# МИХАЙЛОВА К.К., РОГОВ К.С.
# РАЗРАБОТКА ПРИЛОЖЕНИЯ “GLOVELIN”

***Иркутский государственный университет, Россия***

В современном мире новое поколение всё больше времени проводит, играя в игры на телефонах, компьютерах, приставках и т.д. Но получают ли они знания в процессе игр, развиваются ли? Можно ли разработать игру, которая будет интересна детям и в то же время научит их чему-то новому? Именно эти вопросы положили начало разработке приложения Glovelin.

Целью проекта стала разработка игрового приложения, способного обучить детей понятию алгоритма. Алгоритм - это совокупность последовательных шагов, схема действий, приводящих к желаемому результату.

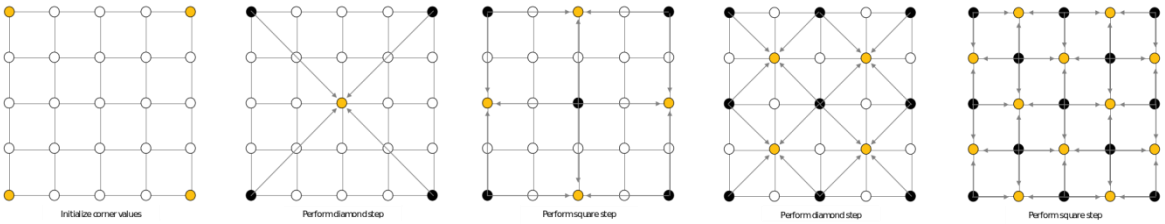
Основу приложения составляет карта, на которой расположен герой, различные объекты (монеты, сундуки, ключи и т.д.) и препятствия. Игрок с помощью различных кнопок действий составляет свой собственный алгоритм решения задачи. По нажатию на кнопку, приложение поэтапно показывает работу алгоритма и проверяется его на наличие ошибок.

В ходе разработки были выделены несколько основных этапов, о которых пойдет речь далее.

**Генерирование карты**

Карта в приложении представляет собой набор клеток, каждая из которых представляет собой определенный тип ландшафта (горы, равнина или вода). На равнинах располагаются различные объекты, с которыми можно взаимодействовать. Для удобства было решено хранить карты на удаленном сервере в формате JSON.

Ландшафт для новой карты генерируется с помощью алгоритма Diamond-Square.




Алгоритм начинает работу с двумерного массива размера 2n + 1. В четырех угловых точках массива устанавливаются начальные значения высот. Шаги diamond и square выполняются поочередно до тех пор, пока все значения массива не будут установлены.

Шаг diamond (ромб). Для каждого квадрата в массиве, находится срединная точка, в которую устанавливается среднее значение четырех угловых точек плюс случайное значение.

Шаг square (квадрат). Для каждого ромба в массиве, устанавливается срединная точка, которой присваивается среднее арифметическое из четырех угловых точек плюс случайное значение.

После генерирования ландшафта на карту добавляются объекты, карта переводится в формат JSON и сохраняется.

**Получение и подготовка данных**

При запуске любого уровня в приложении первым делом происходит загрузка карты с сервера, затем программа разбивает ее на отдельные объекты и хранит в виде списков, с которыми впоследствии и будет осуществлять работу.

**Построение карты**

Построение карты происходит сразу после загрузки данных. На заданном пространстве программа рисует ландшафт, а поверх него размещает все объекты. Также в заданной точке размещает герой, у которого также есть важное свойство - направление, которое отвечает за сторону, в которую он смотрит и в дальнейшем пойдет.

**Предварительный показ работы алгоритма и его проверка**

После того, как алгоритм готов, игрок нажимает кнопку “Проверить” и программа пошагово показывает работу алгоритма, по ходу проверяя его правильность. В случае ошибки в алгоритме программа завершит пошаговый показ и отметит данный шаг алгоритма неправильным для удобства игрока.

**Разработка интерфейса**

Экран приложения разделен на 3 главные области: карта, строка алгоритма и поле шагов алгоритма.

**Карта:** на данной карте присутствует герой и 3 монеты. Цель уровня: собрать все монеты. Герой направлен вверх. Для достижения цели уровня подойдет алгоритм: “повернуть направо” -> “идти вперед” -> “идти вперед” -> “взять” -> “идти вперед” -> “взять” -> “идти вперед” -> “взять”. Также над картой есть кнопка “play”, которая запускает проверку и поэтапный показ алгоритма.

**Строка алгоритма:** содержит кнопку очистки алгоритма, заполняется пользователем с помощью нажатий на кнопки действий, каждый шаг алгоритма имеет статус: непроверенный (отсутствует круг в правом нижнем углу), проверенный (зеленый круг) и ошибочный (красный круг). После запуска алгоритма происходит его валидация и каждому шагу присваивается статус (шаги после ошибочного не проверяются). В данном примере ошибка

происходит из-за того, что алгоритм ведёт героя в воду, а герой может перемещаться только по равнинам.

**Поле шагов алгоритма:** все возможные шаги алгоритма делятся на движения (идти вперед, повернуть налево, повернуть направо) и действия (взять). По нажатию на определенную кнопку в строку алгоритма добавляется новый шаг со статусом “непроверенный”.

**Заключение**

На данный момент приложение находится в разработке, планируется добавить еще множество функций, например, группировка одинаковых шагов алгоритма для улучшения его читаемости, добавление новых объектов и типов ландшафта, мультиплеер и многое другое.