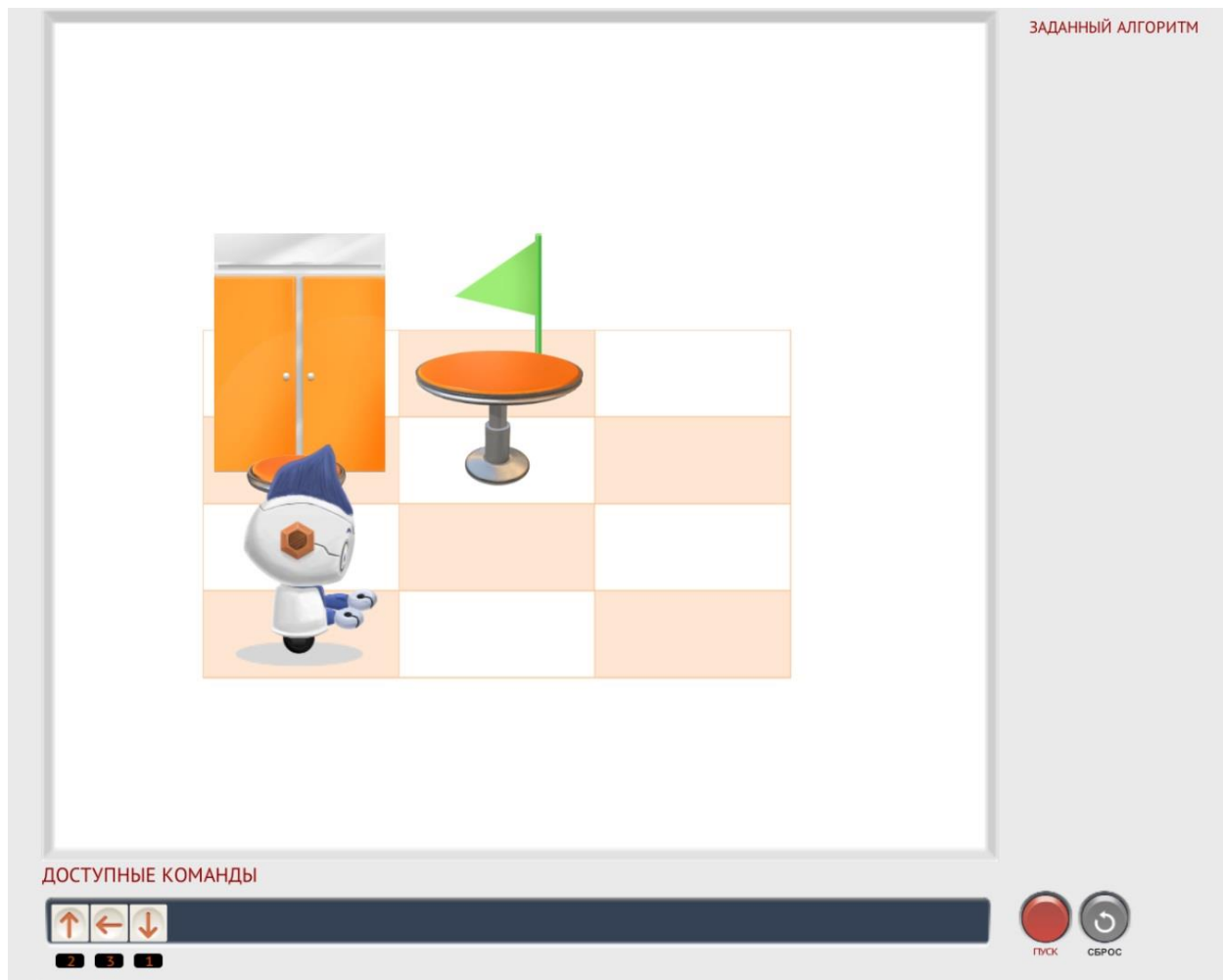


## Общая информация

- Игра в жанре головоломка, основанная на общих принципах алгоритмизации, в мире, населенном роботами
- Игрок выступает в роли "молодого" робота, который находит древний компьютер и решает помочь ему в его миссии - узнать, как изменился мир за то время, что он «спал»
- Управление роботом происходит через построение алгоритма его движения из доступных действий
- Игра состоит из уровней-головоломок, заставляющих игрока придумывать неочевидные решения с использованием ограниченного количества действий
- Рассчитана на игроков от 13 лет, им будет понятна и интересна механика игры и полезна образовательная составляющая



## Структура уровней

Все уровни делятся на три условных этапа - Лёгкий (дом), Средний (улицы Города) и Тяжёлый (Фабрика по производству роботов). Каждый этап строится вокруг одной основной механики: Лёгкие уровни знакомят игрока с особенностями передвижения и взаимодействия с модулями, в уровнях средней сложности главный герой получает способность строить простые циклы и использовать функции, а на тяжелых уровнях игроку предстоит программировать модули и других роботов. Подробнее:

### Лёгкие уровни:

Лёгкие уровни построены таким образом, чтобы научить игрока работать с арсеналом функций и не несут такой большой образовательной роли, но, тем не менее, тоже могут показать игроку много важных деталей. Именно на лёгких уровнях игрок учится продумывать работу своей маленькой программы наперёд, справляться с задачей в условиях нехватки ресурсов и искать альтернативные пути решения, когда обычные не работают.

1. Базовое обучение - игрока учат передвигаться по уровню, вводятся понятия "алгоритм" и "действие"

2-3. Здесь игрок продолжает использовать простые действия и учится планировать время их применения

4. Действие "прыжок". Иногда проще перепрыгнуть препятствие, чем обходить его.

5. ...а иногда нет. На этом уровне игроку на примере прыжков объясняют, что самый очевидный вариант - не всегда лучший, и что нужно заранее учитывать какие функции уместнее применить

6. Знакомство с модулями, отталкивающие модули

7-10. Прочие модули (рычаг, притягивающий и вращающий модули) и действия с ними. Модули станут неотъемлемой частью игры и с этого момента будут появляться практически в каждом уровне.

8-9. Вместе с модулями игрока знакомят с действием "толкать"

### Средние уровни:

Циклы, подпрограммы и условия, которые вводятся на Среднем этапе, практически напрямую повторяют таковые в реальном программировании, поэтому Средний этап самый продолжительный, а все вышеперечисленные механики используется в дальнейшем как основа геймплея. Приёмы и методы решения нетривиальных задач также копируют реальные (Рекурсия, вложенные циклы).

1. Циклы - обучение. На примере самого первого уровня игроку показывают как сильно облегчается задача с использованием циклов

2-4. Продолжаем использовать циклы. Они становятся сложнее, игроку приходится представлять в голове маршрут движения на много ходов вперед

5. Знакомство с "ЕСЛИ" и условиями.

6-8. "ЕСЛИ" в различной комбинации с модулями, циклы с "ЕСЛИ"

9. Введение понятия "Функция".

10. Использование функций в составе более сложных запросов

11-12. Знакомство с понятием "Рекурсия", использование рекурсии для быстрого решения уровней

### Тяжелые уровни:

Тяжёлые уровни, построенные практически целиком на опыте предыдущих этапов, учат игроков не сосредотачивать внимание на одном объекте, а строить сложные схемы и распараллеливать работу программы (для этого в игре используются роботы и конвейеры). Таким образом, игрок

знакомится с принципами многопоточного программирования в упрощенной форме, а также понимает ценность работы в команде.

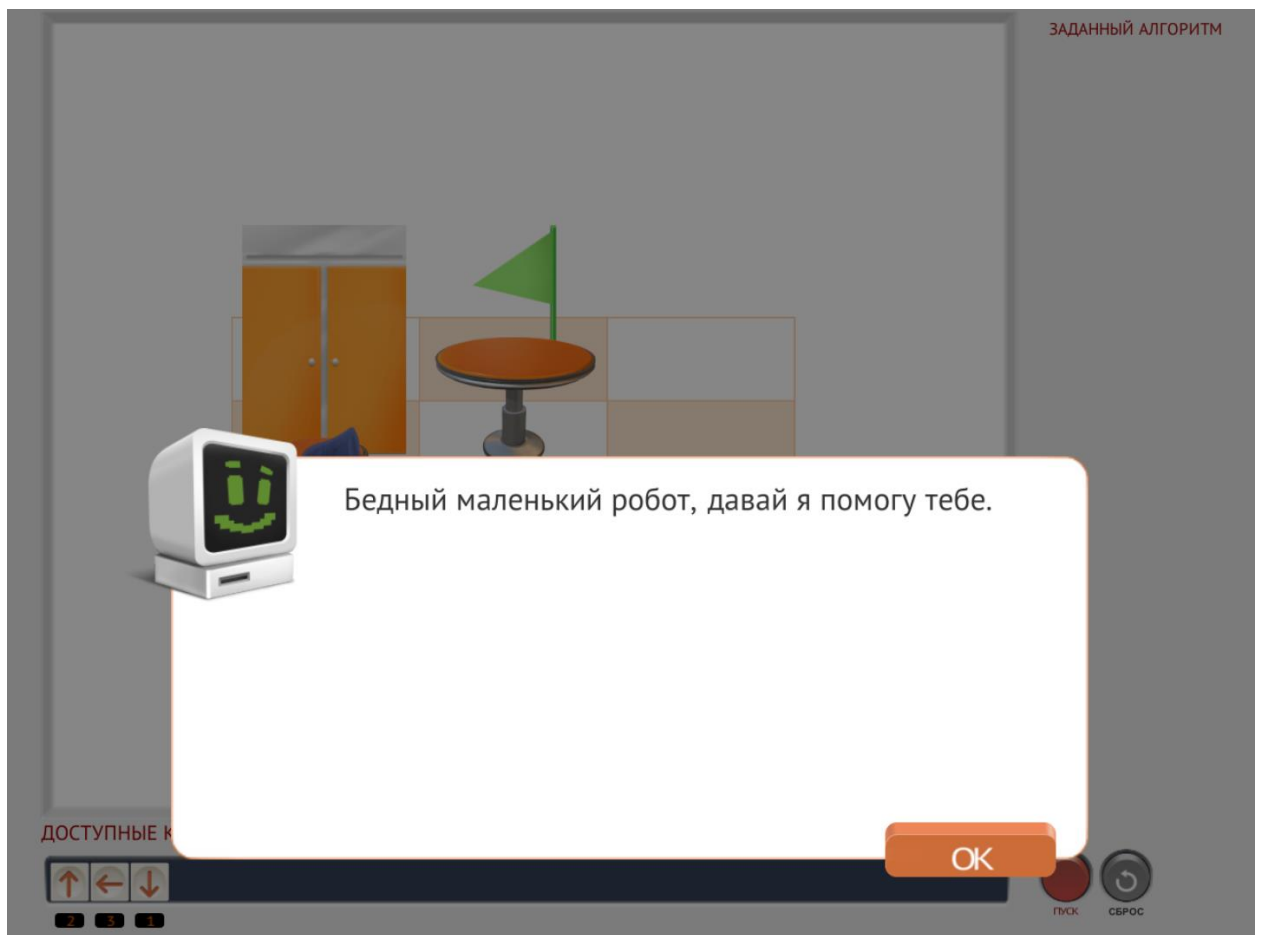
1. В игру вводятся конвейеры, у каждого из которых – свой алгоритм. Игрок задаёт им простейшие алгоритмы движения, чтобы облегчить себе выполнение задачи на уровне и уложиться в заданное количество действий

2-4. На этих уровнях игрок узнает о способах применения конвейеров – перемещение объектов, удаленное управление положением модулей, корректировка курса движения самого робота.

5. Игрок получает возможность управлять другими роботами и создавать для них свои алгоритмы.

6-7. Игрок пробует управлять несколькими роботами и синхронизировать их работу с движениями основного робота

8-10. Игрок использует в различных комбинациях всё, с чем сталкивался в игре с поправкой на использование роботов.



## Типы предметов

Непроходимые блоки (стол, стена, станок на заводе и т.д.) - эти блоки непроходимы, их нельзя перепрыгнуть или протолкнуть, только обойти.

Толкаемые предметы (коробка, стул и т.д.) - эти блоки можно сдвигать с места.

Препятствия (лужа, провал, шипы и т.д.) - на эти блоки можно наступать, в отличие от непроходимых, но контакт с ними закончится смертью героя. Зато их можно перепрыгнуть.

Отталкивающий модуль (к примеру, вентилятор) - отталкивает всё, что может двигаться (игрока, коробки, другие модуль) с ближайшей к себе клетки на следующую, но не дальше. Его тоже можно двигать.

Притягивающий модуль (турбина) - по аналогии с отталкивающим притягивает всё, что находится на второй от него клетке к себе, на ближайшую клетку. Как и все модули может быть толкаем.

Вращающий модуль - поворотный механизм, который берет любой перемещаемый объект с ближайшей к себе клетки и поворачивает вокруг своей (модуля) оси по часовой или против часовой стрелки на 90°. То есть, условно, если представить пространство рядом с модулем в виде перекрестия 3 на 3 клетки с модулем в центре, то модуль возьмет объект "перед собой" и поставит налево или направо (модули бывают и такие и такие, на них есть пометка о том, в какую сторону они вращаются).

Рычаг и ворота - ворота закрыты, нажатие на рычаг их открывает.

Маркер (светофор, к примеру) - что-то, с чем герой может сравнить (используя "ЕСЛИ", например) заданную команду. Может передавать условные ноль и единицу - включен или выключен маркер. Допустим, игрок должен, как хороший робот, идти вперед только по сигналу светофора-маркера. Для этого он строит алгоритм, в котором написано "если светофор горит - иди". Как раз маркер в данном случае выступает таким светофором.

Конвейер - двигающаяся лента, может передвигать любые передвигаемые предметы. Объекты на ней остаются активными, робот может ходить и выполнять команды, а модуль может продолжать толкать робота. Движение конвейера задается игроком в отдельном алгоритме конвейера в стиле "двигаться влево - двигаться вправо".

Робот - такой же робот как главный герой, только двигается не по собственной воле, а по воле игрока. Имеет свой алгоритм, может выполнять то же самое, что и обычный робот.

ЗДАНЫЙ АЛГОРИТМ

ДОСТУПНЫЕ КОМАНДЫ

↑ ← → ↶ ↷

0 0 0 0 0

ГЛУК СЕРОС

The image shows a 5x5 grid environment for a robot navigation task. The robot is located at the bottom-left cell (row 3, column 1). A green flag is located at the top-right cell (row 1, column 5). The environment contains several obstacles: a door at (row 2, column 1), a desk with a monitor at (row 2, column 3), a chair at (row 3, column 2), and a table at (row 3, column 3). The interface includes a control panel at the bottom with five directional buttons (up, left, right, up-left, up-right) and five indicator lights below them. On the right side, there is a section titled "ЗДАНЫЙ АЛГОРИТМ" (Given Algorithm) with a 3x3 grid of directional buttons (up, left, up, up, left, up, up, right).

## Сюжет

*«Робот переезжает в старинный огромный дом, где находит компьютер древних времен – века людей. Компьютер когда-то, ввиду невозможности как-то передвигаться и вообще взаимодействовать с окружающей средой после того, как люди ушли, начал автоматизировать дом и перестраивать оставленные людьми агрегаты под свои нужды. Так появились модули (объясняется диалогом по ходу игры), так же объясняется их наличие в доме – это ранние эксперименты компьютера, и так же объясняется наличие вполне человеческой мебели в доме – здесь реально когда-то жили люди.*

*Со временем все позабыли про заслуги Компьютера, и он оказался погребен под слоем пыли в старом доме, а плоды его трудов стали роботами. Все роботы создаются («рождаются») на Фабрике. Компьютер просит Робота сопроводить его на Фабрику, чтобы посмотреть на то, к чему привели его невинные эксперименты. Робот, который не совсем знает, чем ему ещё заняться, соглашается сопроводить Компьютер на Фабрику и загружает его ИИ в свой мозг.*

*Компьютер по ходу действия учит Робота основам алгоритмизации, потому что это единственное что он знает. Со временем Компьютер понимает, что для решения некоторых задач необходим новый подход, и тогда Робот показывает, что сам может придумывать неожиданные решения.*

*Финальным аккордом в становлении главного героя становится развитие в нём функции программировать также и других роботов, механизмы, которые так же сложны, как и он сам.*

*Добравшись до Фабрики, наши герои видят, что, несмотря на заметное улучшение технологий производства, сам принцип остался практически неизменным. Компьютер читает своим «детям» лекцию о том, что нельзя вечно просто копировать старые проверенные схемы, и что он понял, что должен уступить роль новатора новому поколению в лице Главного героя (что является обращением не только к персонажам игры, но и к игрокам).*

*В Городе роботов наступает бум технологий»*

История подаётся в виде диалога между роботом и искусственным интеллектом компьютера. В диалогах компьютер вспоминает своё прошлое, объясняет, почему модули выглядят именно так, и каким общество роботов было на заре своего существования. В тех моментах, когда история не может ограничиться диалогом (завязка, развязка), возможно использование коротких комиксовых врезок.

## Диалоги

Как уже говорилось, история подается в диалогах. В определенных моменты в нижней части экрана открывается диалоговое окно, в котором игрок может прочитать (или пропустить) информацию о мире и о персонажах.

Обучение механикам игры происходит в виде всплывающих окон со стрелками, которые указывают на элементы, на которые в данный момент стоит обратить внимание. Подсказки, вызываемые игроком после некоторых неудачных попыток по нажатию специальной кнопки, работают похожим образом. Текст подсказок и обучения компактен за счет разнесения на несколько окон, а указание на элементы UI и подсвечивание важных функций игры делает подсказки и обучение интуитивно понятным.

Образовательную функцию в основном несёт механика игры, но также присутствует возможность для некоторых элементов вызвать более подробную справку со ссылкой на реальный мир. К примеру, если кликнуть на слово «алгоритм», то в отдельном окне откроется текст с подробным описанием понятия алгоритм, исторической справкой и фактах применения алгоритмизации в повседневной жизни.



## Образовательная функция

Игра ставит перед собой цель показать игрокам, как работают простейшие операции и выработать у них соответствующий подход к решению алгоритмических задач. Следовательно, вся механика игры должна быть построена по тому же принципу, что и алгоритмические задачи.

Хотя объекты сложно разделить на "обучающие" и "не обучающие", используемые механики можно сгруппировать по их важности для образовательного процесса.

Первая группа - "основные" механики, вокруг которых строится как в целом геймплей, так и образовательный процесс. К ним относятся циклы, функции, условия, программируемые объекты и, отдельным пунктом, алгоритм и его ограничения в количествах действий. Все вышеперечисленные механики используются в течение практически всей игры, большинство задач построено именно на них, а сложность уровней зависит от сложности используемых в алгоритме процедур.

Вторая группа - "второстепенные" механики. Они служат для усложнения уровней и борьбы с однообразием игрового процесса. К ним относятся различные модули (отталкивающие, притягивающие, вращающие, конвейеры) и "необязательные" действия (прыжок, возможность толкать объекты). Механики этой группы не несут сами по себе ключевой роли в игре, но зачастую играют большую роль в конкретных уровнях. Именно они меняют правила игры и заставляют игроков искать альтернативные, непривычные решения, а также играют роль разнообразных начальных условий. Это функционал, с которым игроку приходится работать и мириться, как программисту приходится мириться с необходимыми для работы других людей условиями.

В целом, игра своей основной задачей ставит именно образование, и, хотя наличие «увеселительных» элементов неизбежно, они, тем не менее, вплетены в механику игры таким образом, чтобы не мешать образовательному процессу.

