборе. Вот где это в книге:

Таблица В.2

Что пригодится на отборе	Глава
Общие знания о том, какие бывают игровые механики, жанры, элементы игр; опыт игры в квесты	1. Игры и жанры: язык геймдизайнера
Продумывание механик, баланс ресурсов и сложности; вычисления по формулам, дроби, пропорции, проценты	2. Геймдизайн: механики, баланс, документ игры
Левел-дизайн: карты уровней и внимание игрока; эскизы и визуализация идей	3. Левел-дизайн: карты и пути
Базовые элементы логики и комбинаторики	4. Логика, выборы и случайность
Алгоритмы и исполнители; программирование поведения персонажей (ПРИМС из программы сферы)	5. Машины состояний: как думает NPC
Переменные и условия; отладка программ и поиск ошибок в логике; модульное программирование	6. Переменные, условия и поиск ошибок
Придумывать истории про персонажей; разветвлённый сюжет с выборами и последствиями, диалоги	7. Сюжет и диалоги: язык Ink
Опыт работы в 2D- и 3D-редакторах; Blender: модели, текстуры, экспорт в игру	8. 3D и визуал: Blender

Начни с глав 1 и 2: в них появляются слова, на которых говорит вся остальная книга. Главы 3 и 4 можно читать в любом порядке, для них хватит тетради в клетку и карандаша. Главу 5 читай раньше главы 6: в шестой мы добавляем к машинам состояний переменные и ищем в них ошибки. Главе 7 пригодится глава 4 – там разобраны развилки и выборы. Для главы 8 нужен компьютер с Blender, её можно начать в любой момент.

Как пользоваться книгой

В каждой главе – теория и разборы задач:

1. **Теория.** Коротко и по делу, с рисунками. Каждая новая тема начинается с большой картинкой «на пальцах», а самое важное собрано в рамках «Запомни». Все «Запомни» из всех глав повторяются в конце книги, в шпаргалке.
2. **Разбираем задачи.** Задачи такие же, как на отборе, и к каждой – решение по шагам: что дано, что делаем и почему, где обычно ошибаются, какой ответ получился и похож ли он на правду. Сразу после разбора – две похожие задачи **«Реши сам»**: сначала посмотри, как решают, потом реши сам похожую.
3. **Ответы** к «Реши сам» – в конце книги, в разделе «Ответы и решения». Там только ответ, а если в задаче есть подвох – ещё короткая подсказка.

Хочешь пробежаться по всей программе быстро – есть отдельная короткая книжка **«Быстрый старт»** (около 60 страниц): из каждой главы главное, один разбор и несколько задач «Реши сам».

Обычная задача – без пометки. У трудных задач рядом с номером стоит звёздочка (*): в них нужно несколько идей сразу, они труднее, чем на отборе. Лёгкие задачи, в одно действие, помечены словом «разминка».

Это уровни нашей книги, а не баллы олимпиады. Сколько задач будет на отборе и сколько баллов стоит каждая, заранее не объявляют.

Совет. Разбор читай с листком бумаги. Закрой решение, попробуй решить сам, а потом сверяйся по шагам. Так ты запомнишь в три раза больше, чем если просто прочитаешь.

У каждой задачи внизу написано, откуда она взялась. «По мотивам задачи» – значит, похожая задача была на олимпиаде прошлых лет: у школьников постарше, на профиле НТО «Разработка компьютерных игр» для 8–11 классов, или в прежней игровой сфере НТО Junior. Мы пересказали её своими словами, упростили и поменяли числа. «Своя задача по программе сферы» – значит, мы придумали её по теме из программы 2026/27 года.

Как вводить ответы на отборе

Так было на отборочных этапах прошлых лет. На «Орбите» перед каждой задачей читай инструкцию: там сказано, в каком виде нужен ответ.

- **Только число**, без единиц измерения и без знака процента, если не сказано иначе.

- **Номера вариантов – подряд, без пробелов и запятых.** Если в трёх вопросах ты выбрал варианты 4, 1 и 6, ответ – 416. Если в одном вопросе верных вариантов два, например 2 и 4, их обычно пишут по возрастанию: 24.
- **Дробную часть** на сайте обычно отделяют **точкой**: 0.25, а не 0,25. В книге мы пишем по-русски, через запятую.
- **Доля или проценты?** Шанс 25 % можно записать как 25 и как 0.25. Смотри, что просит условие.
- **Удары, ходы и копии – целые.** Если врагу нужно 3,3 удара, ударов будет 4: округляем вверх. Промежуточные результаты не округляй – только ответ.
- **У каждого могут быть свои числа.** В некоторых задачах условие генерируется для каждого участника отдельно, так что списать ответ не выйдет.
- **Попытки ограничены.** На задачу обычно даётся от 3 до 5 попыток, и каждая неверная снимает 30 % балла. Сначала проверь ответ в тетради или в редакторе, и только потом вводи.

Что понадобится

- Тетрадь в клетку, карандаш, ластик – для карт уровней, схем состояний и деревьев выборов. Ластик нужен всерьёз: схемы перерисовывают.
- Калькулятор. Хватит обычного на телефоне: проценты, дроби, умножение.
- Компьютер с интернетом. Планшета мало: Blender ставится на компьютер (Windows 10 или 11, или Linux; в программе сферы рекомендуют от 8 ГБ оперативной памяти).
- **Браузер** (например, Яндекс Браузер или Google Chrome) – для «Мастерской игр «Урсула»» (games.kruzhok.org). Там же курсы сферы, редактор игр и твои проекты.
- **Редактор диаграмм «Кибериада Graph Editor»** – в нём рисуют поведение персонажей на языке ПРИМС. Как его открыть, рассказано в курсах «Урсулы».
- **Blender** (blender.org) – бесплатная программа для 3D-моделирования, подойдёт версия 3.0 или новее.
- **Inky** – бесплатный редактор для языка Ink от его авторов (inklestudios.com/ink). Пишешь историю слева – и сразу играешь в неё справа.

Inky: редактор историй на Ink

Inky – бесплатный редактор языка Ink от его авторов, студии inkle. Слева пишешь историю, справа сразу в неё играешь. Интерфейс на английском, но слов в нём немного – ниже все нужные. Как устроен сам язык, рассказано в главе 7; здесь – как работать в программе и три задачи, чтобы руки привыкли.

Шаг 1. Скачать. Открой страницу github.com/inkle/inky/releases (ссылка есть и на inklestudios.com/ink) и в последней версии скачай **Inky_windows_64.zip** (около 225 МБ; для Mac – **Inky.dmg**). Устанавливать ничего не нужно: распакуй архив в любую папку и запусти **Inky.exe**. Если Windows спросит, можно ли запускать программу из интернета, – нажми «Подробнее» → «Выполнить в любом случае» вместе со взрослым.

Шаг 2. Окно Inky. При первом запуске в нём лежит маленький пример. Сотри его и пиши своё.

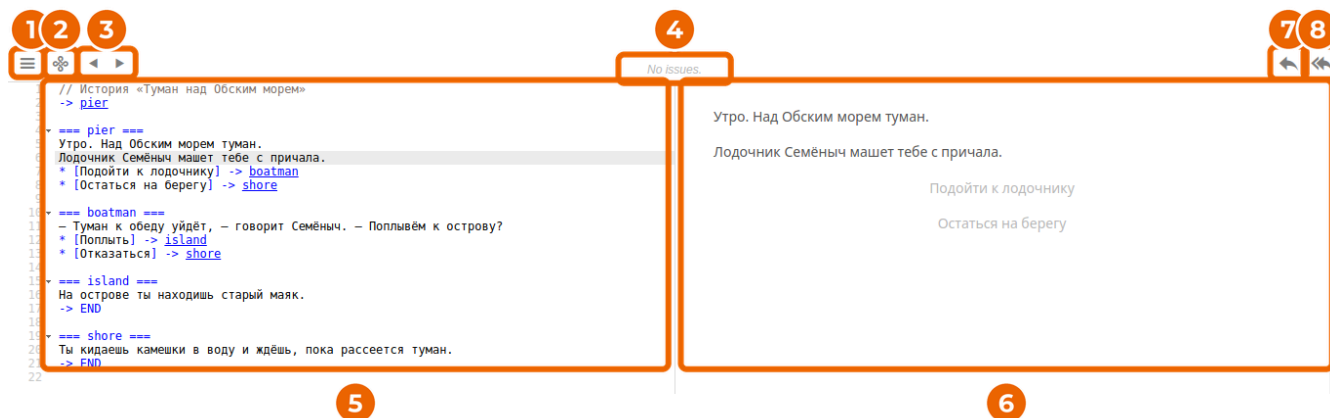


Рисунок В.1 – Окно Inky

1. **Toggle file browser** – список файлов проекта (нужен, когда история из нескольких файлов).
2. **Toggle knot browser** – список узлов истории: щёлкнул по имени – попал в узел.
3. **Navigate back / forward** – назад и вперёд по местам в коде, где ты был.
4. **Строка ошибок.** «No issues» – ошибок нет. Если есть – тут появится число и красный кружок.
5. **Редактор** – здесь пишут историю. Номера строк слева помогают найти ошибку.
6. **Игра** – история, как её увидит игрок. Обновляется сама, пока ты печатаешь.
7. **Rewind a single choice** – отменить последний выбор.
8. **Restart story** – начать историю сначала.

Шаг 3. Сохранить. «File → Save Project» (Ctrl + S). История сохранится файлом `.ink`. Открыть снова – «File → Open...». «File → Export for web...» соберёт из истории папку с веб-страницей: её можно открыть в браузере и дать поиграть друзьям.

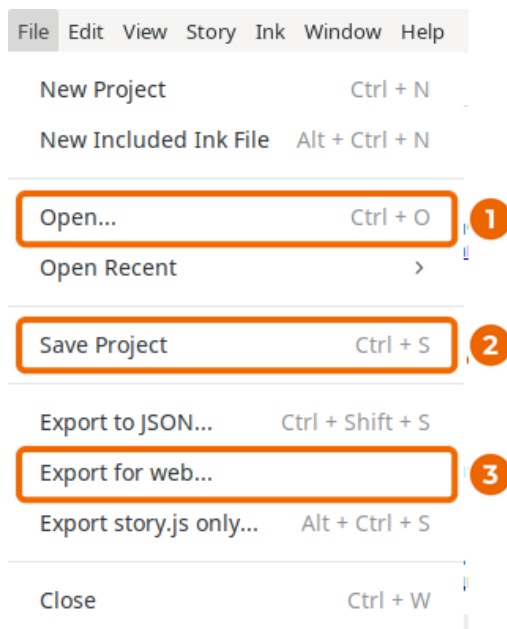


Рисунок В.2 – Меню File: открыть (1), сохранить (2), выгрузить в веб-страницу (3)

Задача П.1. Сыграй все пути. Набери в редакторе историю «Туман над Обским морем» из раздела 7.3. Справа нажми «Подойти к лодочнику» (1), потом «Поплыть». Начни заново кнопкой Restart story (8 на карте окна) и пройди остальные пути.

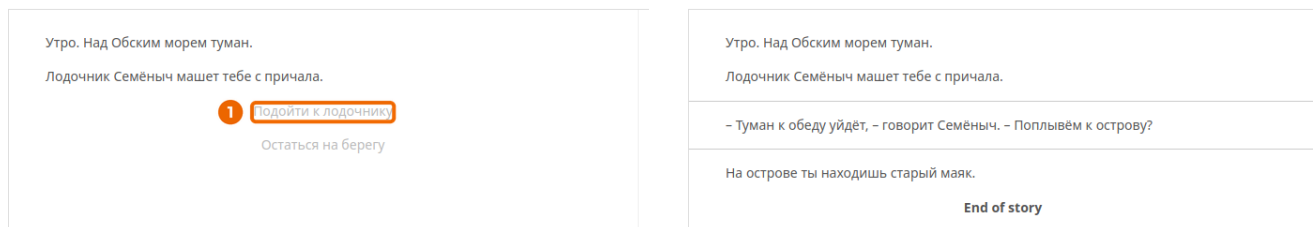


Рисунок В.3 – Слева – первый выбор, справа – концовка «остров»

Что должно получиться: три разных прохождения, две концовки («остров» и «берег»), в конце каждого Inky пишет «End of story». Сравни со схемой истории в разделе 7.3.

Задача П.2. Найди ошибку. В строке * [Отказаться] → shore сотри последнюю букву, чтобы получилось → shor . Посмотри, что делает Inky.

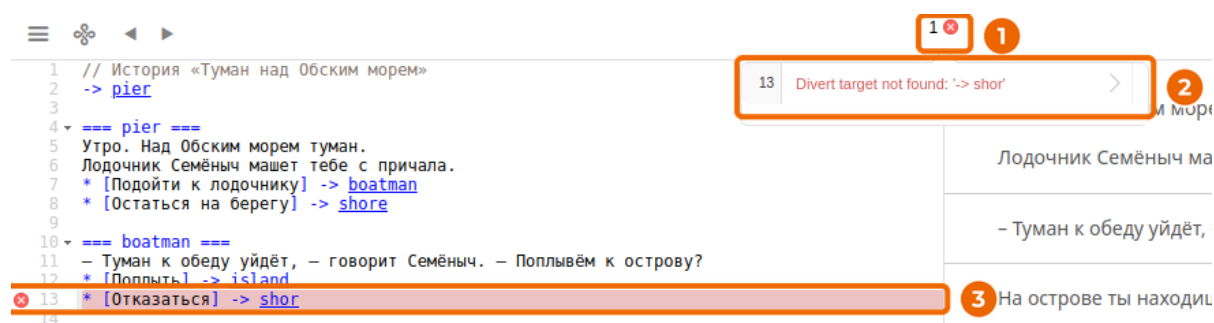


Рисунок В.4 – Ошибка: число ошибок (1), сообщение (2), строка с ошибкой (3)

Что должно получиться: вверху вместо «No issues» – «1» и красный кружок (1). Щёлкни по нему – появится список: номер строки 13 и сообщение «Divert target not found: '-> shor'» (2) – «узел для перехода не найден». Сама строка подсвечена красным (3). Верни букву – ошибка пропадёт. Так Inky ловит главную ошибку новичков: опечатку в имени узла.

Открой список узлов кнопкой Toggle knot browser (1). Там видно все четыре узла (2) – удобно проверять, что каждый узел, куда ведут стрелки, действительно есть.

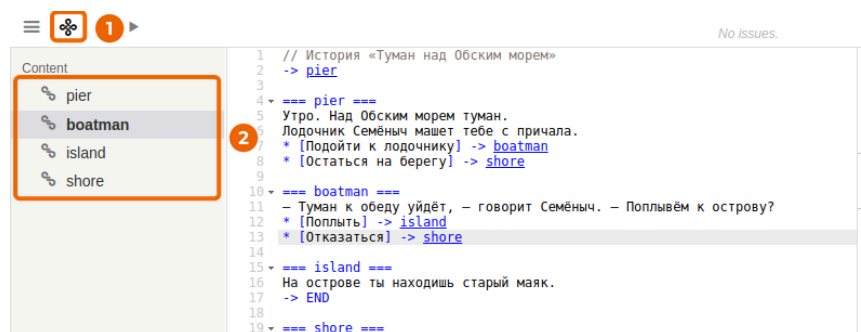


Рисунок В.5 – Список узлов истории

Задача П.3. История запоминает выбор. Сделай, чтобы Семёныч помнил, говорил ли ты с ним. Добавь в начало VAR talked = false , первой строкой узла boatman – ~ talked = true , а в узел shore перед → END – строку {talked: Семёныч машет тебе с лодки на прощание. |Лодка так и стоит у причала.} .

```

1  VAR talked = false
2  -> pier
3
4  ▾ === pier ===
5  Утро. Над Обским морем туман.
6  Лодочник Семёныч машет тебе с причала.
7  * [Подойти к лодочнику] -> boatman
8  * [Остаться на берегу] -> shore
9
10 ▾ === boatman ===
11 ~ talked = true
12 - Туман к обеду уйдёт, – говорит Семёныч. – Поплывём к острову?
13 * [Поплыть] -> island
14 * [Отказаться] -> shore
15
16 ▾ === island ===
17 На острове ты находишь старый маяк.
18 -> END
19
20 ▾ === shore ===
21 Ты кидаешь камешки в воду и ждёшь, пока рассеется туман.
22 {talked: Семёныч машет тебе с лодки на прощание. | Лодка так и стоит у причала.}
23 -> END
24 |

```

Рисунок В.6 – История с переменной talked

Пройди два пути, которые ведут на берег: сразу «Остаться на берегу» и через лодочника – «Отказаться».

Утро. Над Обским морем туман. Лодочник Семёныч машет тебе с причала. Ты кидаешь камешки в воду и ждёшь, пока рассеется туман. Лодка так и стоит у причала. End of story	Утро. Над Обским морем туман. Лодочник Семёныч машет тебе с причала. – Туман к обеду уйдёт, – говорит Семёныч. – Поплывём к острову? Ты кидаешь камешки в воду и ждёшь, пока рассеется туман. Семёныч машет тебе с лодки на прощание. End of story
---	---

Рисунок В.7 – Слева – сразу на берег, справа – через лодочника: концовки разные

Что должно получиться: в первом пути – «Лодка так и стоит у причала.», во втором – «Семёныч машет тебе с лодки на прощание.». Узел один и тот же, а текст разный – его решила переменная. Подробнее о переменных и условиях – раздел 7.4.

Blender: установка и первые шаги

Blender – бесплатная программа для 3D-моделей, в ней делают модели для игр «Урсулы». Ниже – как её поставить, где что в окне и четыре задачи, которые повторяют главу 8 руками: масштаб, массив, зеркало и экспорт в игру. Кадры сняты в Blender 5.2 LTS (октябрь 2026); в других версиях кнопки те же, могут немного съехать.

Шаг 1. Скачать и поставить. Открой blender.org/download и скачай версию для Windows: установщик `blender-...-windows-x64.msi` (ставится как обычная программа) или архив `.zip` (распаковал – и запускай `blender.exe`, установка не нужна; удобно в школьном классе). Видеокарта должна поддерживать OpenGL 4.3 – на очень старых компьютерах Blender может не запускаться.

Шаг 2. Первый запуск. Blender спросит настройки. Язык (1) оставь английским: в книге и на вебинарах сферы команды называют по-английски – Scale, Array, Mirror. Нажми «Continue» (2), а в следующем окне – «General»: откроется сцена с кубом, камерой и лампой.

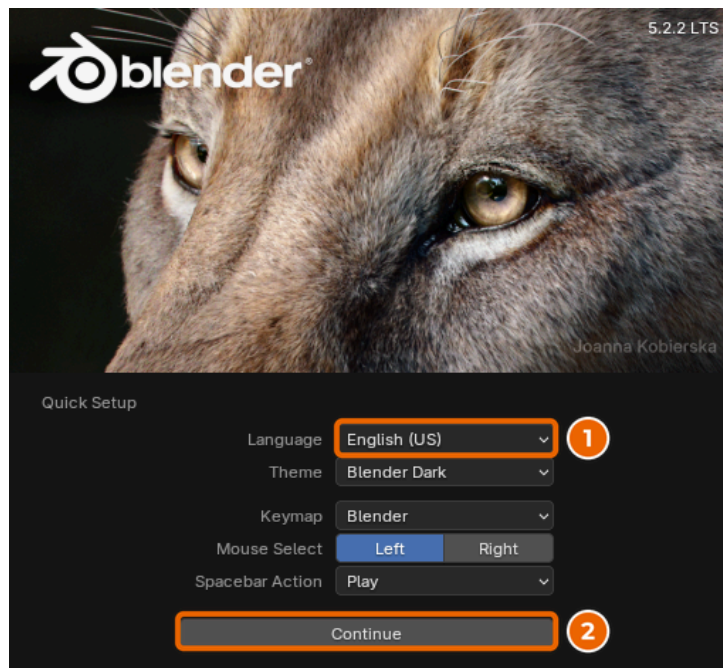


Рисунок В.8 – Первый запуск: язык (1) и «Continue» (2)

Шаг 3. Окно Blender.

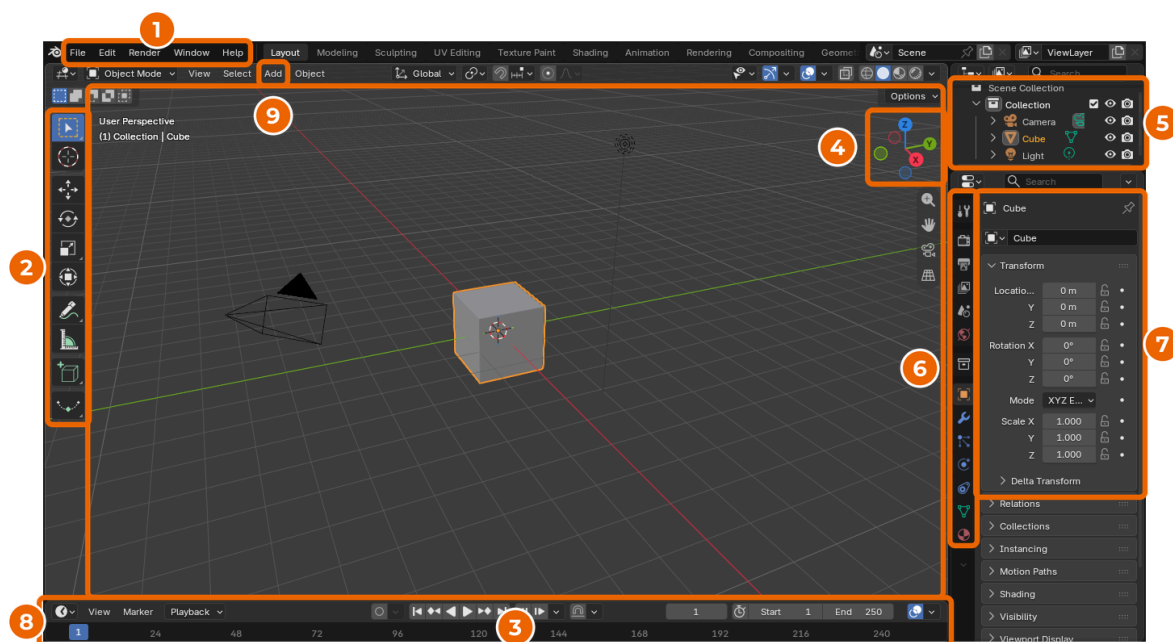


Рисунок В.9 – Окно Blender

1. **Меню** – File (сохранить, экспорт), Edit (отменить – Ctrl + Z) и другие.
2. **Инструменты** – выделить, двигать, вращать, масштабировать.
3. **3D-вид** – сцена. Колесо мыши – приблизить и отдалить; зажатое колесо – повернуть вид; Shift + зажатое колесо – сдвинуть вид; клавиша Home – показать всю сцену.
4. **Оси** – X красная, Y зелёная, Z синяя (вверх). Щёлкни по кружку оси – посмотришь на сцену с этой стороны.
5. **Список объектов сцены** (Outliner): Camera, Cube, Light. Щёлкнул по имени – выделил объект.

6. **Вкладки свойств.** Гаечный ключ – модификаторы, оранжевый квадрат – свойства объекта.
7. **Свойства** выделенного объекта на выбранной вкладке.
8. **Шкала времени** – для анимации, в задачах книги не нужна.
9. **Add** – добавить объект (то же, что Shift + A в 3D-виде).

Клавиши работают там, где стоит мышь: G, R, S, Tab и N жми, когда курсор над 3D-видом.

Задача П.1. Ящик высотой 0,5 м. Выдели куб, наведи мышь на 3D-вид и нажми N – справа откроется панель с числами: Scale (1) – во сколько раз растянут объект, Dimensions (2) – его настоящие размеры. У куба по умолчанию 2 м.

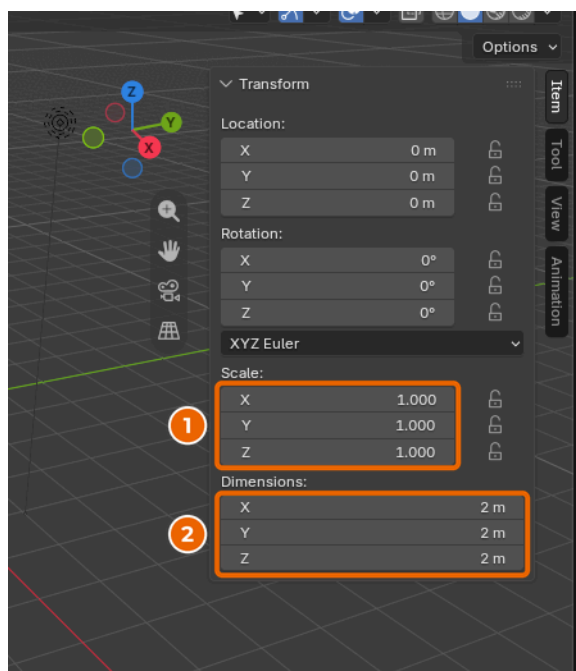


Рисунок В.10 – Панель N: масштаб (1) и размеры (2)

Дважды щёлкни по Scale Z (1), впиши 0.25 и нажми Enter. Посмотри на Dimensions Z (2).

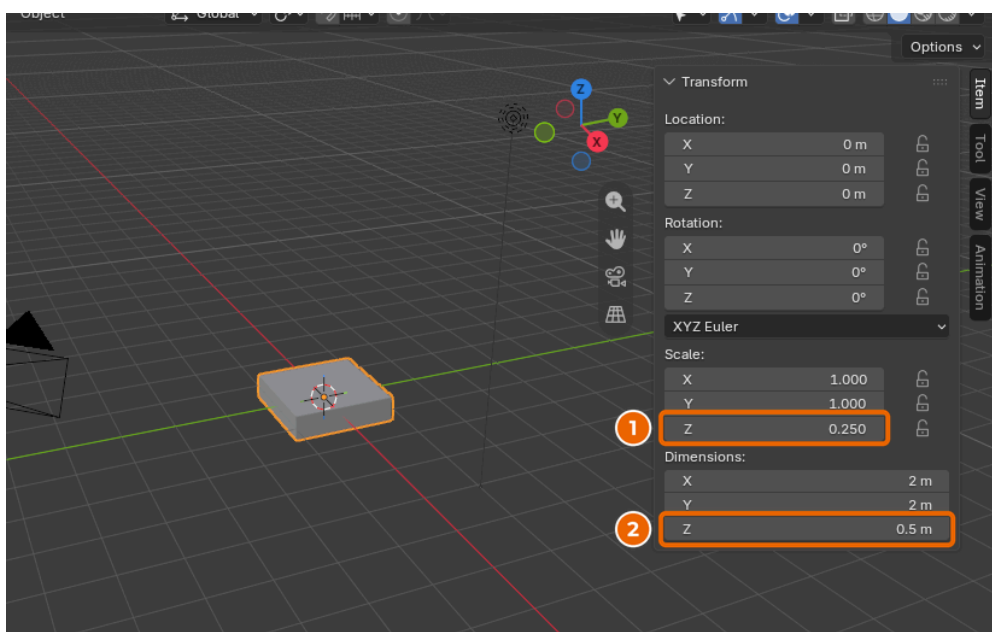


Рисунок В.11 – Scale Z = 0,25 – высота 0,5 м

Теперь примени масштаб: мышь над 3D-видом, Ctrl + A, в меню Apply – Scale (1).

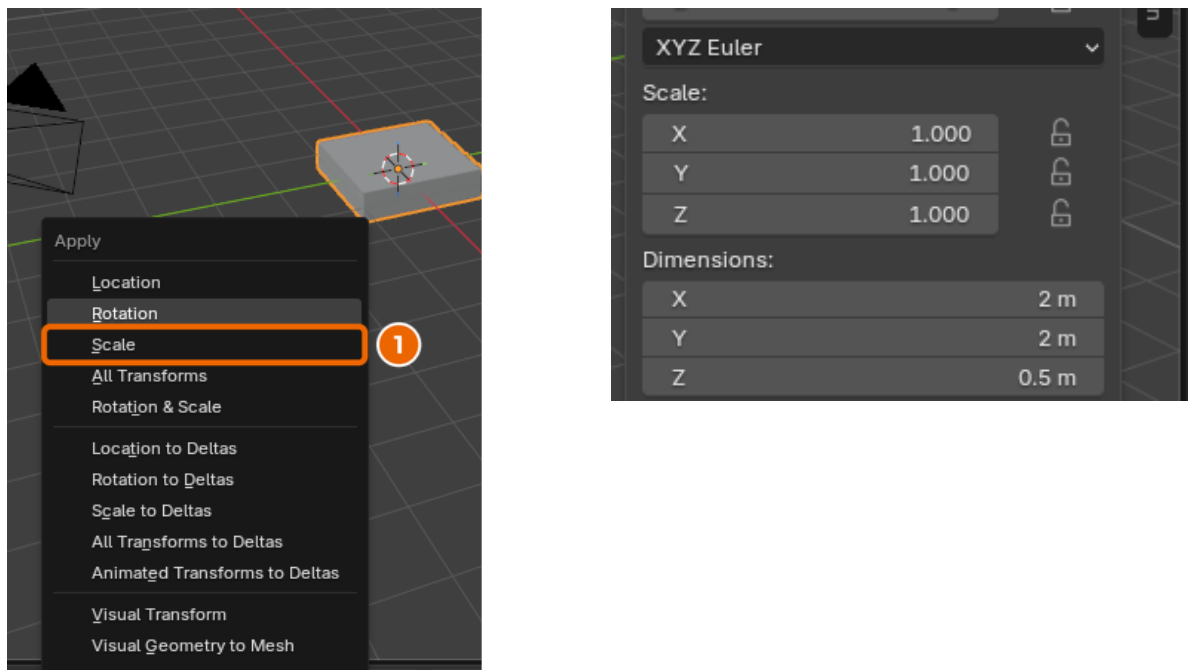


Рисунок В.12 – Слева – меню Apply, справа – числа после применения масштаба

Что должно получиться: после ввода 0.25 размер по Z – 0,5 м (сетка куба 2 м · 0,25). После Ctrl + A → Scale масштаб снова 1.000 по всем осям, а Dimensions Z остаётся 0,5 м. Так и готовят модель для игры (раздел 8.5).

Задача П.2. Забор из одной доски. Открой вкладку модификаторов (гаечный ключ, 6 на карте окна) и нажми «Add Modifier» (1).

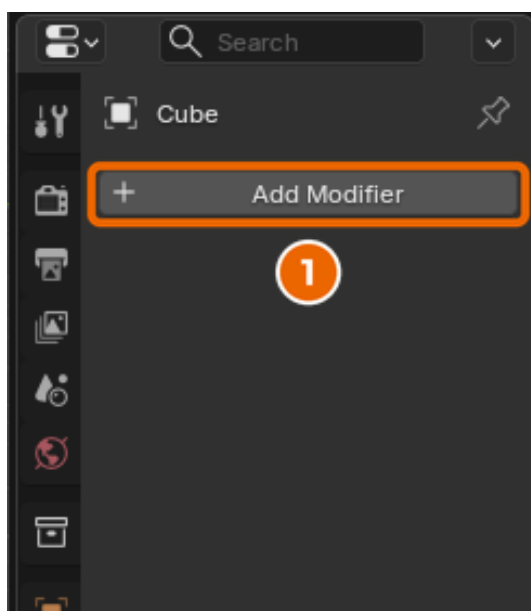


Рисунок В.13 – Вкладка модификаторов

В меню наведи на «Generate» (1) и выбери «Array» (2). В Blender 5 есть и «Array (Legacy)» – старый массив; у нового те же главные настройки.

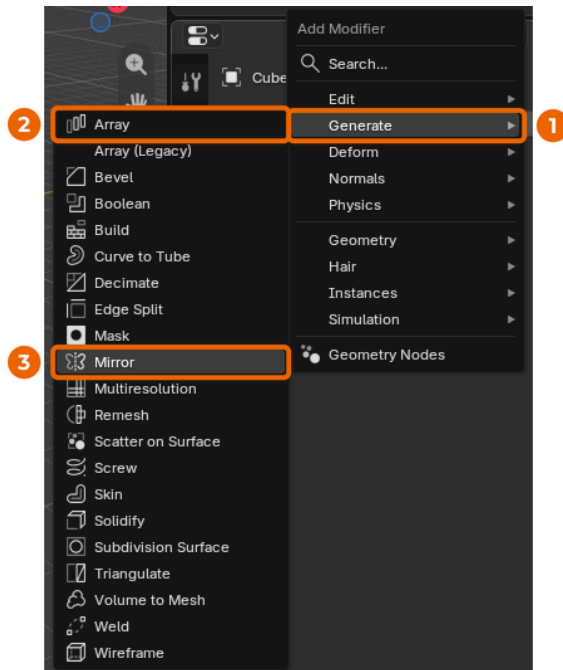


Рисунок В.14 – Меню Add Modifier → Generate: Array (2) и Mirror (3)

Поставь Count (1) – 6, Offset X (2) – 1.5. Чтобы посчитать вершины, включи статистику: стрелка рядом со значком наложений (Overlays) над 3D-видом (1), галочка Statistics (2).

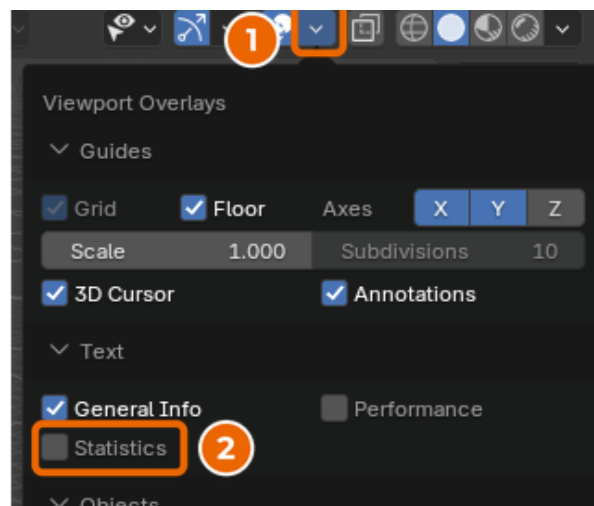


Рисунок В.15 – Включаем статистику

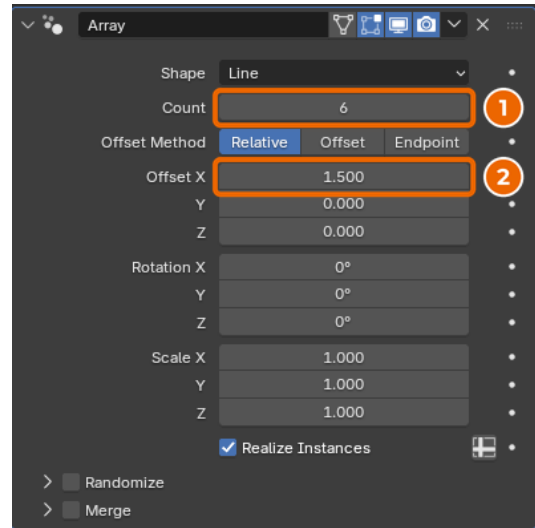
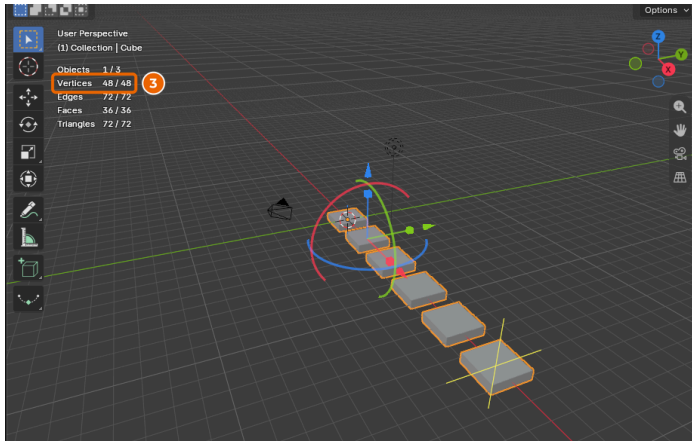


Рисунок В.16 – Массив из 6 досок: слева – 3D-вид со статистикой (3), справа – Count (1) и Offset X (2)

Что должно получиться: шесть досок в ряд с промежутками (сдвиг 1,5 ширины доски). В статистике Objects 1/3 – это всё ещё один объект, а Vertices 48 – у бруска 8 вершин, и Array умножил их на Count: $8 \cdot 6 = 48$ (раздел 8.4). Поставь Offset X = 1 – доски сомкнутся в одну длинную.

Задача П.3. Зеркало отражает от центра объекта. Удали Array (крестик в его заголовке). Наведи мышь на 3D-вид, нажми Tab (режим редактирования), A (выделить всё), потом по очереди G, X, 3 и Enter – сетка сдвинется на 3 м вправо, а центр объекта (оранжевая точка) останется на месте. Tab – обратно. Добавь модификатор Mirror (3 на рисунке с меню).

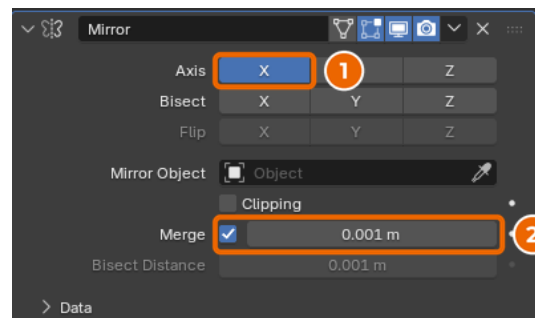
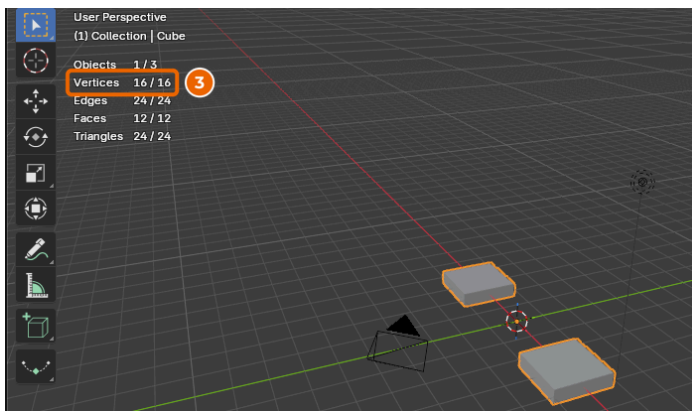


Рисунок В.17 – Mirror: копия появилась по другую сторону центра объекта; ось X (1), Merge (2), вершины (3)

Что должно получиться: копия бруска встала на 3 м **влево** от центра объекта – зеркало стоит в центре объекта, а не посередине бруска. Ось – X (1), галочка Merge (2) включена сама. Вершин 16 (3): по формуле $2V - k$ из раздела 8.3 $2 \cdot 8 - 0 = 16$ – на зеркале нет ни одной общей вершины, склеивать нечего.

Задача П.4. Экспорт в игру. File → Export (1) → glTF 2.0 (.glb/.gltf) (2).

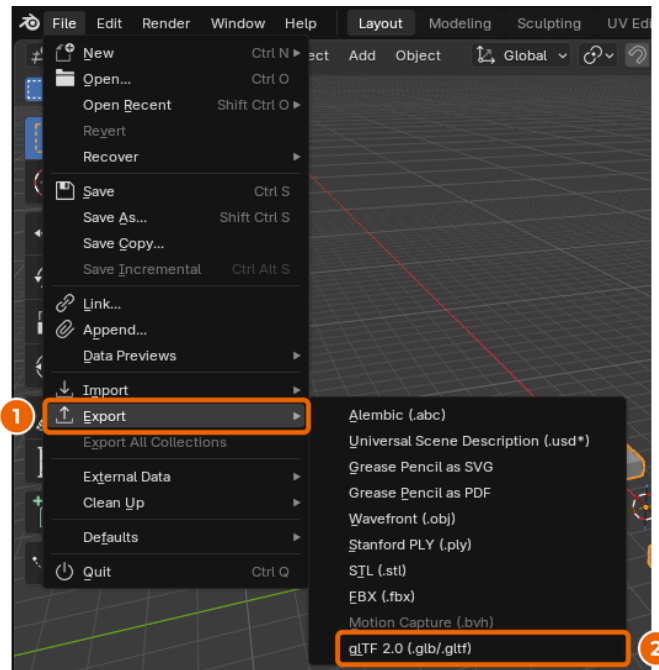


Рисунок В.18 – Меню экспорта

Справа в окне экспорта проверь: формат – glTF Binary (.glb) (1); в разделе Include – галочка Selected Objects (2), чтобы в файл не попали камера и лампа; в разделе Data → Mesh – галочка **Apply Modifiers** (3). Она по умолчанию **выключена**, без неё в файл уйдёт брусок без зеркальной копии. Нажми «Export glTF 2.0» (4).

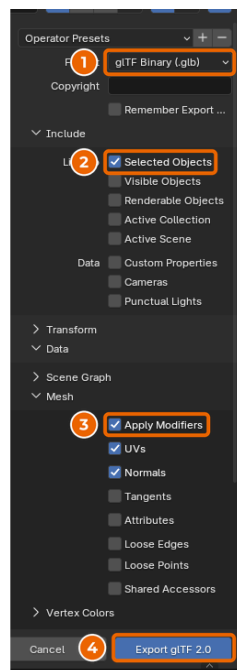


Рисунок В.19 – Настройки экспорта

Что должно получиться: файл `.glb` (несколько килобайт) с одним объектом. Мы проверили: с галочкой Apply Modifiers в файле обе половины – 24 треугольника, без неё – только 12. Именно такой файл загружают в редактор игры.

План на месяц до отбора

Если начинаешь в конце сентября, вот спокойный план: примерно по часу в день.

Таблица В.3

Неделя	Что читаем	Что делаем
1	главы 1–2	каждый день решать «Реши сам» по свежим разборам; войти на games.kruzhok.org и начать курс сферы
2	главы 3–4	нарисовать в клетку карту своего уровня; собрать похожую локацию в «Урсуле»
3	главы 5–6	повторить в «Кибериаде» машины состояний из разборов, загрузить в «Урсулу» и проверить в игре
4	главы 7–8 и шпаргалка	написать в Inku историю с двумя концовками; сделать в Blender простую модель по главе 8; заново решить задачи, где были ошибки; пройти тест курса сферы (+5 баллов)

Отбор длится до 10 ноября, и попыток две. Если первая попытка прошла неудачно, вернись к главам, где потерял баллы, и иди на вторую.

Родителям и наставникам

- **Помогайте разбираться, но не решайте за ребёнка.** На командном финале сразу видно, кто разобрался сам, а за кого решили. Лучше пусть ребёнок ошибётся на отборе и научится, чем попадёт в финал и растеряется там.
- **Играйте вместе – это тоже подготовка.** В списке того, что пригодится на отборе, первым стоит опыт игры в квесты. Пройдите вдвоём короткую игру-головоломку и спросите: где ты застрял? Как игра подсказала, что делать? Почему было интересно? Смотрите на возрастную маркировку. В книге мы старались упоминать только игры, которые подходят детям 11–14 лет.
- **Садитесь рядом на первых шагах в «Урсуле».** Детей чаще всего останавливает не сложность, а «проект не сохранился», «объект не программируется» или «непонятно, куда нажать». Разработчики сферы честно говорят, что освоиться в редакторе поначалу непросто.
- **Всё про регистрацию, бонусы и финал** для Новосибирской области – на nto.kvantorium54.ru (вкладки «Я наставник» и «Я участник или родитель»). Для наставников у сферы есть свой раздел с вебинарами и материалами для кружка: games.kruzhok.org/laboratory/mentor (вход через «Талант»).

- **Региональный координатор НТО Юниоры в Новосибирской области** – детский технопарк «Кванториум», info@kvantorium54.ru.

Честно об этой книге

Главное, что нужно знать про эту книгу: заданий отбора 2026/27 мы не видели – организаторы их заранее не показывают. Угадывать их бесполезно: наставники, которые следят за олимпиадой не первый год, говорят, что задания из года в год не повторяются. Поэтому книга учит темам из программы сферы и приёмам решения, а не готовит к конкретным задачам. Когда организаторы выложат пробные задания, мы сверим с ними книгу и выпустим новую версию.

Это не официальное издание НТО. Книгу составил методист «Кванториума» по открытым материалам олимпиады: программе сферы «Основы геймдизайна и разработки игр» 2026/27, записям вебинаров сферы для наставников (сентябрь 2026 года) и сборнику задач профиля «Разработка компьютерных игр» НТО для 8–11 классов (сезон 2025/26).

Задач отбора «Игротеха» 2026/27 года заранее никто не видел: их откроют только 20 октября. Разобранных задач прошлых лет по нынешним программам сферы («Урсула», ПРИМС, Ink) в открытом доступе мы тоже не нашли. Сборник 2022/23 года для прежней сферы «Технологии и компьютерные игры» был про Roblox и язык Lua – из него мы взяли только то, как выглядят сами задания. Задачи старшего трека сложнее, чем нужно тебе: мы брали из них идею, упрощали и пересчитывали. Поэтому эта книга учит понимать темы программы сферы, а не угадывать задания отбора.

«Урсула» и редактор диаграмм обновляются. Поэтому кнопки и окна мы описываем в общих чертах, а смысл – подробно: смысл от версии к версии не меняется. Точную инструкцию к нынешней версии смотри в курсах на games.kruzhok.org.

Каждое число в ответах мы пересчитали на компьютере, а каждую историю на Ink запустили. Если ты всё-таки нашёл ошибку или что-то осталось непонятным, напиши автору: osinovsky.aa@kvantorium54.ru. Следующие версии книги станут лучше благодаря тебе.

Игры и жанры: язык геймдизайнера



Прежде чем делать игру, надо договориться о словах. Что такое механика и чем она отличается от сюжета? Кто такой NPC? Почему «квест» – это и жанр, и задание? В этой главе мы соберём словарь геймдизайнера и научимся узнавать жанр игры по короткому описанию. На отборе такие задачи выглядят как тест: выбрать верные варианты, сопоставить слова и описания, найти лишнее в ряду. Ответ – строка из цифр, например «3142». Считать почти не нужно, а вот условие читай по слову: одна пропущенная частица «не» – и верный ход мысли даёт неверный ответ.

НА ПАЛЬЦАХ

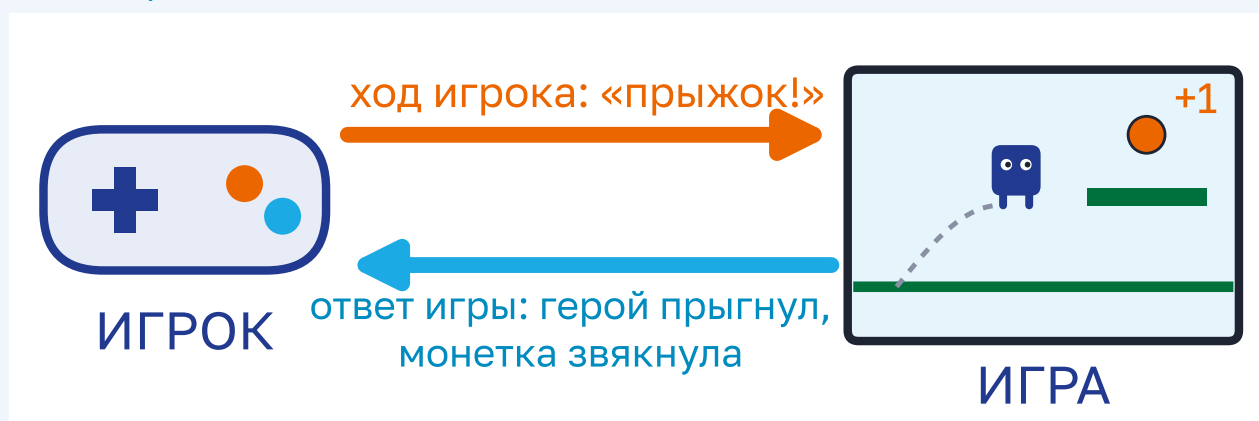


Рисунок 1.1 – Игра – это разговор: ты делаешь ход, игра отвечает

Мультфильм можно смотреть, сложив руки на коленях. Игру – нельзя: пока ты ничего не нажал, она ждёт. Нажал «прыжок» – герой прыгнул, монетка звякнула, счёт вырос на 1.

Запомни. Цель – чего хочешь. Правила – что можно и что за это будет. Препятствие – что мешает. Обратная связь – как игра показывает результат. Если в задаче просят «сопоставить часть игры с описанием», спроси себя про каждую фразу: «Она про то, чего добиться, как устроен мир, что мешает или как игра ответила?»

НА ПАЛЬЦАХ

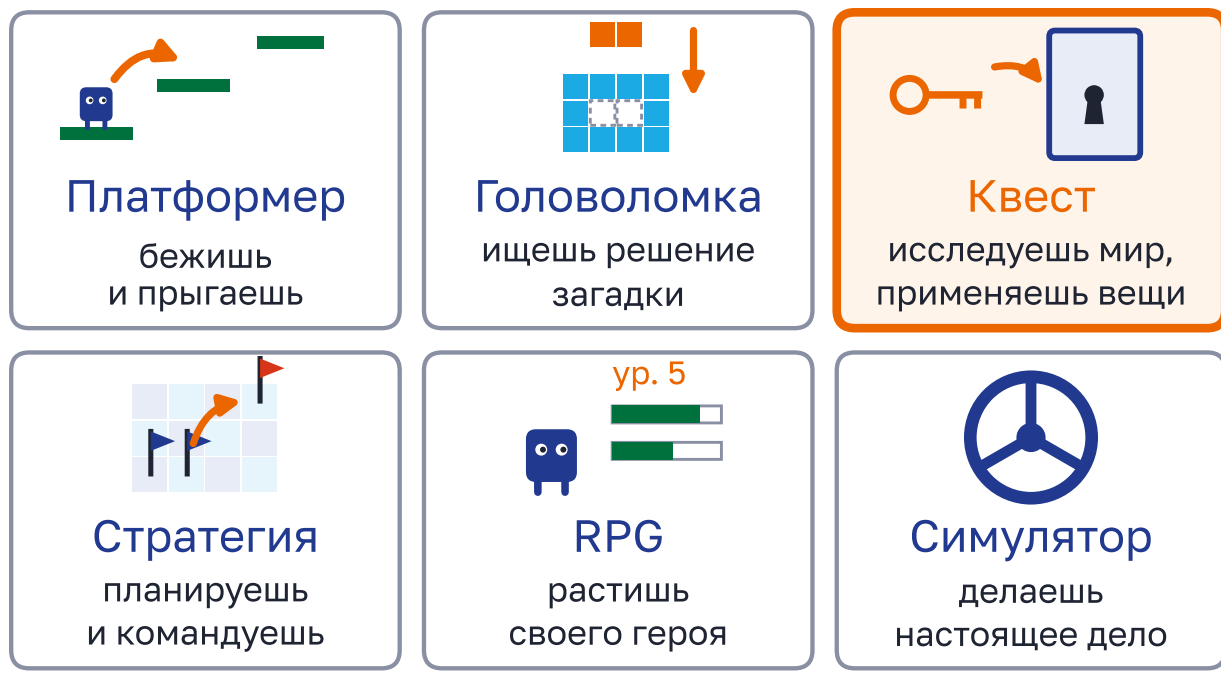


Рисунок 1.2 – Жанр – это то, что ты делаешь в игре большую часть времени

Названия жанров запоминать не нужно – нужно запомнить глаголы. Прыгаешь – платформер. Разгадываешь – головоломка. Исследуешь мир и применяешь найденное – квест.

Запомни. Жанр определяют по главному действию игрока. Тема и картинка не в счёт: игра про ферму бывает и симулятором, и стратегией. Если признаков двух жанров поровну, перед тобой смесь – и тогда в задаче обычно просят отметить все подходящие.

НА ПАЛЬЦАХ



Рисунок 1.3 – Обычный экран игры, а на нём – половина словаря геймдизайнера

Три сердечка и монеты – ресурсы. Зелёный персонаж – NPC, им управляет игра. Внизу – инвентарь, всё, что герой несёт с собой.

Запомни. Проверка на механику: разложи фразу на «событие → условие → изменение мира». Разложилась – механика. Не разложилась – перед тобой сеттинг, сюжет или атмосфера. А в инвентаре неполная стопка занимает целую ячейку, поэтому число ячеек – это деление с округлением вверх.

Запомни. Квест – и жанр, и задание. Шаги задания: выдача → цель → препятствие → условие → исход. Состояний у задания четыре: не взят, взят, выполнен, провален, – и в каждый момент только одно. Свободное дело можно вставить в ряд из трёх дел четырьмя способами, в ряд из четырёх – пятью.

Запомни. Прочитай, в каком виде нужен ответ, **до** того, как начнёшь решать. «Все верные» – по возрастанию. «Сопоставь» – по порядку букв. Лишний пробел или запятая могут сделать верный ответ неверным.

Разбор для старта

Задача 1.1 Дворник против сугробов

разминка · цель, правила, препятствия, обратная связь, сопоставление

Описание мобильной игры «Чистая дорожка» разбили на четыре фразы.

А. С крыши то и дело падают сосульки, а снегоуборочная машина сгребает снег обратно на дорожку.

- Б.** Нужно за 2 минуты расчистить дорожку от подъезда до школьного автобуса.
- В.** Каждая расчищенная клетка вспыхивает зелёным и звенит, а таймер в углу показывает, сколько секунд осталось.
- Г.** Одно нажатие лопатой убирает одну клетку снега, а сугроб выше колена – за два нажатия.

Сопоставь каждую фразу с частью игры: **1** – цель, **2** – правило, **3** – препятствие, **4** – обратная связь. В ответ запиши четыре цифры для фраз А, Б, В, Г по порядку.

Дано: четыре фразы (А–Г) и четыре части игры (1–4).

Найти: для каждой фразы – номер части игры.

Решение

Шаг 1. Вспомним вопросы-ключи. Цель – «чего добиться?». Правило – «как устроен мир, что будет за действие?». Препятствие – «что мешает?». Обратная связь – «как игра показала результат?».

Шаг 2. Разберём каждую фразу.

- **А.** Сосульки и машина, которая возвращает снег, не дают дворнику спокойно чистить – они **мешают**. Это препятствие, **3**.
- **Б.** «Нужно за 2 минуты расчистить дорожку» – это то, **чего надо добиться**. Цель, **1**.
- **В.** Вспышка, звон и таймер – игра **показывает**, что происходит. Обратная связь, **4**.
- **Г.** Сколько нажатий нужно на клетку – так **устроен мир**, так работает лопата. Правило, **2**.

Шаг 3. Собираем строку. А – 3, Б – 1, В – 4, Г – 2: получаем «3142».

Шаг 4. Проверим строку. Каждая цифра от 1 до 4 встретилась по одному разу: частей игры четыре, и фраз тоже четыре. Если какая-то цифра повторилась, значит, две фразы попали в одну «корзину» и одну из них надо перечитать.

 **Типичная ошибка.** Назвать фразу Г препятствием: «сугроб выше колена – это же трудно». Но фраза описывает, **как работает** лопата, а не что мешает. Вторая ошибка – записать ответ в порядке номеров частей («цель – Б, правило – Г...») и получить «БГАВ» или «2413». Порядок в ответе – по буквам фраз.

Ответ: 3142.

Своя задача по программе сферы, модуль 1

Проверь себя

1.1. Правила «Морского боя» разбили на четыре фразы. **А.** Соперник отвечает: «мимо», «ранил» или «убил». **Б.** Нужно первым потопить все корабли соперника. **В.** Ходите по очереди; выстрел – это название клетки, например «Б-5». **Г.** Корабли соперника спрятаны: его поле тебе не видно. Сопоставь фразы с частями игры: 1 – цель, 2 – правило, 3 – препятствие, 4 – обратная связь. Ответ – четыре цифры для А–Г.

1.2. Сопоставь термины и определения. Термины: **А** – NPC, **Б** – инвентарь, **В** – локация, **Г** – геймплей, **Д** – ресурс. Определения: 1) место в игровом мире: лес, пещера, двор замка; 2) то, что игрок копит и тратит и чего бывает мало; 3) персонаж, которым управляет не игрок, а игра; 4) что игрок делает в игре и как игра ему отвечает; 5) список вещей, которые герой несёт с собой. Ответ – пять цифр для А–Д.

1.3. Определи жанр по описанию: 1 – квест, 2 – платформер, 3 – головоломка, 4 – стратегия, 5 – RPG, 6 – симулятор. **А.** Ты водишь городской автобус: останавливаешься на остановках, открываешь двери, следишь за расписанием и скоростью. **Б.** Лиса бежит по лесу, прыгает по веткам и пенькам, переправляется через реку по кочкам. **В.** Расставь на поле зеркала так, чтобы луч попал в мишень; у каждого уровня одно решение, истории нет. **Г.** Девочка попадает в заброшенный музей, осматривает залы, говорит со зрителем, находит в вазе ключ и открывает им витрину. Ответ – четыре цифры для А–Г.

1.4.* В каждой игре отметить **все** жанры, признаки которых есть в описании: 1 – квест, 2 – платформер, 3 – головоломка, 4 – стратегия, 5 – RPG, 6 – симулятор. **Игра А.** Ты управляешь пекарней как настоящей: замешиваешь тесто, ставишь противни в печь, следишь за температурой. За каждый рабочий день пекарь получает опыт, растёт в уровне и учит новые рецепты. **Игра Б.** Робот бежит и прыгает по платформам. Иногда путь перегорожен, и надо переставить ящики так, чтобы открыть проход; у каждой такой загадки одно решение. **Игра В.** Сыщик в старой библиотеке осматривает полки, говорит с библиотекарем, собирает предметы и применяет их; драк и спешки нет. Запиши для каждой игры её цифры по возрастанию, игры – по порядку А, Б, В, всё подряд без пробелов.

Геймдизайн: механики, баланс, документ игры



Сколько здоровья у слизня? Сколько стоит новая кирка? Через сколько минут игрок накопит на замок? Каждое такое число кто-то придумал и проверил на калькуляторе. Это работа геймдизайнера. В главе ты научишься считать баланс, проверять документ игры и разбирать итоги плейтеста. На отборе такая задача – короткая история с правилами игры, а ответ – одно число или строка из номеров вариантов.



Рисунок 2.1 – Игровой цикл – колесо, которое крутит сам игрок

Стукнул киркой по камню – выпала руда. Продал руду – купил кирку покрепче. Новой киркой можно разбить камень потвёрже – и колесо пошло на новый круг.

Запомни. Цикл – кольцо: действие → награда → новая цель → снова действие. Выпало любое звено – кольцо рвётся, и игрок уходит. Короткие циклы (секунды) вложены в длинные (уровень, вся игра).

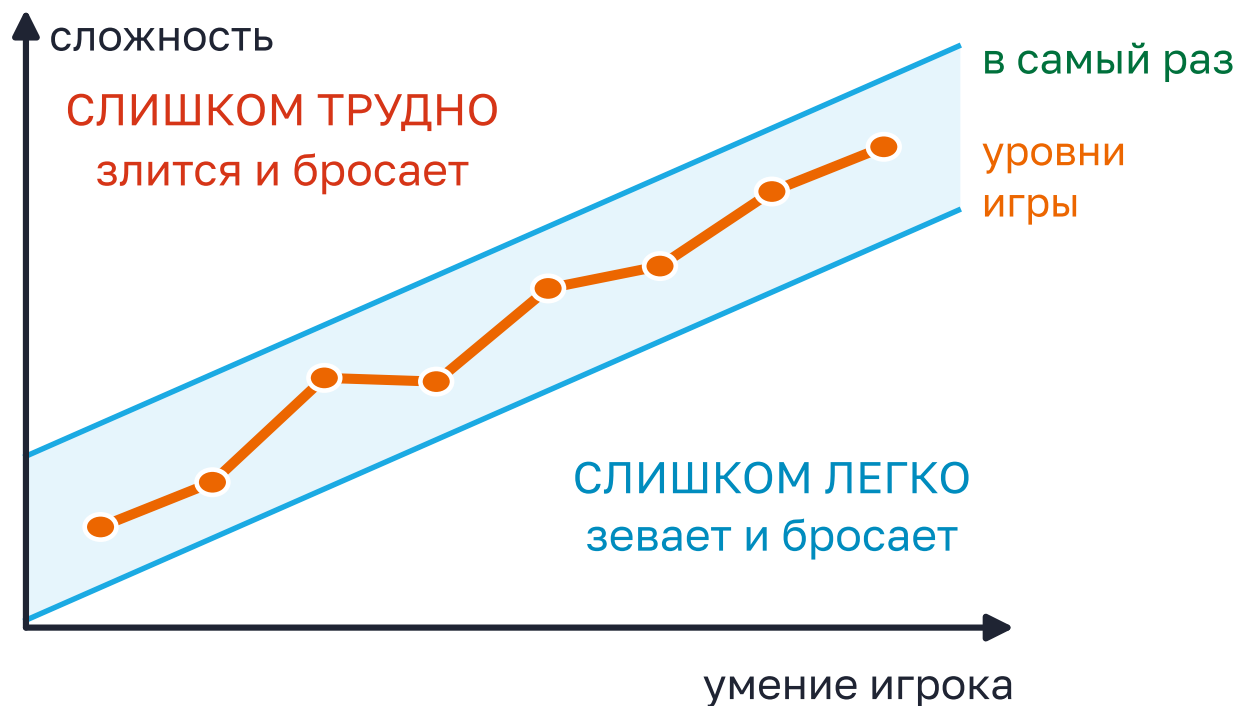


Рисунок 2.2 – Коридор интереса: сложность растёт вместе с умением игрока

Выше коридора – слишком трудно, игрок злится. Ниже – слишком легко, игрок зевает. Баланс держит уровни внутри коридора.

Запомни. Удары, ходы и порции обмена бывают только целыми – округляй вверх, чтобы хватило. Минус $p\%$ – умножить на $(1 - p/100)$, плюс $p\%$ – на $(1 + p/100)$. Проценты подряд перемножают, а не складывают.

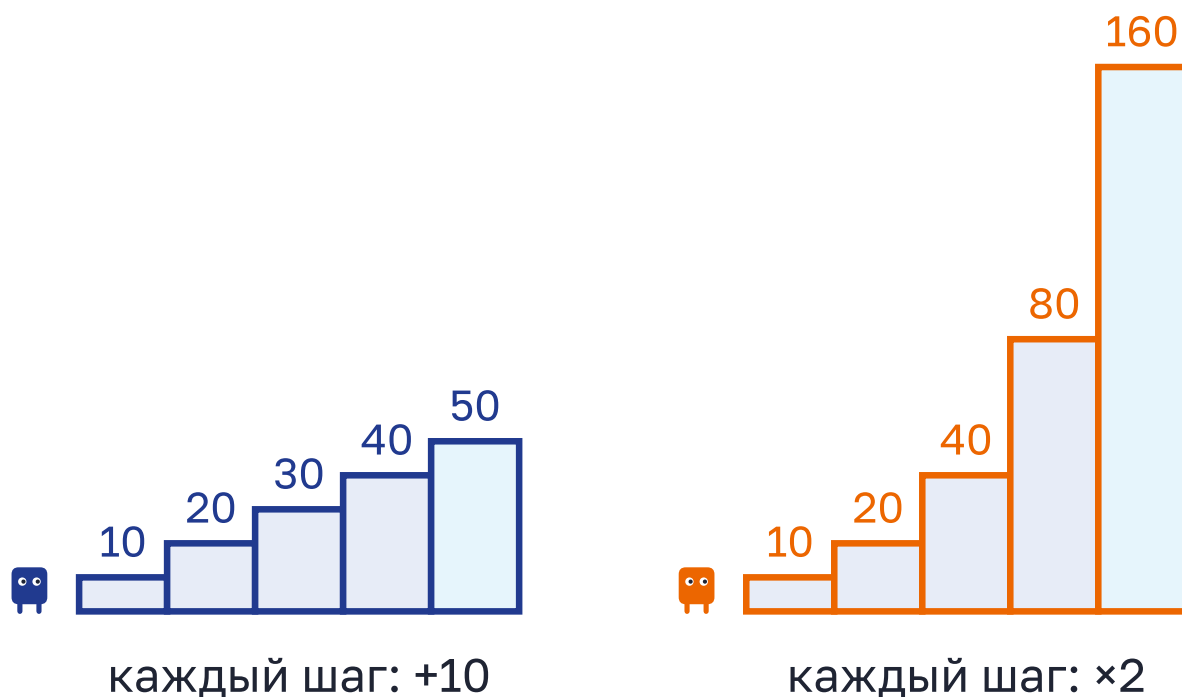


Рисунок 2.3 – Две лестницы: у левой ступеньки одинаковые, у правой – каждая выше предыдущей

Слева каждый шаг прибавляет одно и то же число. Справа каждый шаг умножает на одно и то же число – и лестница быстро уходит вверх.

Запомни. От первого числа ряда до n -го – $n - 1$ шаг, как и от 1-го уровня до n -го – $n - 1$ переход. Шаг «плюс»: первое $+$ $(n - 1) \cdot$ прибавка. Шаг «умножить»: первое $\cdot$ множитель в степени $n - 1$. Сумму надёжнее считать таблицей со строкой «всего».

Запомни. Время = недостающее : доход, округлить вверх. Если доход дан «за действие», сначала находим число действий (вверх), потом время. Улучшение окупается за «цена : прибавка» минут.

Запомни. Плейтест проводят на тех, кто игру ещё не видел. Смотри и записывай, а не объясняй. Долю ушедших считают от дошедших до этапа, а не от всех.

Разбор для старта

Задача 2.1 Каменный слизень

разминка · урон, броня в процентах, округление вверх

У каменного слизня 200 очков здоровья и броня, которая снижает любой урон на 25 %. Меч героя наносит 40 урона за удар. Сколько ударов мечом нужно, чтобы победить слизня? Ответ – целое число.

Дано: $HP = 200$; $d = 40$; броня $p = 25 \%$.

Найти: n – число ударов.

Решение

Шаг 1. Что происходит. Броня забирает часть каждого удара. Сначала узнаем, сколько урона доходит до слизня, потом – сколько таких ударов нужно.

Шаг 2. Урон после брони. Броня снимает 25 %, значит, до слизня доходит $100 - 25 = 75 \%$ удара: $40 \cdot 0,75 = 30$ урона.

Шаг 3. Число ударов. $200 : 30 = 6,67$.

Шаг 4. Округляем вверх. Проверим оба соседних целых числа. После 6 ударов: $6 \cdot 30 = 180$ – это меньше 200, у слизня осталось 20 здоровья, он жив. После 7 ударов: $7 \cdot 30 = 210$ – больше 200, слизень побеждён. Нужно 7 ударов.

Прикинь без брони. Хватило бы $200 : 40 = 5$ ударов. Броня ослабила каждый удар на четверть – ударов прибавилось, но не вдвое: было 5, стало 7.

 **Типичная ошибка.** Вычесть проценты из урона как очки: $40 - 25 = 15$, и тогда $200 : 15 = 13,3$ – «14 ударов». Броня дана в процентах, а не в очках. Ещё одна ловушка – округлять «по правилам». Здесь 6,67 по правилам тоже даёт 7, но если бы вышло 6,2, округление по правилам дало бы 6, и слизень остался бы жив. Удары округляем только вверх.

Ответ: 7.

Своя задача по программе сферы, модуль 1

Проверь себя

2.1. У робота-охранника 250 очков здоровья и щит, который снижает любой урон на 40 %. Лазер героя наносит 55 урона за выстрел. Сколько выстрелов нужно, чтобы победить робота? Ответ – целое число.

2.2. Удар героя – 80 урона. Зелье силы увеличило урон на 25 %, а потом проклятие уменьшило получившийся урон на 25 %. Каким стал урон? На сколько процентов он меньше исходных 80?

2.3. Сопоставь играм их игровые циклы. Один цикл лишний. В ответ запиши номера циклов по порядку игр А, Б, В, Г – четыре цифры подряд.

Игры: **А** – Pokémon; **Б** – Super Mario Bros.; **В** – Just Dance; **Г** – Machinarium.

1. Повторять движения танцора на экране в такт музыке → получить очки за точность → выбрать новую песню.
2. Осмотреть место → найти и подобрать предметы → решить с их помощью головоломку → попасть в новое место.
3. Собрать команду футболистов → сыграть матч → получить деньги → купить игроков посильнее.
4. Встретить дикое существо → сразиться с ним → твоя команда получает опыт и становится сильнее → идти к соперникам посильнее.
5. Бежать и прыгать через ямы и врагов, собирая монеты → дойти до флагштока в конце уровня → открыть следующий уровень.

2.4. Геймдизайнер набросал циклы для четырёх игр. Про каждый набросок реши: 1 – цикл целый; 2 – за действие игрок ничего не получает; 3 – награду некуда потратить. В ответ запиши четыре цифры по порядку набросков А, Б, В, Г.

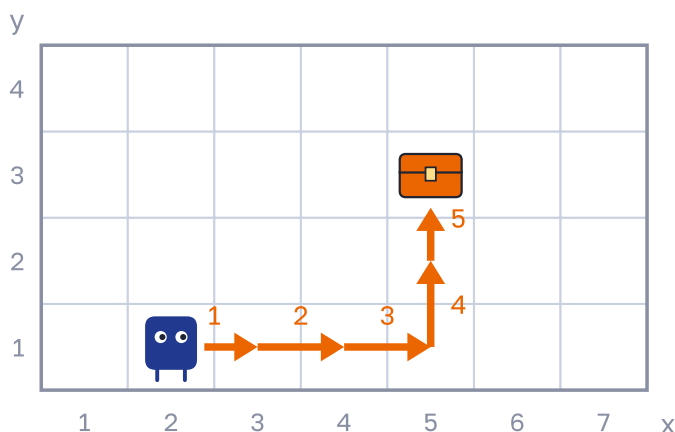
- **А.** Игрок ловит рыбу и продаёт её в лавке, а на монеты покупает удочку, которой можно поймать рыбу крупнее.
- **Б.** Игрок собирает на уровне звёзды. Счётчик звёзд растёт, но звёзды ничего не открывают, и купить за них нечего.
- **В.** Игрок поливает цветы в саду. Цветы от этого не растут, монет и очков за полив нет. А в лавке продаются новые лейки за монеты.
- **Г.** Игрок побеждает слизней и получает опыт. С каждым новым уровнем героя открывается пещера, где слизи сильнее.

Левел-дизайн: карты и пути



Левел-дизайнер – это человек, который строит уровень: где стены, где проходы, где лежит ключ и откуда игрок увидит выход. Прежде чем собрать локацию в редакторе, её рисуют на бумаге в клетку. Здесь ты научишься читать такую карту: находить клетку по адресу, вести по ней робота, искать самую короткую дорогу и считать, сколько всего дорог ведёт к замку. В списке «что пригодится на отборе» это пункты про алгоритмы, исполнителей и комбинаторику. Ответ в таких задачах – обычно одно число: шаги, метры, число путей или номер верной программы.

НА ПАЛЬЦАХ



Сундук – в клетке (5; 3)

5-й столбец слева, 3-я строка снизу

3 шага вправо + 2 шага вверх

= 5 шагов по клеткам

Сначала столбец (x), потом строка (y)

Рисунок 3.1 – Адрес клетки – два числа: столбец и строка

Чтобы сказать другу, где сундук, не нужно рисовать – достаточно двух чисел.
Первое – столбец, второе – строка.

Запомни. Координаты (x; y): сначала столбец, потом строка. Без диагоналей шагов столько, сколько разница по x плюс разница по y. С диагоналями – сколько большая из двух разниц. Клетки в метры – умножением на размер клетки.

НА ПАЛЬЦАХ

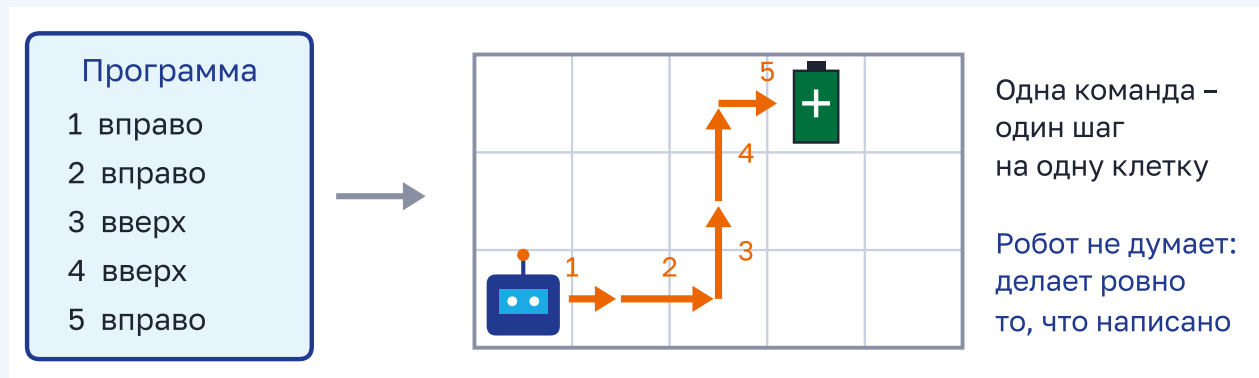


Рисунок 3.2 – Программа – это список команд. Робот выполняет их по порядку, ровно одну за другой

Робот не видит батарейку и не знает, куда ему надо. Он просто делает шаг за шагом то, что написано в карточке.

Запомни. Каждая команда движения меняет одну координату на единицу. Чтобы узнать, где окажется робот, проще всего сложить сдвиги: сколько всего раз вправо минус сколько раз влево – это изменение x; вверх минус вниз – изменение y. А вот врежется ли он в стену, по сумме не узнать: тут придётся пройти путь клетка за клеткой.

НА ПАЛЬЦАХ

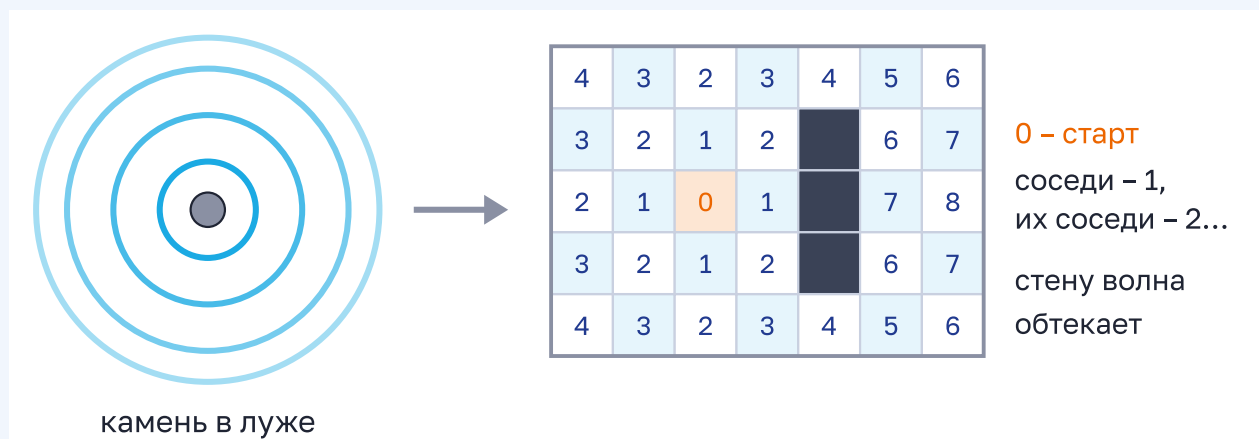


Рисунок 3.3 – Брось камень в лужу – круги расходятся во все стороны. Так же расходятся по клеткам числа

Число в клетке – сколько шагов до неё от старта. Стена волну не пропускает, и она обходит её сбоку.

Запомни. Волна: старт – 0, соседи – 1, их соседи – 2... Число в финише – длина кратчайшего пути. Путь – обратным ходом, по числам на 1 меньше. Клетку, где число уже стоит, не переписываем.

Запомни. Граф: комната – вершина, проход – ребро, над ребром длина. Кратчайший путь – метками: закрепляй комнату с самым маленьким числом и обновляй соседей. Ключи – по кругам: куда можно дойти → какие ключи там → какие двери они открывают.

Запомни. Только вправо и вверх: клетка = слева + снизу. На нижнем и левом краю – по 1, но только до первой запрещённой клетки. Запрещённая клетка – 0. Если выбор идёт по шагам, варианты шагов перемножают: так считают путь через обязательную клетку.

Запомни. Игрок идёт туда, где свет, где виден ориентир и куда ведут «хлебные крошки». Одинаковые коридоры, темнота везде и подсказки текстом каждые десять секунд – признаки слабого уровня.

Разбор для старта

Задача 3.1 Тропинка к колодцу

разминка · координаты, шаги по клеткам, масштаб

На плане деревни каждая клетка – 2 м на 2 м. Герой стоит в клетке (2; 3), колодец – в клетке (9; 7). Ходить можно только по сторонам клеток, без диагоналей, стен и заборов на пути нет. Сколько метров пройдёт герой до колодца по кратчайшему пути? Ответ – целое число.

Дано: герой (2; 3); колодец (9; 7); клетка – 2 м; ходы только по сторонам.

Найти: длину пути в метрах.

Решение

Шаг 1. Разница по столбцам. x меняется от 2 до 9: $9 - 2 = 7$ шагов вправо.

Шаг 2. Разница по строкам. y меняется от 3 до 7: $7 - 3 = 4$ шага вверх.

Шаг 3. Всего шагов. Без диагоналей шаги складываются: $7 + 4 = 11$ шагов. В каком порядке их делать – неважно, стен нет.

Шаг 4. Шаги в метры. Один шаг – одна клетка, то есть 2 м: $11 \cdot 2 = 22$ м.

22 м – это много? Колодец на 7 клеток правее и на 4 выше – это 14 м вбок и 8 м вперёд. Без диагоналей путь равен сумме этих отрезков, 22 м. Примерно столько от подъезда до детской площадки.

 **Типичная ошибка.** Считать клетки, а не шаги, и учесть стартовую: «клетки со 2-й по 9-ю – это 8 клеток». Так получится $8 + 5 = 13$ клеток и 26 м. Но в стартовой клетке герой уже стоит, её проходить не надо. Проверка на пальцах: из клетки 2 в клетку 3 – это один шаг, а не два.

Ответ: 22 м.

Своя задача по программе сферы, модуль 1

Проверь себя

3.1. На карте лагеря одна клетка – 1,5 м. Вожатый стоит в клетке (3; 8), флагшток – в клетке (10; 1). Ходить можно только по сторонам клеток, препятствий нет. Сколько метров до флагштока по кратчайшему пути? Ответ – число.

3.2. Рыцарь стоит в клетке (2; 2), портал – в клетке (9; 5). Каждый шаг на соседнюю клетку занимает 0,5 с. Сначала рыцарь умел ходить только по сторонам клеток, а после улучшения научился ходить и по диагонали (диагональный шаг – тоже 0,5 с). На сколько секунд быстрее он теперь доберётся до портала? Препятствий нет. Ответ – десятичная дробь.

3.3. На поле нет стен. Робот стоит в клетке (2; 3) и выполняет программу:

```
вправо  
повтори 4 [вверх, вправо, вправо]  
вниз  
вниз
```

В какой клетке он окажется? Ответ запиши двумя числами: сначала x , потом y .

3.4. Карта: # – стена, . – свободная клетка, Р – робот, Б – батарейка.

```
#####  
#...#...#  
#.#.#.#  
#.#...#.#  
#Р..#..Б#  
#####
```

Программист написал для робота программу:

```
повтори 2 [вправо]  
вверх  
повтори 2 [вправо]  
повтори 2 [вверх]  
повтори 2 [вправо, вниз]  
вниз
```

Робот врезался в стену. На какой по счёту команде это случилось, если считать каждую команду внутри цикла отдельно? Ответ – номер команды.

Логика, выборы и случайность



Эта глава – про математику, которая прячется внутри квеста. Подсказки вида «кузнец живёт не у колодца» распутывают таблицей с крестиками. Лжеца ловят на противоречии. Дальше – условия с И, ИЛИ, НЕ, подсчёт концовок в ветвистой истории и шанс, что из босса выпадет редкий питомец. На отборе такая задача – короткая история, а ответ – число, доля или строка цифр вроде «213». Формул почти нет, зато много аккуратного перебора: пропустил один случай – ответ неверный.

НА ПАЛЬЦАХ

1. Кузнец живёт не у колодца и не у мельницы.
2. Травница живёт не у колодца.

	колодец	мельница	мост
Кузнец	×	×	●
Травница	×	●	×
Стражник	●	×	×

одна пустая клетка – кузнец живёт здесь

× крестик из подсказки
× дом уже занят

Рисунок 4.1 – Три жителя, три дома и две подсказки

Каждая подсказка с «не» – это крестик. Когда в строке осталась одна пустая клетка, житель живёт там.

Запомни. Строки – кто, столбцы – где. Подсказка с «не» – крестик. Одна свободная клетка в строке – точка. Точка закрывает свой столбец для всех остальных.

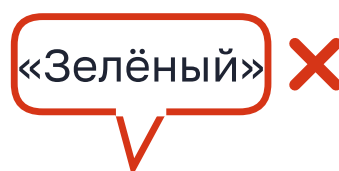
НА ПАЛЬЦАХ

«Какого цвета снег?»



РЫЦАРЬ

всегда говорит правду



ЛЖЕЦ

всегда лжёт

Рисунок 4.2 – Рыцарь и лжец отвечают на один и тот же вопрос

Рыцарь говорит правду всегда. Лжец лжёт всегда – даже про цвет снега.

Запомни. Предположи – проверь – ищи противоречие. Противоречие значит, что предположение неверно. «А говорит, что Б лжец» – они разные. «А говорит, что Б рыцарь» – они одинаковые.

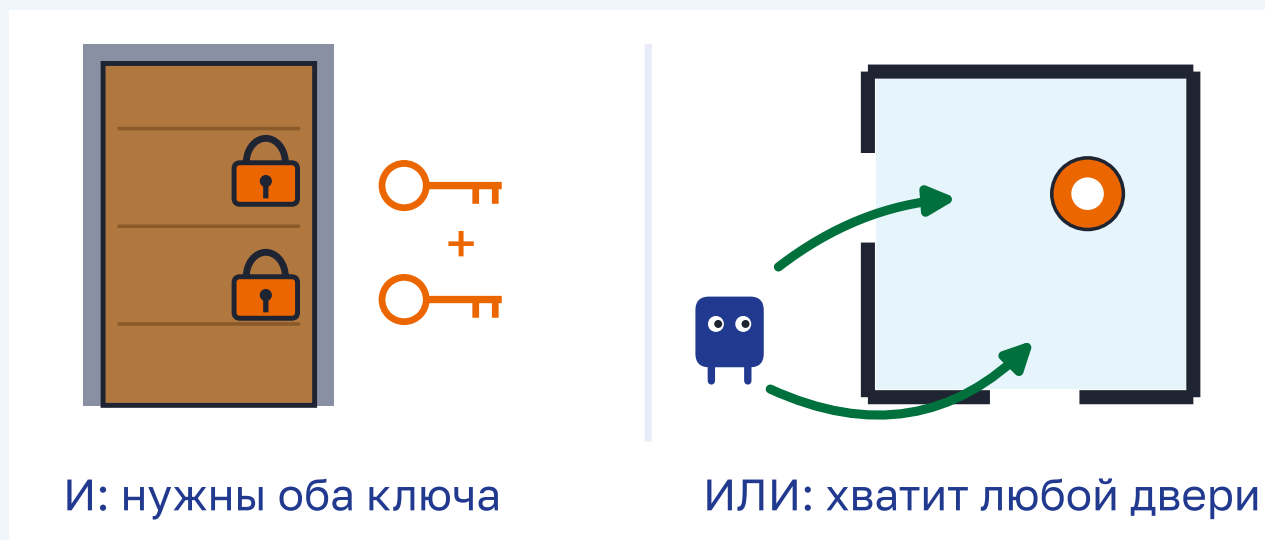


Рисунок 4.3 – Дверь с двумя замками и комната с двумя дверями

Два замка на одной двери – нужны оба ключа: это И. Две двери в одну комнату – хватит любой: это ИЛИ.

Запомни. И – нужны обе части. ИЛИ – хватит одной, обе тоже годятся. НЕ – наоборот. Порядок: скобки, НЕ, И, ИЛИ.

Запомни. Выборы один за другим – умножай. Разные ветки – складывай. k развилок по 2 варианта без схождений – 2^k концовок. Ветки сходятся – рисуй карту и пиши у каждого места сумму чисел, из которых в него пришли.

Запомни. Шанс = подходящие исходы : все исходы. $20\% = 0,2 = 1/5 = \text{«1 из 5»}$ – в среднем, а не каждый пятый раз. Все исходы вместе – 100% . Независимые события сразу – шансы умножаются. На двух кубиках 36 исходов, чаще всего выпадает сумма 7.

Запомни. Прошлые неудачи не меняют шанс следующей попытки, если в правилах нет счётчика. «Хотя бы раз за n попыток» считают через «ни разу»: $1 - (1 - p)^n$. Даже много попыток не дают 100% .

Разбор для старта

Задача 4.1 Потерянные вещи

разминка · логическая задача, таблица исключений, ответ строкой цифр

В подземном квесте три гнома – Бур, Гром и Тиль – потеряли по одной вещи: фонарь, кирку и карту. Герой должен вернуть каждому его вещь. Подсказки:

1. Бур не терял ни фонарь, ни карту.
2. Тиль не терял фонарь.

Вещи пронумерованы: 1 – фонарь, 2 – кирка, 3 – карта. Запиши номера вещей Бура, Грома и Тиля – по порядку, без пробелов.

Дано: три гнома, три вещи, у каждого ровно одна; две подсказки.

Найти: номера вещей Бура, Грома, Тиля.

Решение

Шаг 1. Рисуем таблицу. Строки – гномы, столбцы – вещи. По подсказке 1 ставим крестики «Бур – фонарь» и «Бур – карта». По подсказке 2 – крестик «Тиль – фонарь».

Шаг 2. Ищем строку с одной пустой клеткой. У Бура пустая только кирка – ставим точку. Кирка занята, зачёркиваем её у Грома и у Тиля.

Шаг 3. Дальше по цепочке. У Тиля закрыты фонарь (подсказка) и кирка (занята) – остаётся карта. Грому остаётся фонарь.

Таблица 4.1

	Фонарь	Кирка	Карта
Бур	х	•	х
Гром	•	х	х
Тиль	х	х	•

Шаг 4. Записываем ответ по порядку вопроса. Бур – кирка (2), Гром – фонарь (1), Тиль – карта (3).

Сверим с подсказками. Бур не терял ни фонарь, ни карту – и правда, у него кирка. Тиль не терял фонарь – у него карта. В каждой строке и в каждом столбце по одной точке.

 **Типичная ошибка.** Записать ответ в порядке вещей, а не гномов: «кому фонарь, кому кирка, кому карта». Получится совсем другая строка. И ещё легко забыть, что точка закрывает весь столбец. Тогда у Тиля останутся две пустые клетки, кирка и карта, и кажется, что решений два.

Ответ: 213.

Своя задача по программе сферы, модуль 1

Проверь себя

4.1. На школьной ярмарке игр Лёва, Миша и Соня показывали каждый свою игру: платформер, квест и гонки. Лёва не делал гонки. Соня не делала ни квест, ни гонки. Жанры пронумерованы: 1 – платформер, 2 – квест, 3 – гонки. Запиши номера жанров Лёвы, Миши и Сони по порядку.

4.2. В деревне квеста на одной улице стоят четыре дома, пронумерованные слева направо от 1 до 4. В них живут мельник, пекарь, рыбак и кузнец, в каждом доме один житель. Известно:

1. Пекарь живёт рядом с кузнецом – в соседнем доме.
2. Мельник живёт левее пекаря.
3. Мельник живёт не в крайнем доме.
4. Пекарь живёт не в доме 4.

Запиши номера домов мельника, пекаря, рыбака и кузнеца по порядку.

4.3. У двух дверей стоят стражники Том и Рик, каждый – рыцарь или лжец. Том сказал: «Хотя бы один из нас – лжец». Рик промолчал. Запиши для Тома и Рика по порядку: 1 – рыцарь, 0 – лжец.

4.4. Три жителя острова, А, Б и В, – каждый рыцарь или лжец. А сказал: «Среди нас ровно один рыцарь». Б сказал: «Среди нас ровно два рыцаря». В сказал: «Среди нас нет ни одного рыцаря». Запиши для А, Б и В по порядку: 1 – рыцарь, 0 – лжец.

Машины состояний: как думает NPC



Стражник в игре замечает тебя, бежит следом, бьёт, а потеряв из виду, возвращается к обходу. Кажется, что он думает. А у него всего лишь несколько режимов и правила, по которым он прыгает из одного в другой. Такая схема называется машиной состояний. В «Урсуле» персонажей программируют так же, на языке ПРИМС. Задачи по таким схемам похожи одна на другую: дают схему и список событий, а спрашивают, где окажется персонаж, сколько раз что-то сработает или какая стрелка лишняя. Ответ – номер состояния, число или строка цифр.

НА ПАЛЬЦАХ

● сейчас ночник здесь – и только здесь



Рисунок 5.1 – Ночник с одной кнопкой

У ночника два режима, и кнопка переводит его из одного в другой. Быть сразу и включённым, и выключенным он не может.

Запомни. Машина состояний – это прямоугольники-состояния, стрелки-переходы и события на стрелках. Объект всегда находится только в одном состоянии. Стрелка от чёрной точки показывает, с чего всё начинается.

НА ПАЛЬЦАХ

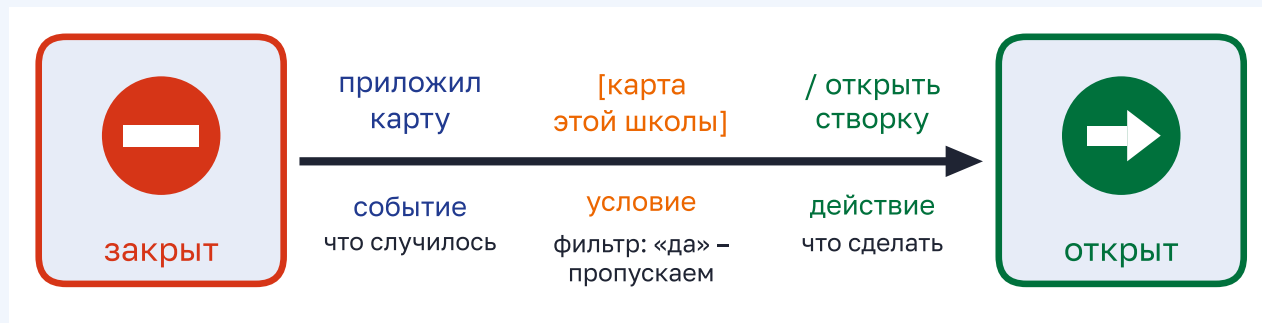


Рисунок 5.2 – Турникет у входа в школу

Приложил карту – это событие. Турникет проверяет, своя ли карта, – это условие. Если своя, открывает створку – это действие.

Запомни. Подпись на стрелке: событие [условие] / действие. «Вход» и «выход» срабатывают по одному разу – когда попали в состояние и когда ушли из него. Внутреннее событие действует, но состояние не меняет. Нет подходящей стрелки – событие пропало.

НА ПАЛЬЦАХ

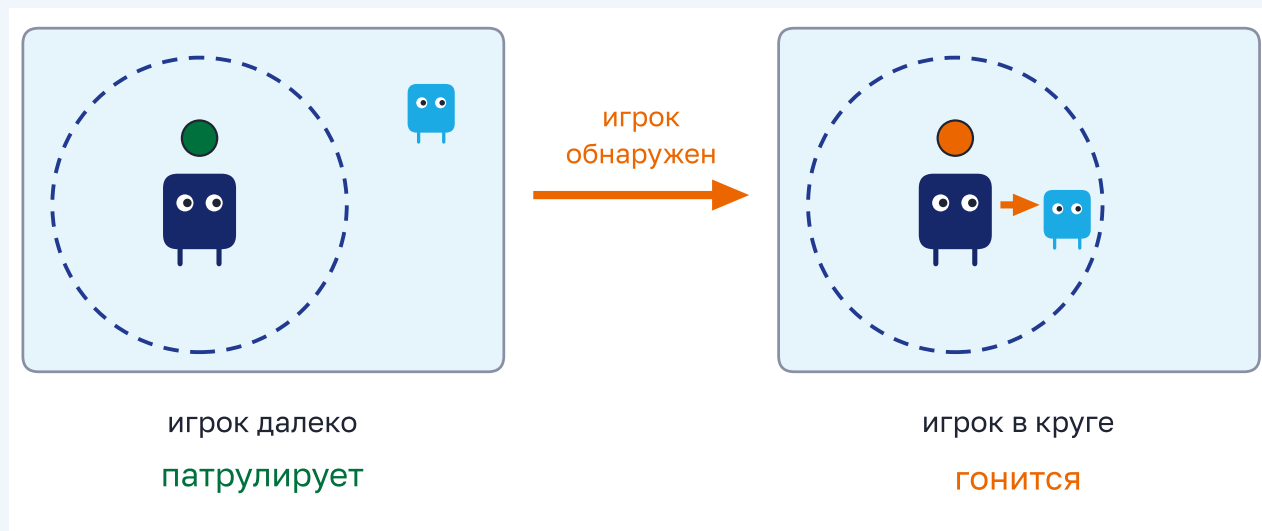


Рисунок 5.3 – Стражник и круг, в котором он замечает игрока

У стражника два режима: обход и погоня. Переключает их одно событие – игрок вошёл в круг. Шар над головой меняет цвет, и по нему видно, какой режим сейчас включён.

Запомни. Поведение NPC – его состояния и события между ними. То, что не меняет режим (дошёл до точки, купил зелье), записывают внутренним событием. Событие приходит только от включённого модуля: не включил поиск – «игрок обнаружен» не придёт никогда.

Запомни. Трассируй по таблице: было – событие – что произошло – стало. Каждое событие – отдельная строка, даже если оно пропало. Если машина ходит по кругу, отбрось целые круги делением с остатком.

Запомни. Недостижимое – не попасть. Тупик – не выйти. Лишний переход – никогда не сработает. Тупик бывает задуманным (конец игры), недостижимое состояние – почти всегда ошибка.

Запомни. Стрелка из рамки родителя действует из любого дочернего состояния. Стрелка в рамку ведёт в начальное дочернее. Общее для нескольких состояний событие – повод сложить их в одного родителя.

Разбор для старта

Задача 5.1 Дверь в подвал

разминка · трассировка, условие на переходе, внутреннее событие

Дверь в подвал работает по схеме из раздела 5.2. У героя есть ключ. События приходят по порядку: 1) потянули; 2) клик по замку; 3) толкнули; 4) потянули; 5) толкнули; 6) клик по замку; 7) потянули.

В каком состоянии дверь окажется в конце и сколько раз игрок увидит реплику «Заперто!»? Запиши номер состояния и число подряд, без пробелов.

Дано: схема двери (раздел 5.2), начальное состояние 1 «заперта», ключ есть; семь событий.

Найти: номер состояния в конце; сколько раз была реплика «Заперто!».

Решение

Шаг 1. Что происходит. У двери три состояния. Реплика «Заперто!» – внутреннее событие в состоянии 1: она появляется, когда дверь потянули, пока та заперта. Условие [есть ключ] у нас всегда верно. Трассируем по таблице.

Шаг 2. Таблица.

Таблица 5.1

Шаг	Было	Событие	Что произошло	Стало
1	1	потянули	«Заперто!» (1-й раз)	1
2	1	клик по замку	ключ есть – переход	2
3	2	толкнули	нет стрелки – пропало	2
4	2	потянули	переход	3
5	3	толкнули	переход	2
6	2	клик по замку	ключ есть – переход	1
7	1	потянули	«Заперто!» (2-й раз)	1

Шаг 3. Ответ. Дверь в состоянии 1, реплика была 2 раза. Записываем подряд: 12.

Представь это в игре. Герой отпер дверь, открыл её, закрыл, снова запер своим ключом – и потянул. Конечно, заперто. Реплика появлялась, только пока дверь была заперта: в самом начале и в самом конце.

 **Типичная ошибка.** На шаге 3 решить, что закрытая дверь от толчка открылась или заперлась. Из «закрыта» стрелка «толкнули» не выходит, событие пропадает. А ещё бывает, что на шаге 6 клик по замку считают бесполезным, хотя из «закрыта» он ведёт обратно в «заперта». С этой ошибкой дверь в конце окажется открытой, и выйдет ответ 31.

Ответ: 12 (дверь заперта, реплика «Заперто!» – 2 раза).

Своя задача по программе сферы, модуль 2

Проверь себя

5.1. Та же дверь в подвал (раздел 5.2), но ключа у героя **нет**. События: клик по замку, потянули, потянули, толкнули, клик по замку, потянули. В каком состоянии дверь в конце и сколько раз появилась реплика «Заперто!»? Запиши номер состояния и число подряд, без пробелов.

5.2. Умная лампа: 1 «выключена» (начальное), 2 «белый свет», 3 «ночник». Переходы: 1 → 2 «кнопка»; 2 → 3 «кнопка»; 3 → 1 «кнопка»; 2 → 1 «хлопок»; 3 → 1 «хлопок». События: кнопка, хлопок, хлопок, кнопка, кнопка, кнопка, кнопка, хлопок, кнопка. В каком состоянии лампа в конце? Запиши номер.

5.3. Другой сундук: 1 «закрыт» (начальное) и 2 «открыт». Стрелка 1 → 2: «клик / выдать 2 монеты», стрелка 2 → 1: «клик / выдать 1 монету». Внутри состояний действий нет. Игрок кликнул 7 раз. Сколько монет он получил?

5.4. Исправленный сундук: 1 «закрыт» (начальное), 2 «открыт», 3 «пуст». Переходы: 1 → 2 «клик» и 2 → 3 «клик». В «открыт» на входе: выдать 5 монет. В «пуст» на входе: реплика «Пусто»; там же внутреннее событие «клик / реплика «Пусто»». Игрок кликнул 8 раз. а) Сколько монет он получил? б) Сколько раз увидел реплику «Пусто»?

Переменные, условия и поиск ошибок



Игра помнит, сколько у героя монет, нашёл ли он ключ и сколько у него здоровья. Для этого у программиста есть переменные. А условия решают, что будет дальше: откроются ворота или нет. Здесь ты научишься читать чужую программу строку за строкой, как её читает компьютер, и находить в ней ошибку. Ещё посчитаем баланс в электронной таблице и увидим, зачем программу режут на части. На отборе такие задачи выглядят так: «что напечатает программа», «в какой строке ошибка», «что окажется в ячейке». Ответ – число, строка цифр или короткая формула.

НА ПАЛЬЦАХ



Рисунок 6.1 – Три коробки с именами. В каждой лежит одно значение

Положил в коробку новое значение – старое пропало. Коробка «монеты» помнит только последнее число.

Запомни. В коробке всегда одно значение, новое стирает старое. `монеты ← монеты + 5` читай справа налево: достань, прибавь, положи обратно. Без начального значения переменную не используют.

НА ПАЛЬЦАХ

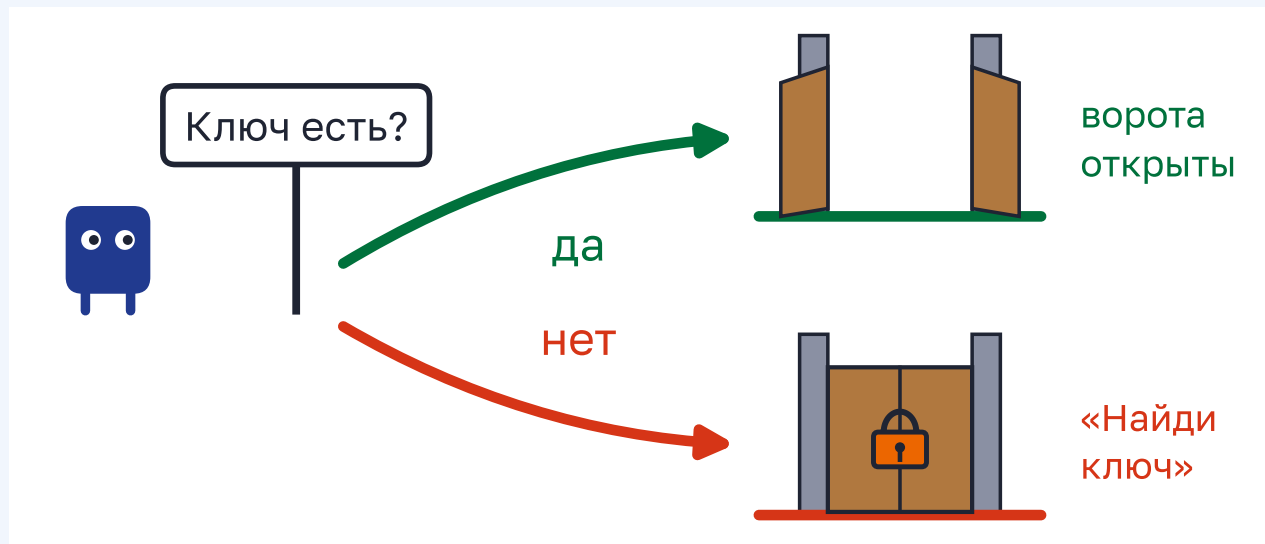


Рисунок 6.2 – Герой у развилки. Табличка спрашивает: «Ключ есть?»

«Да» – дорога к открытым воротам. «Нет» – к закрытым. Каждый раз игра идёт ровно по одной дороге.

Запомни. Условие верно – выполняется ветка `если`, неверно – ветка `иначе`. Одна из двух, не обе. «Хотя бы», «не меньше» – это `≥`. В Python сравнивают знаком `=`, а один `=` кладёт в коробку.



Рисунок 6.3 – Одна и та же цепочка проверок в двух порядках. У игрока 120 очков

Проверки идут сверху вниз, и первое «да» уводит игрока из цепочки. Если первой стоит мягкая проверка « ≥ 10 », золото не достанется никому.

Запомни. Цепочка иначе если останавливается на первом «да». Строгие проверки ставь выше мягких, особый случай – выше всех. В отдельных если без иначе побеждает последнее сработавшее.

Запомни. Трассировка – выполняем программу вручную, строка таблицы на каждый шаг. Первая строка – начальные значения, последняя – проверка, на которой цикл кончился. `range(1, 6)` – это 1, 2, 3, 4, 5, без шестёрки.

Запомни. Баг – «ждали одно, получили другое». Трассируй программу с числами игрока, ищи первое расхождение с описанием. Меняй одно место за раз и проверяй границу.

Запомни. Адрес ячейки – буква столбца и номер строки. Формула начинается с `=`. При копировании обычные ссылки сдвигаются на столько же строк и столбцов, а ссылки с долларом (`$B$7`) стоят на месте. Пустая ячейка в формуле – ноль.

Запомни. Функция – кусок программы с именем, параметрами и результатом. Она работает, только когда её вызвали. Числа при вызове ложатся в параметры по порядку. Модульная программа – из частей, у каждой своя работа.

Разбор для старта

Задача 6.1 Кошелёк торговца

разминка · переменные, присваивание, порядок команд

В игре у торговца волшебный кошелёк. Программа, которая следит за его монетами, такая:

```
монеты ← 12
монеты ← монеты + 8
монеты ← монеты - 5
цена ← 10
монеты ← монеты - цена
монеты ← монеты * 2
```

Сколько монет окажется в переменной `монеты` после выполнения программы? Ответ – целое число.

Дано: шесть команд присваивания; переменные `монеты` и `цена`.

Найти: значение `монеты` в конце.

Решение

Шаг 1. Что происходит. Команды выполняются строго по порядку, сверху вниз. Каждая берёт текущее значение из коробки, считает и кладёт результат обратно. Заведём трассировочную таблицу. Пока в коробку `цена` ничего не положили, ставим прочерк.

Шаг 2. Идём по строкам.

Таблица 6.1

Шаг	Что делает	монеты	цена
1	кладёт 12	12	–
2	+8	20	–
3	–5	15	–
4	цена ← 10	15	10
5	–цена	5	10
6	·2	10	10

На шаге 2 берём 12, прибавляем 8 – получаем 20. На шаге 3: $20 - 5 = 15$. Шаг 4 кладёт 10 в другую коробку, `монеты` не меняются. Шаг 5: $15 - 10 = 5$. Шаг 6: $5 \cdot 2 = 10$.

Шаг 3. Проверяем на Python.

```
монеты = 12
монеты = монеты + 8
монеты = монеты - 5
цена = 10
монеты = монеты - цена
монеты = монеты * 2
print(монеты)    # 10
```

Проверим другим путём. До удвоения торговец получил 8 и отдал $5 + 10 = 15$, в кошельке $12 + 8 - 15 = 5$ монет. Удвоили – 10.

❗ **Типичная ошибка.** Удвоить не то число: сосчитать $12 \cdot 2$ в начале или удвоить всё, что торговец получил ($20 \cdot 2 = 40$), а траты вычесть потом. Удвоение – последняя команда. Она действует на то, что лежит в коробке к этому моменту, то есть на 5. Ещё путаются со строкой `цена ← 10`: кажется, что она что-то отнимает у монет. Нет, она кладёт число в другую коробку.

Ответ: 10.

Своя задача по программе сферы, модуль 2

Проверь себя

6.1. Программа считает опыт героя:

```
опыт ← 25
опыт ← опыт + 15
бонус ← 3
опыт ← опыт * бонус
опыт ← опыт - 50
```

Чему равен `опыт` после выполнения программы?

6.2. Игрок хочет поменять местами предметы в двух ячейках рюкзака. Предметы обозначены номерами. Программист написал:

```
левая ← 4  
правая ← 9  
левая ← правая  
правая ← левая
```

а) Какие числа окажутся в `левая` и `правая` после программы? Запиши два числа через пробел.

б) Какой вариант на самом деле меняет предметы местами? Запиши его номер.

1. `правая ← левая` , потом `левая ← правая`
2. `запас ← левая` , потом `левая ← правая` , потом `правая ← запас`
3. `левая ← правая` , потом `запас ← левая` , потом `правая ← запас`

6.3. За прохождение уровня дают звёзды:

```
если время ≤ 60 то звёзды ← 3  
иначе если время ≤ 90 то звёзды ← 2  
иначе если время ≤ 120 то звёзды ← 1  
иначе звёзды ← 0
```

Время – в секундах. Пять игроков прошли уровень за 90, 61, 121, 60 и 15 секунд. Запиши число звёзд каждого по порядку, без пробелов.

6.4. Другой программист записал звёзды отдельными проверками, без «иначе»:

```
звёзды ← 0  
если время ≤ 60 то звёзды ← 3  
если время ≤ 90 то звёзды ← 2  
если время ≤ 120 то звёзды ← 1
```

Сколько звёзд получают игроки со временем 50, 85, 110 и 130 секунд? Запиши числа по порядку, без пробелов.

Сюжет и диалоги: язык Ink



Помочь лису или прогнать? Пойти в лес или к реке? В игре с выборами это решаешь ты, а сценарист заранее рисует схему: куски текста и стрелки между ними. В «Урсуле» такие истории пишут на языке Ink. Здесь ты научишься читать короткую историю на Ink, считать по схеме прохождения и концовки и находить ошибки в чужом коде. Задачи в главе устроены как на отборе: дан кусок истории, а спрашивают, что увидит игрок, сколько всего концовок или в какой строке ошибка. Ответ – число, строка цифр или номера строк.

НА ПАЛЬЦАХ



Рисунок 7.1 – Книга-игра: каждый абзац говорит, какой читать следующим

Абзац – это узел. «Читай абзац 7» – переход. Куда идти, решает читатель.

Запомни. Узел – кусок истории, выбор – кнопка, переход – куда она ведёт, концовка – узел, где всё кончается. Выбор значимый, если у него есть последствия, которые игрок замечает.

НА ПАЛЬЦАХ

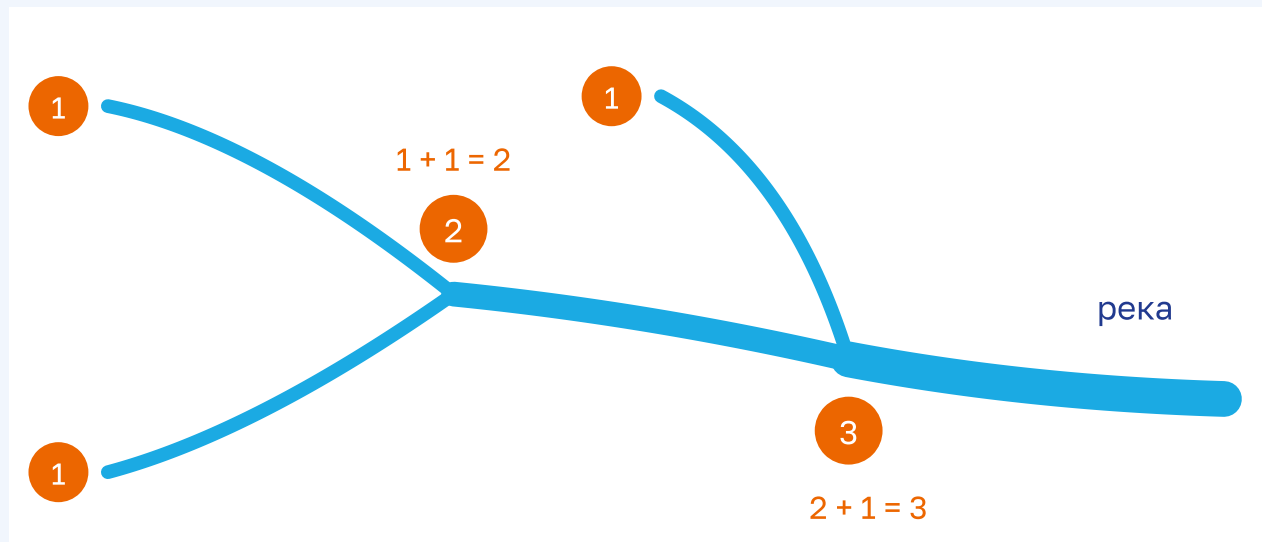


Рисунок 7.2 – Ручьи сливаются в реку

Дороги к узлу складываются, как ручьи: сколько путей пришло по каждой стрелке, столько и слилось в узле.

Запомни. У начала – 1, у узла – сумма чисел тех, кто в него ведёт. Две кнопки в один узел – две стрелки. Всего прохождений – сумма чисел у всех концовок.

НА ПАЛЬЦАХ

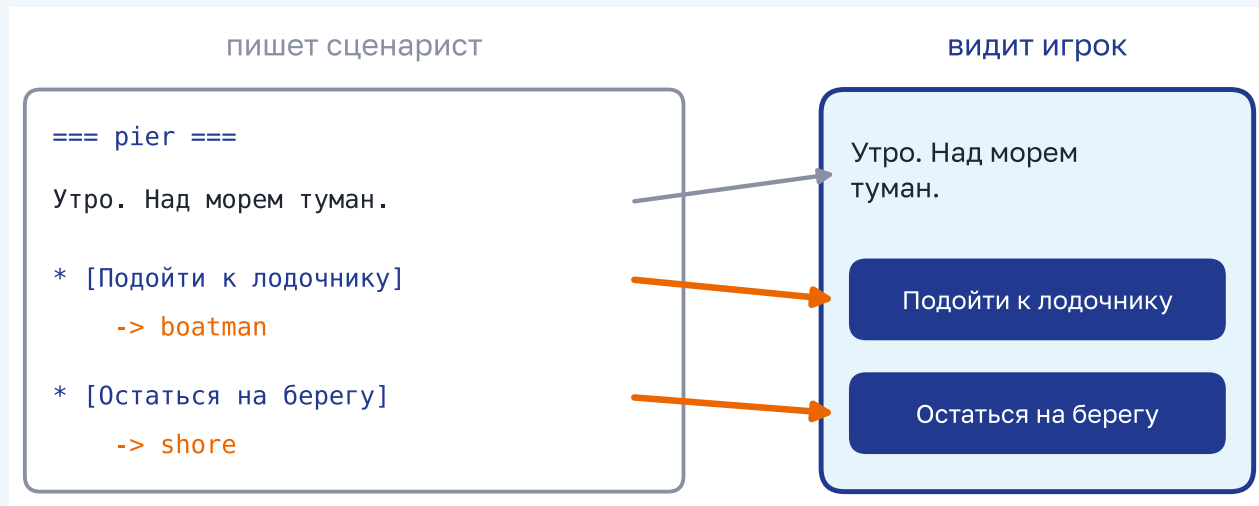


Рисунок 7.3 – Слева – что пишет сценарист, справа – что видит игрок

Обычная строка – это текст на экране. Строка со звёздочкой – кнопка. Стрелка говорит, куда кнопка ведёт.

Запомни. `=== имя ===` – начало узла, `→ имя` – переход, `→ END` – конец истории. `*` – кнопка, которая исчезает, `+` – кнопка, которая остаётся. Текст в `[ ]` виден только на кнопке. `-` – место сбора.

Запомни. `VAR` заводит переменную, `~` меняет её, `{имя}` показывает. `* {условие} [...]` – кнопка появится, только если условие верно. Условия перехода проверяются сверху вниз, срабатывает первое верное. «Равно» – это `=`.

Запомни. Концовка недостижима, если в неё не ведёт ни одна стрелка или условие на стрелке не выполнится никогда. Для проверки считай самое большое и самое маленькое значение переменной. Без `~` команда становится текстом.

Запомни. Узнаёшь героя без подписи – голос есть. Одна реплика – одна мысль, примерно до 15 слов. Выбор, который ведёт туда же, куда и другой, – не выбор.

Разбор для старта

Задача 7.1 Три вопроса о сюжете

разминка · нелинейный сюжет, знаки `Ink`, значимый выбор

Ответь на три вопроса. В ответ запиши цифры выбранных вариантов по порядку, без пробелов.

а) В какой из игр сюжет ветвится: от выборов игрока зависит, как пойдёт история? 1 – «Тетрис»; 2 – Super Mario Bros.; 3 – Minecraft: Story Mode; 4 – Angry Birds.

б) Сценарист хочет, чтобы кнопка «Спросить про остров» была видна каждый раз, когда игрок возвращается к лодочнику. Каким знаком начать строку с этим выбором в Ink? 1 – *; 2 – +; 3 – →; 4 – ~.

в) Как называют выбор, после которого игрок видит последствия: персонаж это запомнил, история пошла по-другому? 1 – случайный; 2 – значимый; 3 – линейный; 4 – запасной.

Дано: три вопроса, в каждом четыре варианта и ровно один верный.

Найти: три цифры по порядку.

Решение

Шаг 1. Вопрос а). Проверим каждую игру вопросом: «Могут ли два игрока увидеть разные истории?» В «Тетрисе» истории нет совсем – падают фигуры. В Super Mario Bros. история одна: спасти принцессу. Игрок выбирает, как прыгать, но сюжет от этого не меняется. В Angry Birds свиньи украли яйца, птицы мстят – и тоже без развилки. В Minecraft: Story Mode игрок принимает решения в разговорах, и от них зависит, как поведут себя персонажи и что случится дальше. Ответ – 3.

Шаг 2. Вопрос б). Выбор со звёздочкой * исчезает после первого нажатия. Вернётся игрок к лодочнику – кнопки уже нет. Чтобы кнопка оставалась, нужен плюс +. Стрелка → – переход, кнопку она не рисует. Тильда ~ – команда для переменной. Ответ – 2.

Шаг 3. Вопрос в). Выбор с заметными последствиями называют значимым – так его называли и на вебинаре сферы. «Запасной» – это выбор * → узел без текста, который Ink делает сам, когда кнопок не осталось. «Линейный» бывает сюжет, а не выбор. Ответ – 2.

Шаг 4. Собираем ответ. Цифры по порядку: 3, 2, 2.

Заметь: в каждом вопросе мы не просто выбрали ответ, а для каждого из трёх лишних вариантов назвали причину. Не нашлась причина для какого-то варианта – вернись к нему, ошибка сидит там.

 **Типичная ошибка.** В вопросе а) выбрать Super Mario Bros.: там есть тайные трубы, которые перебрасывают через несколько миров сразу. Но трубы только сокращают дорогу по уровням: принцесса и финал остаются теми же. В вопросе б) спутать * и +: звёздочку ставят чаще, и рука тянется к ней по привычке.

Ответ: 322.

По мотивам задачи НТО 2025/26, профиль «Разработка компьютерных игр» (старший трек, 8–11 классы), первый отборочный этап, инженерный тур, «Общий. Жанры игр»

Проверь себя

7.1. Соедини знак Ink с тем, что он делает. Знаки: 1 – ; 2 – ; 3 – ; 4 – ; 5 – ; 6 –  в начале строки. Что они делают: А – выбор, который остаётся; Б – начало узла; В – изменить значение переменной; Г – перейти в другой узел; Д – место сбора, где ветки снова сходятся; Е – выбор, который исчезает после нажатия. Запиши номера знаков для А, Б, В, Г, Д, Е по порядку, без пробелов.

7.2. История про школьный буфет:

```
→ canteen

≡≡ canteen ≡≡
Школьный буфет. До звонка пять минут.
* Взять компот.
    Компот из сухофруктов, ещё тёплый.
    → canteen
* [Взять булочку]
    Булочка с маком.
    → canteen
+ [Посмотреть меню]
    Сегодня сырники.
    → canteen
+ [Пойти на урок] → END
```

Игрок нажал по очереди: «Взять компот.», «Посмотреть меню», «Взять булочку», «Пойти на урок». а) Сколько кнопок было на экране при каждом из четырёх посещений буфета? Запиши четыре числа подряд. б) Сколько строк текста (не считая кнопок) прочитает игрок за всю игру?

7.3. Класс на экскурсии в зоопарке:

→ entrance

=== entrance ===

Вход в зоопарк. Куда пойдёт класс?

* [К медведям] → bears

* [К жирафам] → giraffes

=== bears ===

Бурый медведь спит на солнце.

* [К тиграм] → tigers

* [В кафе] → cafe

=== giraffes ===

Жираф тянется к ветке.

* [К тиграм] → tigers

* [К пингвинам] → penguins

=== tigers ===

Амурский тигр ходит вдоль решётки.

* [В кафе] → cafe

* [К выходу] → exit

=== penguins ===

Пингвины ныряют по очереди.

* [В кафе] → cafe

* [Идти за пингвинами до конца дорожки] → lost

=== cafe ===

Весь класс ест мороженое.

→ END

=== exit ===

Автобус уже ждёт у ворот.

→ END

=== lost ===

Ты отстал от класса, и учительница звонит тебе на телефон.

→ END

Сколько у истории концовок и сколько разных прохождений? Ответ – два числа.

7.4. Вернись к истории про Бублика из задачи 7.2. а) В скольких прохождениях игрок поговорил с соседкой? б) Сколько кнопок нажимает игрок в самом коротком прохождении и сколько – в самом длинном?

3D и визуал: Blender



Сундук, ёлка, робот-помощник в игре – всё это 3D-модели, а собраны они из самых простых кусочков: точек, отрезков и плоских граней. Здесь мы научимся их считать, посмотрим, как Blender сам достраивает вторую половину модели и размножает забор из одной доски, как не сделать стул высотой с девятиэтажку и чем «1920 × 1080» отличается от «PNG». На отборе такие задачи короткие: «сколько вершин получится», «какой масштаб поставить», «сколько процентов экрана занято», «найди лишнее в ряду». Ответ – число или строка цифр.

НА ПАЛЬЦАХ

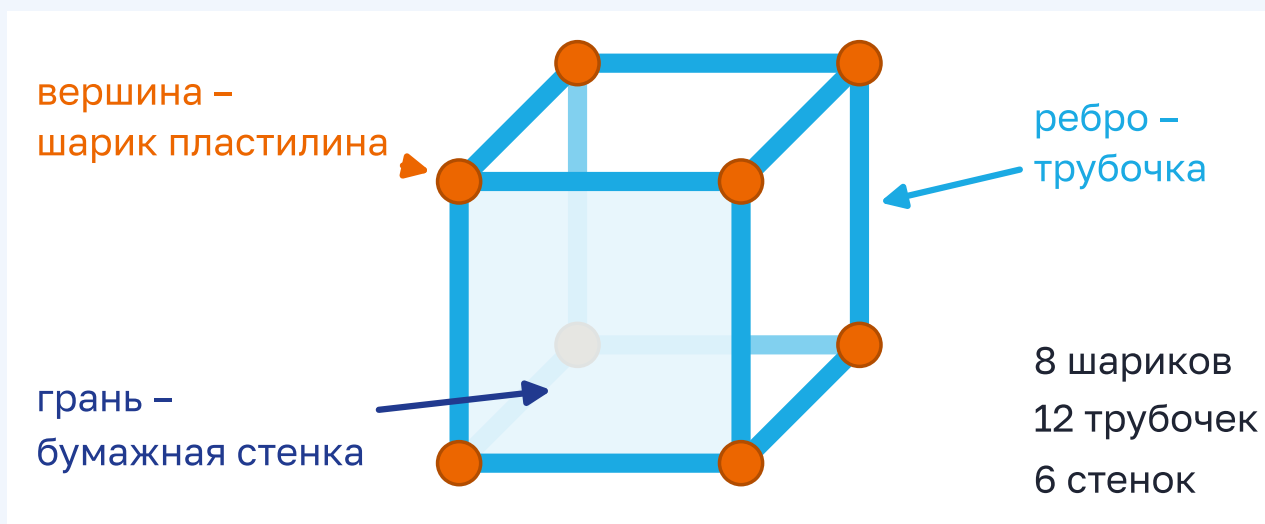


Рисунок 8.1 – Каркас куба из трубочек для коктейля и пластилиновых шариков

Шарики – вершины, трубочки – рёбра, бумажные стенки – грани. Из таких же кусочков сделана любая модель в игре.

Запомни. Вершина – точка, ребро – отрезок, грань (полигон) – плоский кусок. Призма: $2n, 3n, n + 2$. Пирамида: $n + 1, 2n, n + 1$. Проверка: $V - P + \Gamma = 2$. Грань с n углами – это $n - 2$ треугольника.

НА ПАЛЬЦАХ

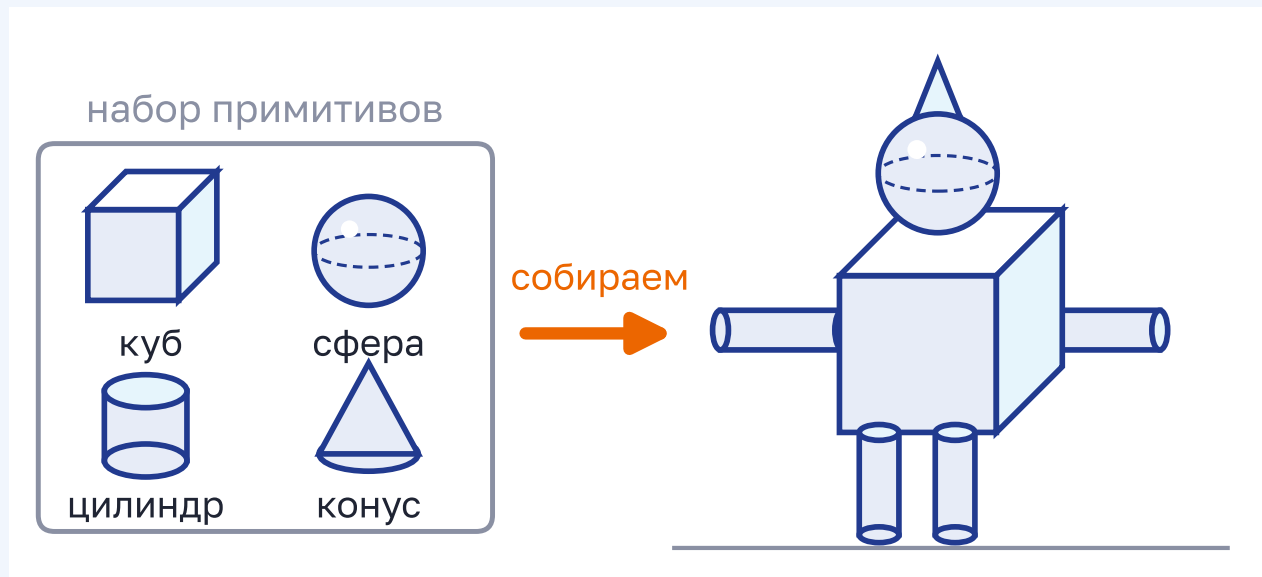


Рисунок 8.2 – Слева – готовые детали Blender, справа – робот, собранный из них

Сначала собираем фигуру из готовых деталей, как из конструктора. Мелочи – потом.

Запомни. Модель начинают с примитивов и собирают черновик (блокинг), детали – потом. Shift + A – добавить, G, R, S – двигать, вращать, масштабировать, Tab – режим редактирования. Куб по умолчанию – 2 м, у цилиндра 32 вершины по кругу.

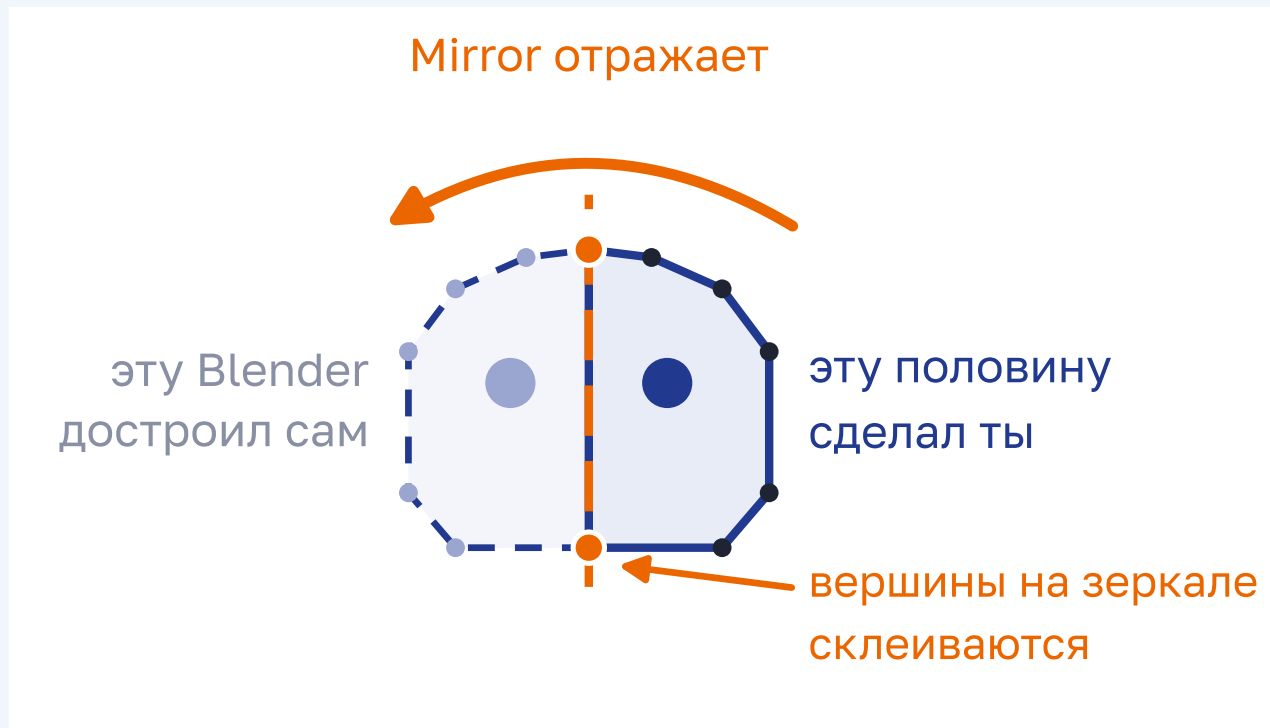


Рисунок 8.3 – Mirror отражает сделанную половину и склеивает её со второй по средней линии

Как клякса на сложенном листе: рисуешь половину бабочки, складываешь бумагу – бабочка целая.

Запомни. Mirror – делаешь половину, вторую строит Blender. Вершины: $2V - k$, где k – вершины на зеркале (при включённом Merge). Зеркало проходит через центр объекта: $x' = 2x_0 - x$, при центре в нуле просто $x' = -x$.

Запомни. Count – всего штук, с оригиналом. Шаг $s = k \cdot a$, длина ряда $L = a + (n - 1) \cdot s$. Вершины умножаются на Count. Модификаторы в стопке работают сверху вниз, копии перемножаются. Объект после модификаторов – всё равно один.

Запомни. 1 единица Blender = 1 м, Z – вверх. Размер = размер сетки · Scale; куб по умолчанию – 2 м, поэтому нужный Scale – это нужный размер, делённый на 2. Масштаб применяют (Ctrl + A → Scale). В игру – файл GLB с применёнными модификаторами.

Запомни. Пиксель – одна клетка одного цвета. Разрешение – ширина × высота в пикселях; всего пикселей – произведение. Вдвое больше по каждой стороне – вчетверо больше пикселей. Формат (PNG, JPG) – как упакован файл, разрешение (1920 × 1080) – сколько в нём пикселей. SVG – вектор.

Запомни. Эскиз – идея на бумаге, референсы – виды спереди и сбоку для моделирования. Силуэт проверяют заливкой чёрным: героя должно быть видно по форме, а не только по цвету. Персонажа моделируют в Т-позе.

Разбор для старта

Задача 8.1 Карандаш для инвентаря

разминка · вершины, рёбра, грани, призма, формула Эйлера

Для игры-квеста нужен карандаш, который лежит в инвентаре героя. Художник сделал его low-poly: шестигранный незаточенный карандаш – это шестиугольная призма, закрытая с обоих концов. Сколько у этой модели вершин, рёбер и граней? Ответ – три числа через пробел в этом порядке.

Дано: шестиугольная призма, $n = 6$.

Найти: В, Р, Г – ?

Решение

Шаг 1. Что за тело. Призма: сверху и снизу одинаковые шестиугольники, по бокам – прямоугольники. Считаем по частям: верх, низ, бока.

Шаг 2. Вершины. У шестиугольника 6 углов. Сверху 6 вершин, снизу $6 + 6 = 12$.

Шаг 3. Рёбра. По краю верхнего шестиугольника 6 рёбер, по краю нижнего – ещё 6, и 6 вертикальных рёбер соединяют верх с низом: $6 + 6 + 6 = 18$.

Шаг 4. Грани. Верх, низ и 6 боковых граней – по одной на каждую сторону шестиугольника: $1 + 1 + 6 = 8$.

Шаг 5. Проверка Эйлера. $12 - 18 + 8 = 2$ – сошлось.

А по формулам? У призмы $2n = 12$, $3n = 18$, $n + 2 = 8$ – те же числа. Возьми настоящий карандаш: у него 6 граней-боков и два торца, всего 8.

⚠ Чаше всего забывают торцы и отвечают «6 граней» – это только бока. Другая ошибка – считать рёбра по граням: у 6 боковых граней по 4 ребра, у двух торцов по 6, всего $24 + 12 = 36$. Но каждое ребро – общая сторона двух граней, и так оно посчитано дважды: $36 : 2 = 18$. Если Эйлер не дал 2, первым делом ищи такой двойной счёт.

Ответ: 12 18 8.

Своя задача по программе сферы, модуль 3

Проверь себя

8.1. Крыша сторожевой башни в игре – пятиугольная пирамида вместе с дном. Сколько у неё вершин, рёбер и граней? Ответ – три числа через пробел.

8.2. Для настольной игры внутри видеоигры смоделировали двадцатигранный кубик: у него 20 треугольных граней и 12 вершин. Сколько у модели рёбер?

8.3. Колонну для храма делают из стандартного цилиндра (радиус 1 м, высота 2 м). Колонна должна быть диаметром 0,4 м и высотой 3 м. Какие значения Scale поставить по X, Y и Z? Ответ – три числа через пробел.

8.4. Новичок моделировал стол и думал, что одна клетка сетки – это сантиметр. Высота стола у него получилась 75 единиц. а) Какой высоты в метрах окажется стол в игре? б) На какое число нужно умножить Scale стола, чтобы он стал высотой 0,75 м? Ответ – два числа через пробел.

Ответы и решения

Сначала реши задачу сам и только потом сверяйся. Не сошлось – найди в решении шаг, на котором твой путь разошёлся с нашим.

К главе 1. Игры и жанры: язык геймдизайнера

1.1. Ответ: 4123.

Подсказка: цифры пишем для букв А–Г по порядку. Если вышло «2341», ты записал буквы для частей 1–4.

1.2. Ответ: 35142.

1.3. Ответ: 6231.

1.4. Ответ: 56231.

Подсказка: у игры может быть два жанра сразу; внутри каждой игры цифры идут по возрастанию, поэтому «56», а не «65».

К главе 2. Геймдизайн: механики, баланс, документ игры

2.1. Ответ: 8.

2.2. Ответ: 75; на 6,25 %.

Подсказка: второй процент берут от 100 – от урона после зелья, а не от исходных 80. Поэтому +25 % и –25 % не уничтожают друг друга.

2.3. Ответ: 4512.

2.4. Ответ: 1321.

Подсказка: про каждый набросок спроси по очереди: что игрок получает за действие? куда это можно потратить?

К главе 3. Левел-дизайн: карты и пути

3.1. Ответ: 21 м.

3.2. Ответ: 1,5 с.

Подсказка: по сторонам шагов $7 + 3 = 10$, а с диагоналями – столько, сколько большая разница, то есть 7.

3.3. Ответ: (11; 5).

3.4. Ответ: 9.

Подсказка: раскрой все циклы в столбик – получится 12 команд; иди по ним пальцем по карте, а не складывай сдвиги.

К главе 4. Логика, выборы и случайность

4.1. Ответ: 231.

4.2. Ответ: 2314.

Подсказка: начни с мельника. Он не в крайнем доме, значит, в доме 2 или 3, а пекарь правее его и не в доме 4.

4.3. Ответ: 10.

Подсказка: попробуй «пусть Том – лжец». Тогда его фраза оказывается правдой.

4.4. Ответ: 100.

Подсказка: начни с В – его фраза похожа на «Я лжец». Потом заметь, что А и Б не могут оба сказать правду.

К главе 5. Машины состояний: как думает NPC

5.1. Ответ: 13.

Подсказка: без ключа условие [есть ключ] ложно, и клик по замку ничего не меняет.

5.2. Ответ: 2.

5.3. Ответ: 11 монет.

Подсказка: монеты теперь выдаются на стрелках, а переходов «туда» и «обратно» при 7 кликах получается разное число.

5.4. Ответ: а) 5 монет; б) 7 раз.

Подсказка: «вход» в «пуст» срабатывает один раз, а каждый следующий клик – внутреннее событие.

К главе 6. Переменные, условия и поиск ошибок

6.1. Ответ: 70.

6.2. Ответ: а) 9 9; б) 2.

Подсказка: после строки `левая ← правая` четвёрка пропала навсегда. Чтобы её не потерять, сначала отложи её в третью коробку.

6.3. Ответ: 22033.

6.4. Ответ: 1110.

Подсказка: здесь нет «иначе», поэтому проверяются все три строки. Кто прошёл быстрее 120 секунд, у того последняя строка перезапишет звёзды единицей.

К главе 7. Сюжет и диалоги: язык Ink

7.1. Ответ: 415263.

7.2. Ответ: а) 4332; б) 8 строк.

Подсказка: текст «Взять компот.» написан без квадратных скобок, поэтому после нажатия он появляется и в истории.

7.3. Ответ: 3 концовки, 7 прохождений.

7.4. Ответ: а) 7; б) 2 и 4 кнопки.

Подсказка: посчитай прохождения, которые начинаются из узла `neighbor`, так же, как от начала: у `neighbor` пиши 1.

К главе 8. 3D и визуал: Blender

8.1. Ответ: 6 10 6.

8.2. Ответ: 30.

Подсказка: по Эйлеру $P = B + \Gamma - 2$. Или иначе: у 20 треугольников $20 \cdot 3 = 60$ сторон, и каждое ребро – общая сторона двух граней.

8.3. Ответ: 0,2 0,2 1,5.

Подсказка: по X и Y у цилиндра размер задаёт диаметр – 2 м, а не радиус 1 м.

8.4. Ответ: 75 0,01.

Всё, что может понадобиться под рукой во время решения: формулы и правила из глав, мини-шпаргалки по ПРИМС, «Урсуле», Ink и Blender, словарик слов из мира разработки игр и все «Запомни» из глав на нескольких страницах.

Шпаргалка: всё «Запомни» из глав

Глава 1. Игры и жанры: язык геймдизайнера

Запомни. Цель – чего хочешь. Правила – что можно и что за это будет. Препятствие – что мешает. Обратная связь – как игра показывает результат. Если в задаче просят «сопоставить часть игры с описанием», спроси себя про каждую фразу: «Она про то, чего добиться, как устроен мир, что мешает или как игра ответила?»

Запомни. Жанр определяют по главному действию игрока. Тема и картинка не в счёт: игра про ферму бывает и симулятором, и стратегией. Если признаков двух жанров поровну, перед тобой смесь – и тогда в задаче обычно просят отметить все подходящие.

Запомни. Проверка на механику: разложи фразу на «событие → условие → изменение мира». Разложилась – механика. Не разложилась – перед тобой сеттинг, сюжет или атмосфера. А в инвентаре неполная стопка занимает целую ячейку, поэтому число ячеек – это деление с округлением вверх.

Запомни. Квест – и жанр, и задание. Шаги задания: выдача → цель → препятствие → условие → исход. Состояний у задания четыре: не взят, взят, выполнен, провален, – и в каждый момент только одно. Свободное дело можно вставить в ряд из трёх дел четырьмя способами, в ряд из четырёх – пятью.

Запомни. Прочитай, в каком виде нужен ответ, **до** того, как начнёшь решать. «Все верные» – по возрастанию. «Сопоставь» – по порядку букв. Лишний пробел или запятая могут сделать верный ответ неверным.

Глава 2. Геймдизайн: механики, баланс, документ игры

Запомни. Цикл – кольцо: действие → награда → новая цель → снова действие. Выпало любое звено – кольцо рвётся, и игрок уходит. Короткие циклы (секунды) вложены в длинные (уровень, вся игра).

Запомни. Удары, ходы и порции обмена бывают только целыми – округляй вверх, чтобы хватило. Минус $p\%$ – умножить на $(1 - p/100)$, плюс $p\%$ – на $(1 + p/100)$. Проценты подряд перемножают, а не складывают.

Запомни. От первого числа ряда до n -го – $n - 1$ шаг, как и от 1-го уровня до n -го – $n - 1$ переход. Шаг «плюс»: первое $+(n - 1) \cdot$ прибавка. Шаг «умножить»: первое $\cdot$ множитель в степени $n - 1$. Сумму надёжнее считать таблицей со строкой «всего».

Запомни. Время = недостающее : доход, округлить вверх. Если доход дан «за действие», сначала находим число действий (вверх), потом время. Улучшение окупается за «цена : прибавка» минут.

Запомни. Плейтест проводят на тех, кто игру ещё не видел. Смотри и записывай, а не объясняй. Долю ушедших считают от дошедших до этапа, а не от всех.

Глава 3. Левел-дизайн: карты и пути

Запомни. Координаты $(x; y)$: сначала столбец, потом строка. Без диагоналей шагов столько, сколько разница по x плюс разница по y . С диагоналями – сколько большая из двух разниц. Клетки в метры – умножением на размер клетки.

Запомни. Каждая команда движения меняет одну координату на единицу. Чтобы узнать, где окажется робот, проще всего сложить сдвиги: сколько всего раз вправо минус сколько раз влево – это изменение x ; вверх минус вниз – изменение y . А вот врежется ли он в стену, по сумме не узнать: тут придётся пройти путь клетка за клеткой.

Запомни. Волна: старт – 0, соседи – 1, их соседи – 2... Число в финише – длина кратчайшего пути. Путь – обратным ходом, по числам на 1 меньше. Клетку, где число уже стоит, не переписываем.

Запомни. Граф: комната – вершина, проход – ребро, над ребром длина. Кратчайший путь – метками: закрепляй комнату с самым маленьким числом и обновляй соседей. Ключи – по кругам: куда можно дойти → какие ключи там → какие двери они открывают.

Запомни. Только вправо и вверх: клетка = слева + снизу. На нижнем и левом краю – по 1, но только до первой запрещённой клетки. Запрещённая клетка – 0. Если выбор идёт по шагам, варианты шагов перемножают: так считают путь через обязательную клетку.

Запомни. Игрок идёт туда, где свет, где виден ориентир и куда ведут «хлебные крошки». Одинаковые коридоры, темнота везде и подсказки текстом каждые десять секунд – признаки слабого уровня.

Глава 4. Логика, выборы и случайность

Запомни. Строки – кто, столбцы – где. Подсказка с «не» – крестик. Одна свободная клетка в строке – точка. Точка закрывает свой столбец для всех остальных.

Запомни. Предположи – проверь – ищи противоречие. Противоречие значит, что предположение неверно. «А говорит, что Б лжец» – они разные. «А говорит, что Б рыцарь» – они одинаковые.

Запомни. И – нужны обе части. ИЛИ – хватит одной, обе тоже годятся. НЕ – наоборот. Порядок: скобки, НЕ, И, ИЛИ.

Запомни. Выборы один за другим – умножай. Разные ветки – складывай. k развилок по 2 варианта без схождений – 2^k концовок. Ветки сходятся – рисуй карту и пиши у каждого места сумму чисел, из которых в него пришли.

Запомни. Шанс = подходящие исходы : все исходы. $20\% = 0,2 = 1/5 = \text{«1 из 5»}$ – в среднем, а не каждый пятый раз. Все исходы вместе – 100 %. Независимые события сразу – шансы умножаются. На двух кубиках 36 исходов, чаще всего выпадает сумма 7.

Запомни. Прошлые неудачи не меняют шанс следующей попытки, если в правилах нет счётчика. «Хотя бы раз за n попыток» считают через «ни разу»: $1 - (1 - p)^n$. Даже много попыток не дают 100 %.

Глава 5. Машины состояний: как думает NPC

Запомни. Машина состояний – это прямоугольники-состояния, стрелки-переходы и события на стрелках. Объект всегда находится только в одном состоянии. Стрелка от чёрной точки показывает, с чего всё начинается.

Запомни. Подпись на стрелке: событие [условие] / действие. «Вход» и «выход» срабатывают по одному разу – когда попали в состояние и когда ушли из него. Внутреннее событие действует, но состояние не меняет. Нет подходящей стрелки – событие пропало.

Запомни. Поведение NPC – его состояния и события между ними. То, что не меняет режим (дошёл до точки, купил зелье), записывают внутренним событием. Событие приходит только от включённого модуля: не включил поиск – «игрок обнаружен» не придёт никогда.

Запомни. Трассируй по таблице: было – событие – что произошло – стало. Каждое событие – отдельная строка, даже если оно пропало. Если машина ходит по кругу, отбрось целые круги делением с остатком.

Запомни. Недостижимое – не попасть. Тупик – не выйти. Лишний переход – никогда не сработает. Тупик бывает задуманным (конец игры), недостижимое состояние – почти всегда ошибка.

Запомни. Стрелка из рамки родителя действует из любого дочернего состояния. Стрелка в рамку ведёт в начальное дочернее. Общее для нескольких состояний событие – повод сложить их в одного родителя.

Глава 6. Переменные, условия и поиск ошибок

Запомни. В коробке всегда одно значение, новое стирает старое. `монеты < монеты + 5` читай справа налево: достань, прибавь, положи обратно. Без начального значения переменную не используют.

Запомни. Условие верно – выполняется ветка `если`, неверно – ветка `иначе`. Одна из двух, не обе. «Хотя бы», «не меньше» – это `≥`. В Python сравнивают знаком `=`, а один `=` кладёт в коробку.

Запомни. Цепочка иначе если останавливается на первом «да». Строгие проверки ставь выше мягких, особый случай – выше всех. В отдельных если без иначе побеждает последнее сработавшее.

Запомни. Трассировка – выполняем программу вручную, строка таблицы на каждый шаг. Первая строка – начальные значения, последняя – проверка, на которой цикл кончился. `range(1, 6)` – это 1, 2, 3, 4, 5, без шестёрки.

Запомни. Баг – «ждали одно, получили другое». Трассируй программу с числами игрока, ищи первое расхождение с описанием. Меняй одно место за раз и проверяй границу.

Запомни. Адрес ячейки – буква столбца и номер строки. Формула начинается с `=`. При копировании обычные ссылки сдвигаются на столько же строк и столбцов, а ссылки с долларом (`$B$7`) стоят на месте. Пустая ячейка в формуле – ноль.

Запомни. Функция – кусок программы с именем, параметрами и результатом. Она работает, только когда её вызвали. Числа при вызове ложатся в параметры по порядку. Модульная программа – из частей, у каждой своя работа.

Глава 7. Сюжет и диалоги: язык Ink

Запомни. Узел – кусок истории, выбор – кнопка, переход – куда она ведёт, концовка – узел, где всё кончается. Выбор значимый, если у него есть последствия, которые игрок замечает.

Запомни. У начала – 1, у узла – сумма чисел тех, кто в него ведёт. Две кнопки в один узел – две стрелки. Всего прохождений – сумма чисел у всех концовок.

Запомни. `≡ имя ≡` – начало узла, `→ имя` – переход, `→ END` – конец истории. `*` – кнопка, которая исчезает, `+` – кнопка, которая остаётся. Текст в `[ ]` виден только на кнопке. `-` – место сбора.

Запомни. VAR заводит переменную, ~ меняет её, {имя} показывает. * {условие} [...] – кнопка появится, только если условие верно. Условия перехода проверяются сверху вниз, срабатывает первое верное. «Равно» – это $=$.

Запомни. Концовка недостижима, если в неё не ведёт ни одна стрелка или условие на стрелке не выполнится никогда. Для проверки считай самое большое и самое маленькое значение переменной. Без ~ команда становится текстом.

Запомни. Узнаёшь героя без подписи – голос есть. Одна реплика – одна мысль, примерно до 15 слов. Выбор, который ведёт туда же, куда и другой, – не выбор.

Глава 8. 3D и визуал: Blender

Запомни. Вершина – точка, ребро – отрезок, грань (полигон) – плоский кусок. Призма: $2n, 3n, n + 2$. Пирамида: $n + 1, 2n, n + 1$. Проверка: $V - P + G = 2$. Грань с n углами – это $n - 2$ треугольника.

Запомни. Модель начинают с примитивов и собирают черновик (блокинг), детали – потом. Shift + A – добавить, G, R, S – двигать, вращать, масштабировать, Tab – режим редактирования. Куб по умолчанию – 2 м, у цилиндра 32 вершины по кругу.

Запомни. Mirror – делаешь половину, вторую строит Blender. Вершины: $2V - k$, где k – вершины на зеркале (при включённом Merge). Зеркало проходит через центр объекта: $x' = 2x_0 - x$, при центре в нуле просто $x' = -x$.

Запомни. Count – всего штук, с оригиналом. Шаг $s = k \cdot a$, длина ряда $L = a + (n - 1) \cdot s$. Вершины умножаются на Count. Модификаторы в стопке работают сверху вниз, копии перемножаются. Объект после модификаторов – всё равно один.

Запомни. 1 единица Blender = 1 м, Z – вверх. Размер = размер сетки · Scale; куб по умолчанию – 2 м, поэтому нужный Scale – это нужный размер, делённый на 2. Масштаб применяют (Ctrl + A → Scale). В игру – файл GLB с применёнными модификаторами.

Запомни. Пиксель – одна клетка одного цвета. Разрешение – ширина × высота в пикселях; всего пикселей – произведение. Вдвое больше по каждой стороне – вчетверо больше пикселей. Формат (PNG, JPG) – как упакован файл, разрешение (1920 × 1080) – сколько в нём пикселей. SVG – вектор.

Запомни. Эскиз – идея на бумаге, референсы – виды спереди и сбоку для моделирования. Силуэт проверяют заливкой чёрным: героя должно быть видно по форме, а не только по цвету. Персонажа моделируют в Т-позе.

Об авторе-составителе



Артём Осиновский

Методист и разработчик детского технопарка «Кванториум» (Новосибирск). Придумывает образовательные программы и сам делает для них цифровые инструменты: платформу смены «Миссия на Луну» для фестиваля «Технопром-2026», сайт с памятками НТО Юниоры для Новосибирской области. Эту книгу собрал для школьников области, чтобы к отбору можно было подготовиться самостоятельно – даже без кружка рядом с

домом.

Нашли ошибку – пишите на osinovsky.aa@kvantorium54.ru (рабочая почта): сфера, версия книги и страница. По другим вопросам и проектам – личная: artem081033@gmail.com.