

УДК 372.4:004.7

DOI: 10.25559/SITITO.14.201804.1000-1011

ПРОПЕДЕВТИКА ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ШКОЛЬНОЙ ИНФОРМАТИКЕ: КОМПЬЮТЕРНАЯ ИГРА «ПОЖАРНЫЕ ТАНКИ»

М.А. Плаксин^{1,2}, А.А. Щелкунов¹¹ Национальный исследовательский университет «Высшая школа экономики», г. Пермь, Россия² Пермский государственный национальный исследовательский университет, г. Пермь, Россия

PROPAEDEUTICS OF PARALLEL COMPUTING IN SCHOOL INFORMATICS: THE COMPUTER GAME «FIREFIGHTING VEHICLES»

Mikhail A. Plaksin^{1,2}, Andrey A. Shchelkunov¹¹ National Research University Higher School of Economics, Perm, Russia² Perm State National Research University, Perm, Russia

© Плаксин М.А., Щелкунов А.А., 2018

Ключевые слова

Информатика; параллельное программирование; параллельные вычисления; параллельные алгоритмы; начальная школа; средняя школа; методика; пропедевтика; ТРИЗформатика, ТРИЗформашка.

Аннотация

Статья содержит методические материалы для включения в школьный курс информатики темы «Параллельные вычисления», а именно описание компьютерных игр «Танковый экипаж», «Рой роботов», «Пожарные танки». Две первые были представлены на предыдущих конференциях, поэтому описаны кратко. В первой игре требуется запрограммировать совместные действия членов танкового экипажа по уничтожению огневых точек противника. Сюжет второй игры – снаряжение космической экспедиции. Для этого группа роботов должна загрузить в космические корабли контейнеры с топливом и продовольствием. В случае объединения усилий нескольких роботов проявляется системный эффект: увеличивается физическая сила роботов и их интеллектуальный уровень. С точки зрения программирования происходит изменение количества исполнителей и их возможностей. В игре «Пожарные танки» группа пожарных машин должна потушить пожар. Эта игра – самая сложная. По сравнению с первой игрой у членов экипажа усложнена система команд. Из «Роя роботов» перенесен системный эффект от объединения усилий нескольких пожарных машин. Добавлены два совершенно новых аспекта: два уровня параллельности (параллельные действия танков и членов экипажа каждого танка) и динамическое изменение окружающей среды (сгорание объектов и распространение пожара в зависимости от направления и силы ветра). В 2018/19 уч.г. эта игра используется в межрегиональных Интернет-конкурсах для школьников и студентов «Параллельное программирование – с пелёнок (2018)!» и «ТРИЗформашка-2019» (межрегиональном конкурсе по информатике, системному анализу и теории решения изобретательских задач (ТРИЗ)).

Об авторах:

Плаксин Михаил Александрович, кандидат физико-математических наук, доцент, кафедра информационных технологий в бизнесе, Национальный исследовательский университет «Высшая школа экономики» (Пермский филиал) (614070, Россия, г. Пермь, ул. Студенческая, д. 38); доцент, кафедра математического обеспечения вычислительных систем, Пермский государственный национальный исследовательский университет (614990, Россия, г. Пермь, ул. Букирева, д. 15), ORCID: <http://orcid.org/0000-0002-6288-8610>, mapl@list.ru

Щелкунов Андрей Александрович, магистрант, Национальный исследовательский университет «Высшая школа экономики» (Пермский филиал) (614070, Россия, г. Пермь, ул. Студенческая, д. 38), ORCID: <http://orcid.org/0000-0002-4286-2632>, would.you.kindly@yandex.ru



Keywords

Informatics; parallel programming; parallel computing; parallel algorithms; elementary school; secondary school; teaching methods; propaedeutics; TRIZformatika, TRIZformashka.

Abstract

The article contains the methodological materials for inclusion of the topic "Parallel Computing" in the school informatics. The computer games "Tank crew", "Swarm of robots", "Firefighting vehicles" are considered. The goal of the first game is to program joint actions of tank crew members. The plot of the second game is the putting on foot the space expedition. For this goal, a group of robots should load containers with fuel and food into spacecrafts. Under uniting efforts of several robots, systemic effect is emerged: their physical and intellectual abilities increase. For programming, it means change in the number of performers and their capabilities. In the third game a group of fire engines should fight fire. The game includes the systemic effect of combining the efforts of fire engines, and two new aspects: two levels of parallelism (parallel actions of vehicles and crew members of each vehicle) and dynamic changes in the environment (combustion of objects and the spread of fire depending on the direction and strength of the wind). In the 2018/19 academic year this game is used in interregional Internet contests for schoolchildren and students "Parallel programming – from the cradle (2018)!" and "TRIZformatashka-2019" (interregional competition in informatics, systems analysis and theory of inventive problem solving (TRIZ)).

Введение

Современный этап развития Computer Science связан с массовым распространением параллелизма вычислений на всех уровнях (многомашинные кластеры, многопроцессорные ЭВМ, многоядерные процессоры). В частности, в [1] обсуждается вопрос об обязательности обучения параллельному программированию программных инженеров и необходимости как можно более раннего развития навыков параллельного мышления. Это делает актуальным включение пропедевтики параллельного программирования в школьный курс информатики.

Данная статья посвящена разработке темы «Параллельные вычисления» в рамках работ над «пермской версией» пропедевтического курса информатики. Идеология «пермской версии» изложена в [2, 3, 4], учебники «пермской версии» печатаются издательством «БИНОМ. Лаборатория знаний» [5, 6]; авторский коллектив: М.А. Плаксин, Н.И. Иванова, О.Л. Русакова; рабочее название курса – «ТРИЗформатика». Часть материалов курса ТРИЗформатики, посвященных тематике параллельных вычислений, была представлена на трех предыдущих Международных научно-практических конференциях «Современные информационные технологии и ИТ-образование» X, XI и XII [7, 8, 9].

Для отработки методических материалов курса ТРИЗформатики используется конкурс «ТРИЗформашка» [10] – ежегодный межрегиональный Интернет-конкурс по информатике, системному анализу и ТРИЗ (теории решения изобретательских задач) для школьников и студентов. В марте 2019 г. конкурс состоится в 19-й раз. Возраст участников – от первого класса до четвертого курса. Среднее количество команд – около 100 (рекордное – 202). География конкурса – от Сахалина и Владивостока до Риги и Калининграда. Сайт конкурса www.trizformashka.ru.

Данная статья посвящена описанию нового, только что разработанного компонента курса ТРИЗформатики – компьютерной игры «Пожарные танки».

Комплект компьютерных игр для начального знакомства с параллельным программированием: «Танковый экипаж», «Рой роботов», «Пожарные танки»

Игровая форма обучения является одной из самых эффективных (наряду с робототехникой и системами симуляции). Различные аспекты применения игровых методик для развития алгоритмического мышления и обучения программированию рассматриваются в [11-17]. Первой отечественной компьютерной игрой, нацеленной на знакомство учащихся средней школы с азами параллельного программирования, была игра «Директор строительства», входившая в курс «Алгоритмики» в далеком 1996 г. [18]. К сожалению, тогда она оказалась совершенно невостребованной, опередив свое время лет на двадцать.

В рамках подготовки конкурса «ТРИЗформашка» в период с 2014 по 2017 гг. для начального знакомства с базовыми понятиями параллельного программирования были разработаны две компьютерные игры: «Танковый экипаж» [19] и «Рой роботов» [9]. В 2017/18 уч.г. к ним была добавлена еще одна: «Пожарные танки». Автором компьютерной реализации «Пожарных танков» является студент Пермского филиала Национального исследовательского университета Высшая школа экономики, один из авторов данной статьи А.А. Щелкунов.

Три названных игры представляют собой комплект, в рамках которого учащимся предлагается составлять алгоритмы совместной деятельности нескольких акторов. При этом при переходе к каждой следующей игре процесс взаимодействия становится все более сложным.

В «Танковом экипаже» перед игроком стоит задача уничтожить огнем танкового орудия вражеские огневые точки. Для этого игрок должен составить алгоритм действий трех членов экипажа: водителя, наводчика и заряжающего. Понятно, что действия всех троих должны быть согласованы по времени и ре-



зультату (изменение направления движения танка, которым руководит водитель, должно быть согласовано с поворотом орудия, за который отвечает наводчик; заряжающий должен успеть зарядить оружие к моменту выстрела, либо наводчику и водителю придется сделать паузу, чтобы дождаться момента заряжа-

ния). Соревновательный момент связан с тем, что нужно не просто уничтожить противника, но сделать это за минимальное время (за минимальное количество шагов в алгоритме). Пример игры «Танковый экипаж» приведен на рис.1.

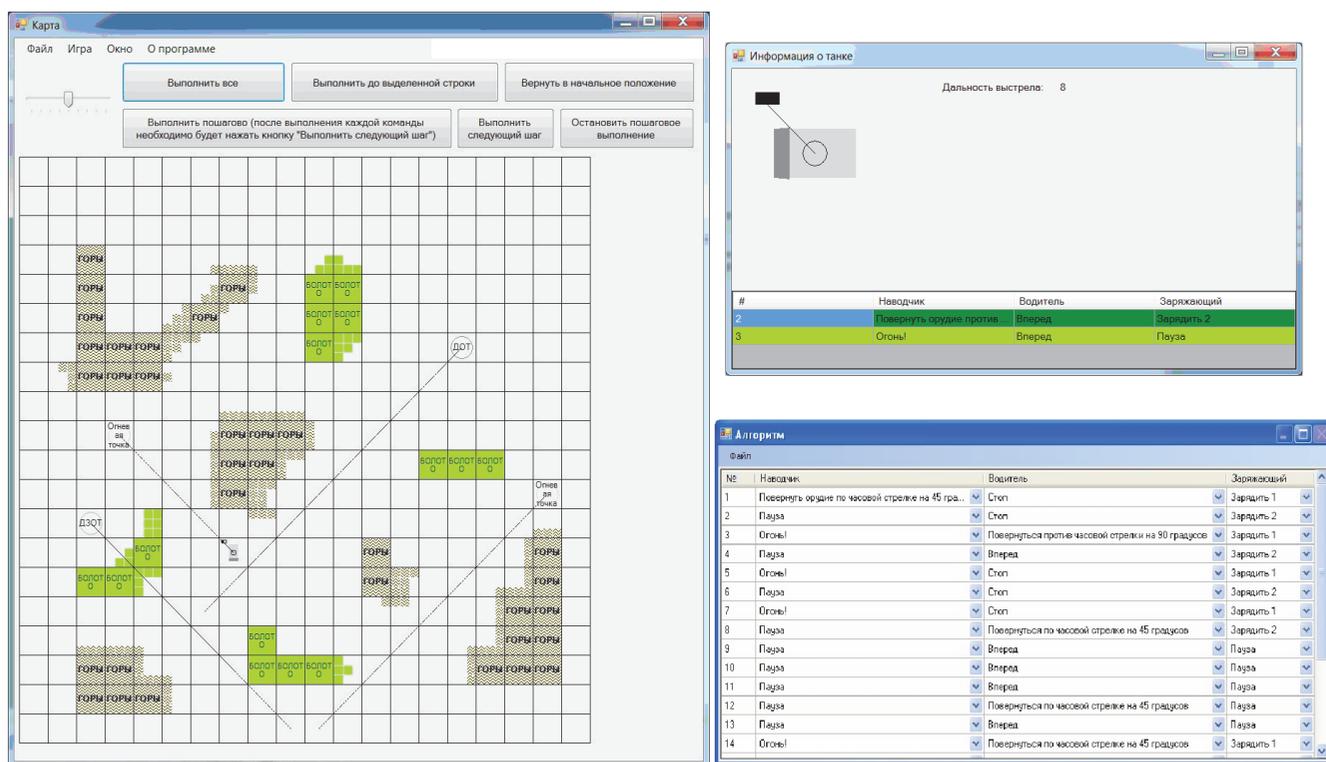


Рис. 1. Пример игры «Танковый экипаж»

Fig. 1. An example of the game «Tank crew»

В «Рое роботов» группа роботов занимается снаряжением космической экспедиции, которая должна стартовать с Марса для исследования Юпитера. Экспедиция состоит из нескольких космических кораблей, в каждый из которых должно быть загружено требуемое количество контейнеров с продовольствием и топливом. Контейнеры рассредоточены по нескольким складам (на каждом – свой запас). Выполнить доставку контейнеров со складов и загрузку их на космические корабли должны транспортные роботы. Роботы обладают определенной грузоподъемностью, скоростью передвижения, способностью к прыжкам через расщелины, сложным маневрам и пр. Как и следует из названия, игра «Рой роботов» должна продемонстрировать обучающемуся системный эффект, возникающий при объединении действия отдельных роботов. Делается это следующим образом. Отдельные роботы могут объединяться в «сцепки». При этом возможности таких «сцепок» возрастают по сравнению с возможностями создавших их роботов, причем возрастают нелинейно. Так одиночный робот способен переносить один контейнер, движется со скоростью одна клетка за ход по горизонтали или вертикали, не способен перепрыгивать через расщелины. Сцепка из двух роботов может нести уже три контейнера, скорость движения возрастает до трех клеток за ход, появляется способность прыжка через расщелины. У сцепки из трех роботов

грузоподъемность возрастает до пяти контейнеров, скорость – до пяти клеток за ход. И, сверх того, появляется способность к маневрированию (робот не обязан двигаться по прямой, а может изменить направление движения в рамках одного хода). Наконец, сцепка из четырех роботов способна перенести сразу восемь контейнеров, передвинуться за один ход на восемь клеток, совершить уже не один, а два маневра и (новое качество) получает способность двигаться по диагонали. При разделении сцепки синергетический эффект пропадает. Способности получившихся после разделения акторов соответствуют их уровню.

Отметим еще одно новшество данной игры с точки зрения построения алгоритмов. Речь идет об изменении количества исполнителей. После сцепки двух роботов они исчезают из игры как самостоятельные сущности. С точки зрения алгоритма они больше не существуют. Зато появляется новый актер – робот-сцепка, обладающий своим набором возможностей, отличным от наборов возможностей создавших его более простых роботов. После разделения сцепки он перестает существовать, зато появляются новые актеры – те роботы, на которые сцепка разделилась. Это явление – изменение количества и качества исполнителей – совершенно невозможно при обучении традиционному последовательному программированию.

Пример игры «Рой роботов» приведен на рис.2.



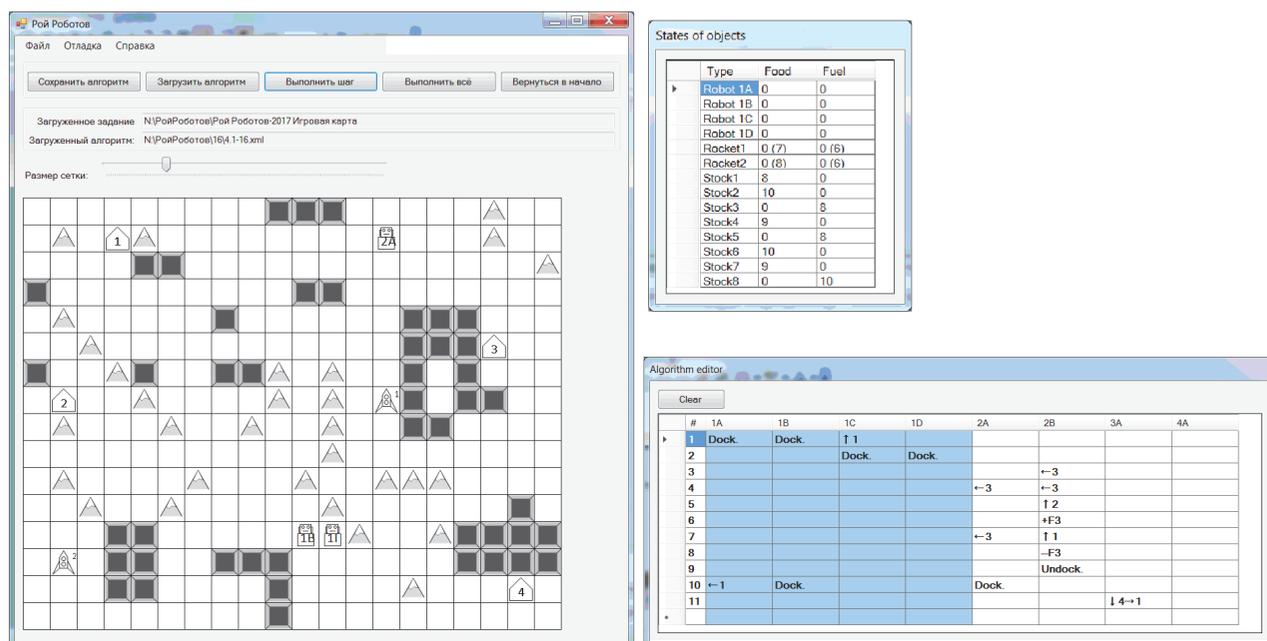


Рис. 2. Пример игры «Рой роботов»

Fig. 2. An example of the game «Roy of robots»

В игре «Пожарные танки» группа пожарных машин должна потушить пожар. Каждая машина управляется экипажем из трех человек и гасит огонь, ведя стрельбу из водяной пушки. (Дискретность алгоритма потребовала отказаться от непрерывности водяной струи и перейти к отдельным водяным выстрелам). На игровом поле может быть расположен лес, озера, дома. Часть деревьев и домов охвачена огнем.

В игре сохранены важные характеристики двух предшествующих игр. Каждый пожарный танк управляется экипажем из трех человек. В этом отношении пожарный танк повторяет игру «Танковый экипаж». Игрок должен запрограммировать согласованные действия водителя, наводчика и заряжающего. Разница только в более сложных системах команд двух последних членов экипажа (наводчика и заряжающего). Из игры «Рой роботов» перенесена идея системного эффекта, возникающего при совместном действии нескольких исполнителей. Для пожарных танков это выражается следующим образом. Пожарный танк должен затушить горящие объекты, производя по ним «водяные выстрелы». При этом для тушения разных объектов требуется разное количество водяных выстрелов. Так вот, системный эффект проявляется в случае, когда несколько машин тушат один и тот же горящий объект одновременно: если для тушения пожара одним танком потребуются три водяных выстрела, то два танка затушат этот пожар одним залпом (то есть только двумя водяными выстрелами, но произведенными одновременно).

Кроме того, в игру «Пожарные танки» добавлены еще два важных момента. Во-первых, параллельность здесь оказывается «двухуровневой». Учащийся, программирующий действия пожарных танков, должен сначала разделить всю необходимую работу между танками (на «верхнем уровне»). А затем обеспечить выполнение каждым танком его задания путем согласованных действий членов экипажа пожарного танка (на «нижнем

уровне»). «Нижний уровень» параллельности «Пожарных танков» соответствует уровню параллельности «Танкового экипажа». «Верхний уровень» – принципиально новая добавка.

Во-вторых, в «Пожарных танках» впервые вводится в рассмотрение изменчивость среды, в которой действуют акторы. И в «Танковом экипаже», и в «Рое роботов» среда, в которой действовали акторы, оставалась неизменной, за исключением тех изменений, которые были внесены в нее самими акторами по ходу исполнения алгоритма. В «Танковом экипаже» изменения сводились к уничтожению вражеских огневых точек, что и было целью игры. В «Рое роботов» изменения заключались в уменьшении количества контейнеров на складах и увеличении количества контейнеров на космических кораблях. Это уменьшение/увеличение было необходимо учесть при составлении алгоритма (робот не может забрать контейнер из склада, если на предыдущих шагах все контейнеры из этого склада были уже выгружены). Но сами изменения были следствием исполнения алгоритма и, значит, были целиком подконтрольны игроку, этот алгоритм составляющему. В «Пожарном танке» картина существенно отличается. Во-первых, цель игры – тушение пожара. Представленные в игре объекты – горят. Горение – процесс конечный. Через определенное количество шагов он закончится. То есть объект – сгорит дотла. Если к этому времени он не будет потушен, восстановить его уже не удастся. В двух предыдущих играх порядок обработки того или иного объекта влиял только на скорость алгоритма, но не результат работы. В «Танковом экипаже» огневые точки можно было уничтожить в любом порядке. Важно было уничтожить их все. И сделать это за возможно меньшее число шагов. Аналогично, в «Рое роботов» контейнеры могли быть перенесены из любого склада на любой космический корабль в любом порядке. Важно было только сделать это как можно быстрее. В «Пожарных танках» скорость горения объекта – независимый параметр. Если не успеть пога-



сечь пожар, то объект сгорит, и восстановить его будет невозможно. Значит, игроку предстоит так определить порядок действий своих пожарных машин, чтобы минимизировать суммарную стоимость всех сгоревших объектов.

Но динамичность среды в «Пожарных танках» не ограничивается возможностью утери сгоревших объектов. Пожар в игре может распространяться! То есть от уже горящих объектов со временем (через несколько ходов) загорятся примыкающие к ним. И тогда тушить придется уже не только «первичные» пожары, но и «наведенные». Поэтому перед игроком встает задача затушить в первую очередь те объекты, которые могут вызвать наибольшее распространение огня. Например, может оказаться, что дешевле пожертвовать объектом, не имеющим соседей и не представляющим опасности с точки зрения распространения пожара.

Дополнительную динамичность игре придает тот факт, что направление и скорость распространения огня зависит от направления и скорости ветра. По ходу игры ветер меняет свое направление и силу. Значит, определять порядок тушения игрок должен не для нынешнего состояния дел, а с учетом его изменения в будущем. Скорость выполнения алгоритма должна быть согласована со скоростью изменения окружающей среды.

Изменение направления и силы ветра – фактор, который очень существенно усложняет игру. Правда, на эти изменения пришлось наложить ограничения, вызванные методическими соображениями.

Во всех трех играх задача игрока – составить алгоритм совместной деятельности нескольких акторов. Во всех трех играх порядок действий каждого актора задается последовательным алгоритмом. Действия всех акторов считаются согласованными по времени (считается, что все действия всех акторов занимают одно и то же время). Это позволяет представить одновременные действия всех акторов как «линейку команд», а параллельный алгоритм для группы акторов как последовательность таких линейек.

Ни для одного актора не предусмотрены ни развилки, ни циклы. Переход от линейных алгоритмов к ветвящимся и циклическим грозит таким резким возрастанием сложности, что, скорее всего, его придется исключить из пропедевтического курса. Во всяком случае, по состоянию на сегодня авторы методики не нашли приемлемого по сложности включения ветвлений и циклов в задачи на построение параллельных алгоритмов. Отметим, что в зданиях на исполнение параллельных алгоритмов ветвления и циклы воспринимаются гораздо легче. Достаточно посмотреть в статье «Рой роботов» в материалах прошлогодней конференции [9] задания про рой роботов-спасателей, про стаю клеточных автоматов, создающих узор, про балансировку нагрузки в вычислительной сети с помощью многоагентной системы. Все они имеют циклический характер и включают в себя проверки и выбор способа действия в зависимости от сложившейся ситуации.

Построить линейный алгоритм учащийся сможет только в том случае, если будет заранее знать направление и силу ветра в любой момент в будущем. Поэтому последовательность изменения направления и силы ветра в игре задается заранее. Указывается, в течении какого количества шагов ветер будет дуть в том или ином направлении, и какова будет его сила. При этом пришлось решать дополнительную проблему, связанную с тем, что количество шагов в алгоритме заранее неизвестно. И направление, и сила ветра должны быть заранее указаны для любого, заранее неизвестного количества шагов. Сделано это просто – путем зацикливания указаний. Цепочка указаний направления и силы ветра имеет конечную длину. Если по ходу выполнения

алгоритма эта длина будет превышена, цепочка указаний возвращается на начало.

Отметим, что изменение направления и силы ветра может быть отключено. Для этого достаточно в цепочке указаний оставить единственный элемент: штиль. Таким образом, в конкретном случае от этих сложностей игры можно избавиться. Но отключить сгорание объектов и распространение огня (даже в случае штиля) невозможно.

Компьютерная игра «Пожарные танки»

«Пожарный танки» – это компьютерная игра, предназначенная для развития параллельного алгоритмического мышления. Игрок должен разработать параллельный алгоритм для управления группой пожарных танков в ходе тушения пожара. Цель игры – разработать наиболее эффективный алгоритм и потушить пожар с наименьшими потерями.

Алгоритм выполняется в пошаговом режиме.

Игровое поле представляет собой карту, разделенную на клетки, на которых располагаются игровые объекты.

Игровыми объектами являются пожарные танки, деревья, скалы, озера, дома.

Деревья и дома представляют собой горючий материал. Часть их горит в начале игры. Часть может загореться позже. Если их не тушить, они через несколько ходов сгорят совсем.

Танки будут тушить пожар, заливая его водой из водяной пушки. Поскольку алгоритм имеет дискретный характер, в игре пришлось отказаться от непрерывной водяной струи в пользу дискретных водяных выстрелов. Собственно, после этого пожарная машина и превратилась в «пожарный танк».

Скалы и озера являются препятствием на пути танка. Проехать через них танк не может. Кроме того, через озеро танк может стрелять «прямой наводкой» (при опущенном орудии), а через скалы – только навесом (при поднятом орудии). И наконец, озера могут служить источником для пополнения запаса воды в танке. Подробнее обо всем этом будет рассказано дальше.

Пример игрового поля приведен на рис.3.



Рис. 3. Пример игрового поля игры «Пожарные танки»
Fig. 3. An example of the playing area of the game "Fire tanks"



В заголовке окна карты стоит список, в котором перечислены тройки «направление, сила ветра и продолжительность ветра».

В указанном направлении огонь будет переходить на рядом стоящие деревья и дома. Скорость распространения огня зависит от силы ветра. Деревья загораются через 4 хода при силе ветра 1, через 3 хода при силе ветра 2, через 2 хода при силе ветра 3. Дома загораются через 8 ходов при силе ветра 1, через 7 ходов при силе ветра 2, через 6 ходов при силе ветра 3.

Изменение направления ветра и его силу может очень существенно влиять на сложность игры. Постоянный ветер минимальной силы сделает игру гораздо более простой, чем ураган переменных направлений.

Деревья сгорают за 10 ходов. Дома сгорают за 15 ходов. В случае, если дерево или дом были потушены, а потом загорелись опять, они продолжит гореть с тем количеством «жизней», с которыми были потушены.

Продолжительность ветра – это количество шагов, в течение которых ветер дует в указанном направлении. После этого из списка берется следующая тройка и ветер меняет направление и силу в соответствии с новыми указаниями. И так далее, пока не дойдем до конца списка. После этого происходит возврат на начало списка. Таким образом, даже короткий список может управлять изменением ветра бесконечно.

В каждом танке находится экипаж из трех исполнителей:

- водителя, отвечающего за передвижение танка,
- заряжающего, отвечающего за подготовку к выстрелу,
- наводчика, отвечающего управление водяной пушкой и производство водяного выстрела.

Каждый исполнитель имеет свою систему команд. За один шаг алгоритма каждый исполнитель выполняет только одну команду. Для выполнения большинства команд необходимо, что-

бы были соблюдены определенные условия, иначе выполнение алгоритма приведет к ошибке.

Иногда результат действий танка определяется действиями нескольких членов экипажа. Например, на одном и том же шаге алгоритма водитель перемещает танк, а наводчик дает выстрел из пушки. В реальной жизни действия разных членов экипажа будут выполняться одновременно. Дискретный характер игрового алгоритма требует некоторых уточнений. В приведенном примере точка попадания выстрела будет зависеть от того, произведен ли выстрел «в начале шага алгоритма» (до того, как танк изменил позицию) или «в конце шага» (уже из нового положения танка). Для подобных случаев принято следующее соглашение. Считается, что сначала выполняется передвижение танка (команда водителя), потом – зарядание пушки (команда заряжающего) и последним – управление орудием (команда наводчика). Это значит, что в нашем примере выстрел произойдет уже из нового положения танка.

Движение танка регламентируется следующими правилами.

Танк может двигаться по горизонтали, вертикали или диагонали. Угол поворота может быть 45 или 90 градусов. За один ход движение возможно на одну клетку вперед, назад или под углом 45 градусов (вправо вперед, влево вперед, вправо назад, влево назад). Для поворота на 90 градусов танк должен потратить отдельный ход.

Считается, что водяная пушка жестко закреплена на танке. Поэтому поворот танка вызывает соответствующее изменение направления орудия. Направление орудия может быть изменено независимо от направления движения танка. Но этим занимается уже не водитель, а наводчик.

Запас хода танка считается неограниченным (какую бы длину не имел алгоритм, топлива для его выполнения хватит!).

Система команд водителя танка:

Команда	Действие	Условие
Вперед	Танк переезжает на 1 клетку вперед	Чтобы выполнить любую из этих команд, необходимо, что клетка, на которой окажется танк после ее выполнения, была свободной, или не находилась за пределами карты.
Назад	Танк переезжает на 1 клетку назад	
Повернуть 45 по часам	Танк поворачивается на 45 градусов по часовой стрелке (орудие также поворачивается)	
Повернуть 45 против часов	Танк поворачивается на 45 градусов против часовой стрелки (орудие также поворачивается)	
Повернуть 90 по часам	Танк поворачивается на 90 градусов по часовой стрелке (орудие также поворачивается)	
Повернуть 90 против часов	Танк поворачивается на 90 градусов против часовой стрелки (орудие также поворачивается)	
Вперед 45 по часам	Танк переезжает на клетку справа перед собой (по диагонали) и поворачивается на 45 градусов по часовой стрелке (орудие также поворачивается)	
Вперед 45 против часов	Танк переезжает на клетку слева перед собой (по диагонали) и поворачивается на 45 градусов против часовой стрелки (орудие также поворачивается)	
Назад 45 по часам	Танк переезжает на клетку слева позади себя (по диагонали) и поворачивается на 45 градусов по часовой стрелке (орудие также поворачивается)	
Назад 45 против часов	Танк переезжает на клетку справа позади себя (по диагонали) и поворачивается на 45 градусов против часовой стрелки (орудие также поворачивается)	
Бездействие	Танк остается на месте	

Зарядание водяного орудия регламентируется следующими правилами.

Перед выстрелом орудие должно быть заряжено.

Дальность выстрела определяется давлением воды и препятствиями, которые могут встретиться на пути водяной струи. Увеличение давления на единицу приводит к увеличению даль-



ности выстрела на одну клетку. Максимальное расстояние, на которое может быть произведен выстрел, равно давлению воды. Существует ограничение на максимальное возможное давление воды (по умолчанию оно равно пяти).

Танк везет с собой определенный запас воды. Каждый выстрел приводит к уменьшению этого запаса на единицу. Запас

воды можно пополнить. Для этого танк должен остановиться около озера (слева, справа, сверху или снизу). Угол поворота танка роли не играет. Для пополнения запаса воды Заряжающий должен отдать специальную команду. Существует ограничение на максимальный запас воды, который может принять танк (по умолчанию оно равно восьми).

Система команд Заряжающего:

Команда	Действие	Условие
Давление +1	Увеличивает давление воды на 1 (увеличивает дальность выстрела на 1 клетку)	Необходимо, чтобы давление воды не превышало максимально допустимое значение давления воды (5) при выполнении команды
Давление +2	Увеличивает давление воды на 2 (увеличивает дальность выстрела на 2 клетки)	
Пополнить запас	Пополняет запасы воды на 1	Необходимо, чтобы танк находился рядом с озером (т.е. на соседней клетке сверху, слева, снизу или справа) под любым углом поворота, и чтобы запас воды не превышал максимальное количество (8) при выполнении команды
Зарядить	Выполняет подготовку орудия к выстрелу	Необходимо, чтобы орудие было опущено и не готово к выстрелу при выполнении команды
Бездействие	Ничего не делает	

Управление орудием регламентируется следующими правилами.

Наводчик может поворачивать орудие, поднимать и опустить его и выполнять выстрел. Поворот орудия выполняется для наведения на цель. Подъем орудия необходим, чтобы при выстреле вода могла перелететь через препятствия (деревья, дома и другие танки).

Дальность выстрела зависит только от давления и не зависит от подъема орудия (это не соответствует реальности, но учет угла наклона при определении дальности выстрела чрезвычайно усложняет игру).

Подъем орудия обеспечивает только одну возможность:

при выстреле вода перелетает через препятствия (скалы, деревья, дома, другие танки).

Если орудие опущено (стрельба «прямой наводкой»), то вода попадет в ближайшее препятствие, находящееся в направлении выстрела на расстоянии не большем дальности выстрела.

Если орудие поднято (стрельба навесом), то вода падает в точности на ту клетку, которая определяется дальностью выстрела.

После выстрела запасы воды уменьшаются на единицу, давление воды сбрасывается до единицы, пушка перестанет быть готова к выстрелу.

Непопадание не считается ошибкой.

Система команд Наводчика:

Команда	Действие	Условие
Повернуть 45 по часам	Поворачивает пушку на 45 градусов по часовой стрелке	
Повернуть 45 против часов	Поворачивает пушку на 45 градусов против часовой стрелки	
Повернуть 90 по часам	Поворачивает пушку на 90 градусов по часовой стрелке	
Повернуть 90 против часов	Поворачивает пушку на 90 градусов против часовой стрелки	
Поднять	Поднимает орудие	Необходимо, чтобы пушка была опущена
Опустить	Опускает орудие	Необходимо, чтобы пушка была поднята
Выстрелить	Выполняет выстрел	Необходимо, чтобы орудие было готово к выстрелу, и запасы воды были больше нуля
Бездействие	Ничего не делает	

«Тушительная мощь» водяных выстрелов определяется следующими правилами.

Деревья всегда тушатся за одно попадание.

Дома тушатся за три попадания, идущих друг за другом, либо за одно попадание залпом одновременно из двух или более орудий. Таким образом, здесь проявляется синергетический эффект. Три последовательные попадания из одного танка равны одному залпу из двух танков. (Поскольку целей, которые требовали бы большее число попаданий, в игре нет, направлять на тушение одного лома больше двух танков сразу нет смысла.)

Пример окон, содержащих алгоритмы для экипажей трех танков, приведен на рис.4.

Смысл значков, стоящих в нижней части окна под описани-

ем алгоритма, следующий:

Значок с каплей означает количество запасов воды в танке. Первое число – текущий запас воды (т.е. количество выстрелов, которые еще можно сделать), второе число – максимальный допустимый запас воды. Запас воды пополняется из озера. Попытка переполнить запас (набрать воды больше максимально возможного) считается ошибкой алгоритма.

Второй значок («пульверизатор») обозначает давление воды. Первое число – текущее давление воды (на какую дальность возможен выстрел, по умолчанию оно равно единице), второе число – максимальное давление воды (то есть максимальная дальность выстрела). Попытка превысить максимально возможное давление считается ошибкой алгоритма. После выстрела давление сбрасывается в единицу.



№	Водитель	Заряжающий	Наводчик	П.
1	Вперед	Зарядить	45° пр. ч.с.	1
2	Вперед	Бездействие	Выстрелить	1
3	Вперед 45° пр. ч.с.	Зарядить	45° пр. ч.с.	2
4	Бездействие	Бездействие	Выстрелить	2
5	90° пр. ч.с.	Зарядить	45° пр. ч.с.	3
6	Вперед	Давление +2	Поднять	3
7	90° пр. ч.с.	Давление +1	Выстрелить	3
8	45° пр. ч.с.	Бездействие	Опустить	4
9	Бездействие	Зарядить	Выстрелить	4
10	Назад	Давление +2	90° пр. ч.с.	5
11	Назад	Пополнить запас	Бездействие	5
12	Бездействие	Зарядить	Выстрелить	5
13	Назад	Пополнить запас	Бездействие	5
14	Назад 45° пр. ч.с.	Пополнить запас	Бездействие	5
15	Бездействие	Давление +2	Выстрелить	5
16	45° пр. ч.с.	Пополнить запас	45° пр. ч.с.	5
17	Назад	Зарядить	Выстрелить	5
18	Бездействие	Зарядить	Выстрелить	5

№	Водитель	Заряжающий	Наводчик	П.
1	Вперед 45° пр. ч.с.	Зарядить	90° пр. ч.с.	1
2	45° пр. ч.с.	Давление +2	Бездействие	1
3	Вперед	Давление +2	Выстрелить	1
4	Вперед	Зарядить	45° пр. ч.с.	2
5	Вперед	Пополнить запас	Выстрелить	2
6	90° пр. ч.с.	Зарядить	Выстрелить	3
7	45° пр. ч.с.	Зарядить	Выстрелить	4
8	45° пр. ч.с.	Зарядить	Поднять	5
9	Вперед	Давление +2	Бездействие	5
10	Вперед	Пополнить запас	Выстрелить	5
11	Вперед	Давление +2	Опустить	6
12	Бездействие	Зарядить	Выстрелить	6
13	Назад	Пополнить запас	Бездействие	7
14	Назад	Зарядить	Поднять	7
15	Вперед	Давление +1	Выстрелить	7

№	Водитель	Заряжающий	Наводчик	П.
1	Вперед	Зарядить	Поднять	1
2	Вперед 45° пр. ч.с.	Давление +2	45° пр. ч.с.	1
3	Бездействие	Давление +1	Выстрелить	1
4	Назад	Зарядить	Опустить	2
5	45° пр. ч.с.	Зарядить	45° пр. ч.с.	2
6	Вперед 45° пр. ч.с.	Давление +2	Выстрелить	2
7	90° пр. ч.с.	Пополнить запас	Бездействие	3
8	Вперед 45° пр. ч.с.	Пополнить запас	Бездействие	3
9	Вперед	Бездействие	Бездействие	3
10	Бездействие	Зарядить	Выстрелить	3
11	45° пр. ч.с.	Зарядить	Бездействие	4
12	Бездействие	Бездействие	Выстрелить	4
13	45° пр. ч.с.	Зарядить	Поднять	5
14	Вперед	Зарядить	90° пр. ч.с.	5
15	Бездействие	Бездействие	Выстрелить	5

Рис. 4. Пример алгоритмов экипажей трех пожарных танков
Fig. 4. An example of the algorithms of the crews of three fire tanks

Третий значок (танк) показывает опущена пушка или поднята. Если опущена, то на значке у танка пушка примет горизонтальное положение, а если поднята, то наклонное.

Четвертый значок «Круг» показывает, заряжена ли пушка. Если заряжена, то кружок будет покрашенным, а если нет, то будет пустым (просто окружность).

Еще четыре значка предназначены для упрощения редактирования и отладки алгоритма.

Плюс позволяет добавить строку с действиями экипажа в любое место алгоритма. При нажатии на плюс открывается окошко, где можно указать номер строки, куда вставить действия экипажа, и указать сами действия (для водителя, наводчика и заряжающего). По умолчанию строка добавляется в конец алгоритма.

Линия с крестиком удаляет выделенную строку алгоритма. Корзина очищает весь алгоритм данного танка.

Знак «Замок» (если он закрыт) блокирует выполнение алгоритма данного танка. То есть даже если алгоритм для танка написан, танк ничего делать не будет. Это сделано для того, чтобы программист мог не отвлекаться на неинтересные ему в данный момент танки, а сосредоточиться на том танке, алгоритм для которого он сейчас составляет.

В таблице, описывающей действия членов экипажа одного танка, есть еще одна колонка, нами до сих пор не упомянутая: самая правая колонка, помеченная буквой «П». Эта колонка содержит ссылки на новый элемент управления, появившийся в «Пожарных танках»: план совместных действий нескольких танков.

Как уже было сказано, игра «Пожарные танки» требует согласования действий исполнителей на двух уровнях: на уровне танков и на уровне членов экипажа. Для игры «Танковый экипаж» такой проблемы не возникало. В той игре ставились задачи, которые можно и нужно было осмысливать на уровне действий отдельных членов экипажа. В «Пожарных танках» картина изменилась качественно. Здесь игроку предлагается решать задачи, осмыслить которые на уровне действий отдельных членов экипажа невозможно. Происходит переход количества в качество. Необходимо сначала осмыслить задачу на «верх-

нем» уровне – уровне танков, а потом уже спускаться на «нижний» уровень – уровень отдельных членов экипажа.

При этом возник вопрос о том, как обеспечить соответствие двух уровней. Для уровня экипажа существует формальный язык, объединяющий системы команд всех членов экипажа. Алгоритмы «нижнего уровня» описываются на этом языке. Вопрос: на каком языке описывать алгоритмы «верхнего уровня», уровня танков? Вариантов ответа было три:

1. Разработать для этого уровня свой формальный язык.
2. Ввести в «язык экипажей» механизм подпрограмм. В качестве команд «языка танков» использовать процедуры, написанные на «языке экипажей».
3. Использовать в качестве языка «верхнего уровня» неформальный язык. Связь между командами верхнего и нижнего уровней оставить неформальной.

Эксперименты показали, что два первых подхода приводят к резкому усложнению игры (которая и так представляется весьма сложной). В первом случае возникает необходимость формального установления эквивалентности команд «танкового языка» соответствующим группам команд «экипажного языка». Во втором линейную запись алгоритма экипажей, отображающую линейки одновременно исполняемых команд всех членов экипажей всех танков приходилось заменять вызовами подпрограмм, описанных где-то в другом месте. В результате терялась единовременность действий всех членов всех экипажей. Что для игры, призванной ознакомить учащихся с азами параллельности, представляется гораздо более важным, нежели формальное следование некоторым программистским догмам.

Поэтому был выбран третий вариант. В игру был введен новый элемент управления: план совместных действий всех танков. Этот план записывается в отдельную таблицу в неформальном виде (на русском языке). Команды, его составляющие, представляют собой тексты вида «Тушить дерево на клетке с координатами такими-то», «Тушить дом такой-то совместно с танком таким-то», «Перейти на клетку с координатами такими-то для того, чтобы набрать воды» и т.д. Все действия каждого танка перенумерованы.

После создания плана действий на уровне танков, начина-



ется его реализация на уровне экипажей. Для этого каждый пункт плана действий каждого из танков раскрывается через действия членов экипажа этого танка. Для связи между двумя уровнями алгоритма в правую колонку в таблице команд экипажа записывается номер пункта плана, к которому относится данная команда.

Пример плана совместных действий нескольких танков приведен на рис.5.

План			
№	Красный танк (№1)	Синий танк (№2)	Желтый танк (№3)
1	Тушим дерево К-12	Тушим дерево F-12	Тушим дерево J-2
2	Тушим дерево J-9	Тушим дерево I-9	Тушим дерево H-6
3	Тушим дерево J-7	Тушим дерево I-7	Тушим дом С-5 вместе с синим танком
4	Тушим дерево I-10	Тушим дерево H-7	Тушим дом С-5 вместе с синим танком
5	Тушим дом N-8	Тушим дом С-5 вместе с желтым танком	Тушим дом В-6 вместе с синим танком
6		Тушим дом С-5 вместе с желтым танком	
7		Тушим дом В-6 вместе с желтым танком	

Рис. 5. Пример плана совместных действий трех танков

Fig. 5. An example of a plan of joint action of three tanks

Такая «неформальная двухуровневость» алгоритма чревата несоответствием между его «верхним» и «нижним» уровнями. Никто не может гарантировать, что алгоритм нижнего уровня действительно реализует тот пункт плана, который записан на верхнем уровне. Однако для линейных алгоритмов небольшого размера, имеющих учебный характер (то есть не предназначенных для долговременного хранения, многократного исполнения, доработки и переработки) этот недостаток представляется приемлемым. Во всяком случае, более приемлемым, нежели усложнение игры, которое требовалось для двух альтернативных подходов.

Если объединить вместе на одном экране все вышесказанное (игровое поле, алгоритмы отдельных экипажей и план совместных действий всех танков), получим следующую картину (см. рис. 6). Именно так выглядит экран компьютера во время игры в «Пожарные танки».

Игра «Пожарные танки» предназначена для освоения следующих элементов параллельного программирования:

1. Совместная работа нескольких исполнителей.
2. Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).
3. Исполнители – однотипные (танки) и исполнители – разнотипные (члены экипажа)
4. Соотношение «исполнители–работы» – N : M (разные танки тушат разные цели).
5. Согласование деятельности исполнителей. Виды согласования: по делению работы на части, по времени, по результатам. Необходимо распределить между танками, кто что и когда тушит. Для получения синергетического эффекта несколько танков должны действовать одновременно.
6. Оптимальный порядок действий.
7. Системный эффект: возрастание совместной мощности при объединении усилий нескольких танков.

Все алгоритмы, составляемые для «Пожарных танков», являются линейными (в системе команд отсутствуют развилки и циклы). Исполнение идет строго по заданному плану (отсутствуют механизмы обратной связи, команды ввода данных). Одиночные действия всех исполнителей строго синхронизированы (выполнение любой команды любым исполнителем занимает ровно один такт).

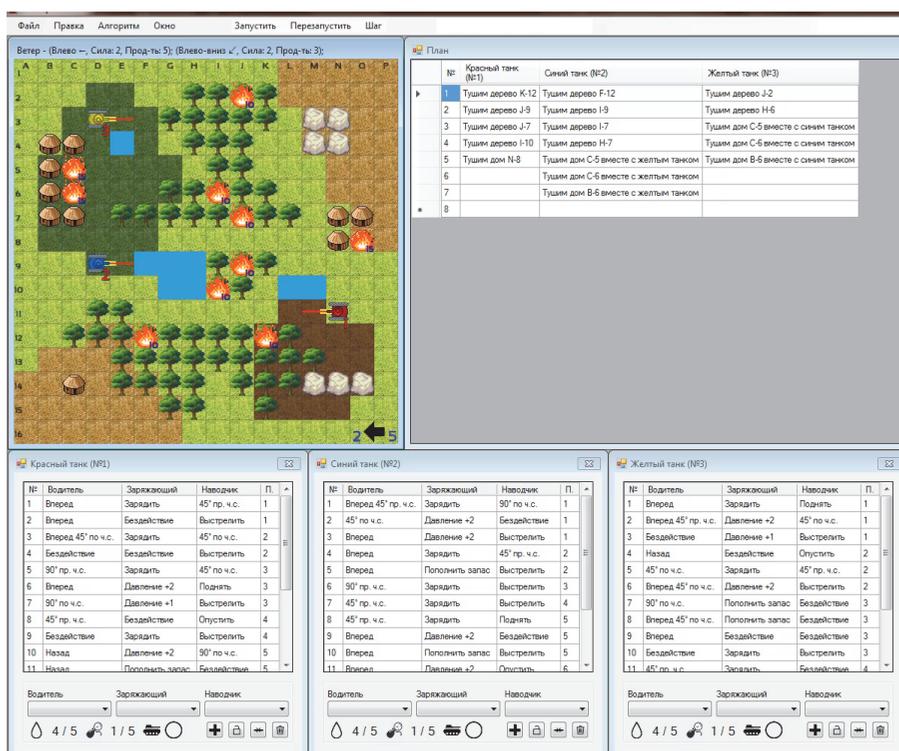


Рис. 6. Пример окна игры «Пожарные танки»

Fig. 6. Example of the game window «Fire tanks»



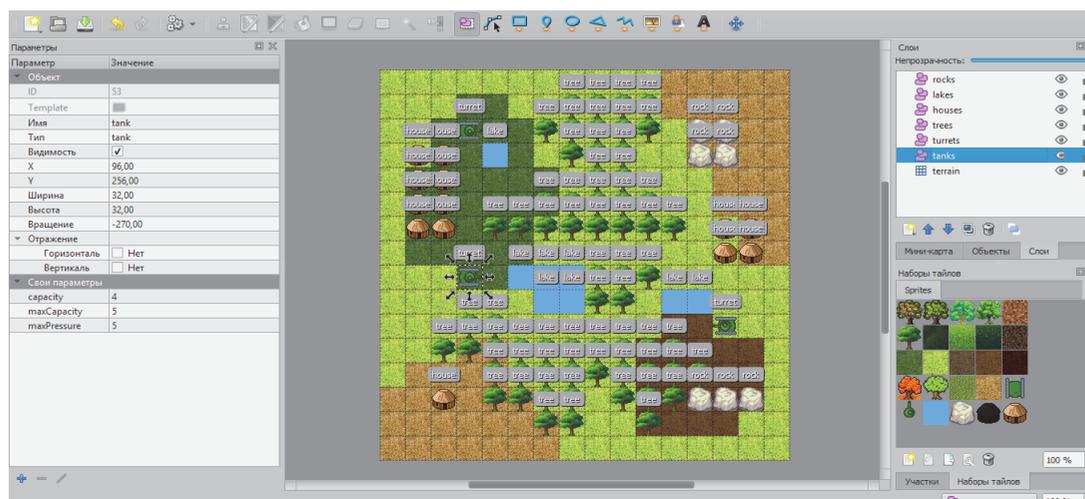


Рис. 7. Окно редактора Tiled Map Editor в процессе подготовки игровой карты для игры «Пожарные танки»

Fig. 7. The window of the TiledMapEditor editor in the process of preparing the game card for the game "Fire tanks"

Реализация игры «Пожарные танки»

Реализация игры «Пожарные танки» выполнена для настольных компьютеров для операционной системы Windows, начиная с версии Windows XP.

Игра состоит из двух программ: редактора зданий и проигрывателя алгоритмов.

В качестве редактора заданий используется Tiled Map Editor (ссылка на сайт: <https://www.mapeditor.org>).

Пример окна редактора в процессе разработки игровой карты для игры «Пожарные танки» приведен на рис. 7.

Апробация

Игра «Пожарные танки» прошла первичную апробацию на студентах Пермского филиала Высшей школы экономики направления «Бизнес-информатика» и на старшеклассниках лицея №10 г.Перми.

Фотография занятий с лицеистами приведена на рис. 8.



Рис. 8. Игра «Пожарные танки» на занятиях с учащимися лицея № 10 г.Перми

Fig. 8. The game "Fire Tanks" in the classroom with students of the Lyceum №10

Perm

В настоящее время проходит межрегиональный Интернет-конкурс «Параллельное программирование – с пелёнок!», участникам которого предлагается поочередно поиграть во все три упомянутые игры: «Танковый экипаж», «Рой роботов» и «Пожарные танки». Возраст участников – как и для конкурса «ТРИЗформашка» – с первого класса школы по четвертый курс университета. Конкурс начался в сентябре 2018 г. и продлится до февраля 2019. Присоединиться к конкурсу можно в любой момент, при условии, что уже упущенное время компенсировано не будет. Сайт конкурса – www.trizformashka.ru. Информация публикуется в группах ТРИЗформашка в социальных сетях Фейсбук и ВКонтакте (<https://www.facebook.com/groups/trizformashka/>, <http://vk.com/trizformashka>). По ходу конкурса планируется этап рефлексии, оценки участниками конкурса полученного ими опыта, степени полезности игр для учебного процесса.

Далее, планируется что игра «Пожарные танки» будет выставлена на конкурс «ТРИЗформашка-2019» в марте 2019 г.

Заключение

Доклад посвящен обсуждению серии компьютерных игр, предназначенных для пропедевтики в школьном курсе информатики темы «Параллельные вычисления». Главное внимание было уделено новой и наиболее сложной игре в данной теме «Пожарные танки». Игра реализована для системы Windows, прошла первичную апробацию на студентах и школьниках, в настоящее время представлена в конкурсе «Параллельное программирование – с пелёнок!», а далее – в конкурсе «ТРИЗформашка-2018».

Планируется, что игра найдет отражение в следующем издании «пермской версии» учебников информатики для начальной и средней школы.



Список использованных источников

- [1] Kirkpatrick K. Parallel Computational Thinking // *Communications of the ACM*. 2017. Vol. 60, issue 12. Pp. 17-19. DOI: 10.1145/3148760
- [2] Плаксин М.А. «Пермская версия» начального курса информатики // *Информатика и образование*. 2003. № 1. С. 84-90.
- [3] Plaksin M.A. TRIZformatics: A Metasubject Uniting Computer and Intelligence Technologies of Information Processing (Response to Information Society Challenge) // *Programming and Computer Software*. 2011. Vol. 37, issue 6. Pp. 279-283. DOI: 10.1134/S0361768811060053
- [4] Плаксин М.А. ТРИЗформатика – метапредмет, объединяющий компьютерные и интеллектуальные технологии работы с информацией (ответ на вызов информационного общества) // *Программирование*. 2011. Т. 37, № 6. С. 26-32. URL: <https://elibrary.ru/item.asp?id=17112942> (дата обращения: 16.07.2018).
- [5] Плаксин М.А. Информатика: учебник для 3 класса: в 2 ч. / М.А. Плаксин, Н.Г. Иванова, О.Л. Русакова. М.: БИНОМ. Лаборатория знаний, 2013.
- [6] Плаксин М.А. Информатика: учебник для 4 класса: в 2 ч. / М.А. Плаксин, Н.Г. Иванова, О.Л. Русакова. М.: БИНОМ. Лаборатория знаний, 2013.
- [7] Плаксин М.А. «Суперкомпьютеры» vs «параллельное программирование». «Параллельное программирование» vs «совместная деятельность». Как изучать тему «параллельные вычисления» в средней школе? // *Современные информационные технологии и ИТ-образование*. 2015. Том 11, № 1. 2015. С. 302-309. URL: <https://elibrary.ru/item.asp?id=25024598> (дата обращения: 16.07.2018).
- [8] Плаксин М.А. О некоторых методических средствах пропедевтики параллельных вычислений в школьной информатике // *Современные информационные технологии и ИТ-образование*. 2016. Том 12, № 3-1. С. 219-227. URL: <https://elibrary.ru/item.asp?id=27411995> (дата обращения: 16.07.2018).
- [9] Плаксин М.А., Цаплин С.В. Пропедевтика параллельных вычислений в школьной информатике: тема «Рой роботов» в конкурсе «Тризмашка-2017» // *Современные информационные технологии и ИТ-образование*. 2017. Том 13, № 4. С. 276-288. DOI: 10.25559/SITITO.2017.4.432
- [10] Иванова Н.Г., Плаксин М.А., Русакова О.Л. ТРИЗформашка // *Информатика. Приложение к газете «Первое сентября»*. № 5. 1-15.03.2010. С. 3-19.
- [11] Yadav A., Stephenson C., Hong H. Computational thinking for teacher education // *Communications of the ACM*. 2017. Vol. 60, issue 4. Pp. 55-62. DOI: 10.1145/2994591
- [12] Kalelioglu F. A new way of teaching programming skills to K-12 students: Code.org // *Computers in Human Behavior*. 2015. Vol. 52. Pp. 200-210. DOI: 10.1016/j.chb.2015.05.047
- [13] Marcelino M.J., Pessoa T., Vieira C., Salvador T., Mendes A.J. Learning Computational Thinking and scratch at distance // *Computers in Human Behavior*. 2018. Vol. 80. Pp. 470-477. DOI: 10.1016/j.chb.2017.09.025
- [14] Tsalapatas H., Heidmann O., Alimisi R., Houstis E. Game-Based Programming Towards Developing Algorithmic Thinking Skills In Primary Education // *Scientific Bulletin of the „Petru Maior” University of Tîrgu Mureş*. 2012. Vol. 9, no. 1. Pp. 56-63. URL: <http://www.oalib.com/paper/2701059?btwaf=76205623#XHbw7IgzblU> (дата обращения: 16.07.2018).
- [15] Gürbüz H., Evlioğlu B., Erol Ç.S. et al. “What’s the Weather Like Today?”: A computer game to develop algorithmic thinking and problem-solving skills of primary school pupils // *Education and Information Technologies*. 2017. Vol. 22, issue 3. Pp. 1133-1147. DOI: 10.1007/s10639-016-9478-9
- [16] Debabi W., Bensebaa T. Using Serious Game to Enhance Learning and Teaching Algorithmic // *Journal of e-Learning and Knowledge Society*. 2016. Vol. 12, no. 2. Pp. 127-140. DOI: 10.20368/1971-8829/1125
- [17] Topal A.D., Budak E.Ç., Geçer A.K. The effect of algorithm teaching on the problem-solving skills of deaf-hard hearing students // *Program*. 2017. Vol. 51, issue 4. Pp. 354-372. DOI: 10.1108/PROG-05-2017-0038
- [18] Алгоритмика: 5-7 классы: Учебник и задачник для общеобразоват. учебных заведений / А.К. Звонкин, А.Г. Кулаков, С.К. Ландо, А.Л. Семенов, А.Х. Шень. М.: Дрофа, 1997. 304 с.
- [19] Кучев А.Д., Плаксин М.А. Пропедевтика параллельных вычислений в школьной информатике. Компьютерная игра «Танковый экипаж» // *Информатика в школе*. 2016. № 9(122). С. 42-48. URL: <https://elibrary.ru/item.asp?id=27538413> (дата обращения: 16.07.2018).

Поступила 16.07.2018; принята в печать 10.09.2018;
опубликована онлайн 10.12.2018.

References

- [1] Kirkpatrick K. Parallel Computational Thinking. *Communications of the ACM*. 2017; 60(12):17-19. DOI: 10.1145/3148760
- [2] Plaksin M.A. “Perm version” of the initial course of informatics. *Informatics and education*. 2003; 1:84-90. (In Russian)
- [3] Plaksin M.A. TRIZformatics: A Metasubject Uniting Computer and Intelligence Technologies of Information Processing (Response to Information Society Challenge). *Programming and Computer Software*. 2011; 37(6):279-283. DOI: 10.1134/S0361768811060053
- [4] Plaksin M.A. TRIZformatics: A metasubject uniting computer and intelligence technologies of information processing (Response to Information Society Challenge). *Programmirovaniye = Programming and Computer Software*. 2011; 37(6):26-32. Available at: <https://elibrary.ru/item.asp?id=17112942> (accessed 10.06.2018). (In Russian)
- [5] Plaksin M.A., Ivanova N.G., Rusakova O.L. Informatika: uchebnik dlja 3 klassa: v 2 ch. [Informatics: the textbook for grade 3: 2 parts]. M.: BINOM. Laboratorija znaniy, 2013. (In Russian)
- [6] Plaksin M.A., Ivanova N.G., Rusakova O.L. Informatika: uchebnik dlja 4 klassa: v 2 ch. [Informatics: the textbook for grade 4: 2 parts]. M.: BINOM. Laboratorija znaniy, 2013. (In Russian)
- [7] Plaksin M.A. “Supercomputers” vs “parallel programming”. “Parallel programming” vs “joint activities”. How to study the topic of “parallel computing” in high school? *Modern Information Technologies and IT-Education*. 2015; 11(1):302-309. Available at: <https://elibrary.ru/item.asp?id=25024598> (accessed 10.06.2018). (In Russian)
- [8] Plaksin M.A. Some methodical tools for propaedeutics parallel computing in school informatics. *Modern Information Technologies and IT-Education*. 2016; 12(3-1):219-227. Available



- at: <https://elibrary.ru/item.asp?id=27411995> (accessed 10.06.2018). (In Russian)
- [9] Plaksin M.A., Tsaplin S.V. Propaedeutics Parallel Computing in School Informatics: the Topic «Swarm of Robots» in the Competition «TRIZformashka-2017». *Modern Information Technologies and IT-Education*. 2017; 13(4):276-288. (In Russian) DOI: 10.25559/SITITO.2017.4.432
- [10] Ivanova N.G., Plaksin M.A., Ruskova O.L. TRIZformashka [The contest TRIZformashka]. *Informatika. Prilozhenie k gazete «Pervoe sentyabrya» = Informatics*. No. 5 (606), 1-15.03.2010, pp. 3-19. (In Russian)
- [11] Yadav A., Stephenson C., Hong H. Computational thinking for teacher education. *Communications of the ACM*. 2017; 60(4):55-62. DOI: 10.1145/2994591
- [12] Kalelioglu F. A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*. 2015; 52:200-210. DOI: 10.1016/j.chb.2015.05.047
- [13] Marcelino M.J., Pessoa T., Vieira C., Salvador T., Mendes A.J. Learning Computational Thinking and scratch at distance. *Computers in Human Behavior*. 2018; 80:470-477. DOI: 10.1016/j.chb.2017.09.025
- [14] Tsalapatas H., Heidmann O., Alimisi R., Houstis E. Game-Based Programming Towards Developing Algorithmic Thinking Skills In Primary Education. *Scientific Bulletin of the „Petru Maior” University of Tîrgu Mureş*. 2012; 9(1):56-63. Available at: <http://www.oalib.com/paper/2701059?btwaf=76205623#.XHbw7IgzblU> (accessed 10.06.2018).
- [15] Gürbüz H., Evlioğlu B., Erol Ç.S. et al. “What’s the Weather Like Today?”: A computer game to develop algorithmic thinking and problem-solving skills of primary school pupils. *Education and Information Technologies*. 2017; 22(3):1133-1147. DOI: 10.1007/s10639-016-9478-9
- [16] Debabi W., Bensebaa T. Using Serious Game to Enhance Learning and Teaching Algorithmic. *Journal of e-Learning and Knowledge Society*. 2016; 12(2):127-140. DOI: 10.20368/1971-8829/1125
- [17] Topal A.D., Budak E.Ç., Geçer A.K. The effect of algorithm teaching on the problem-solving skills of deaf-hard hearing students. *Program*. 2017; 51(4):354-372. DOI: 10.1108/PROG-05-2017-0038
- [18] Zvonkin A.K., Kulakov A.G., Lando S.K., Semenov A.L., Shen A.K. *Algoritmika: 5-7 klassy: Uchebnik i zadachnik dlya obshcheobrazovatel'nykh uchebnykh zavedeniy [Algorithms: 5-7 classes: the textbook and book of problems for schools]*. M.: Drofa, 1997. 304 p. (In Russian)
- [19] Kuchev A.D., Plaksin M.A. Propaedeutics of Parallel Computing in School Informatics. The Computer Game “Tank Crew”. *Informatics in school*. 2016; 9(122):42-48. Available at: <https://elibrary.ru/item.asp?id=27538413> (accessed 10.06.2018). (In Russian)

Submitted 16.07.2018; revised 10.09.2018;
published online 10.12.2018.

About the authors:

Mikhail A. Plaksin, Ph.D. in Computer Science, Associate Professor of the Department of Information Technologies in Business of the Perm Branch, National Research University Higher School of Economics (38 Studencheskaya Str., Perm 614070, Russia); Associate Professor of the Department of Software of Computing Systems, Perm State National Research University (15 Bukireva Str., Perm 614990, Russia), ORCID: <http://orcid.org/0000-0002-6288-8610>, mapl@list.ru

Andrey A. Shchelkunov, graduate student of the Perm Branch, National Research University Higher School of Economics (38 Studencheskaya Str., Perm 614070, Russia), ORCID: <http://orcid.org/0000-0002-4286-2632>, would.you.kindly@yandex



This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted reuse, distribution, and reproduction in any medium provided the original work is properly cited.

