# Глава 1. Обзор предметной области

Xonix (с англ. — «Зоникс») — компьютерная игра. Первая версия была создана в 1984 году для платформы PC как клон игры Qix, появившейся ранее на аркадных автоматах.

Аркада - распространенный в постсоветской индустрии компьютерных игр термин, обозначающий компьютерные игры с нарочно примитивным игровым процессом.

Игровое поле представляет собой прямоугольную сетку из квадратных ячеек. Движение всех объектов (*Xonix`a* и *врагов*) происходит строго по ячейкам, так что в каждый момент времени каждый объект занимает ровно одну ячейку. В Xonix у любого объекта есть всего 4 варианта направления движения: для Xonix`a — вверх/вниз/вправо/влево, для врагов (обоих типов) — то же самое, только по диагонали.

При движении по морю Xonix оставляет за собой след – трек. При движении по суше трека не остается. Когда Xonix вернется на сушу – море поделится треком на две части. Если в одной из областей не окажется врагов, то эта область станет сушей. Если же во всех отсеченных областях окажется по врагу, то сушей станет только трек от Xonix`a.

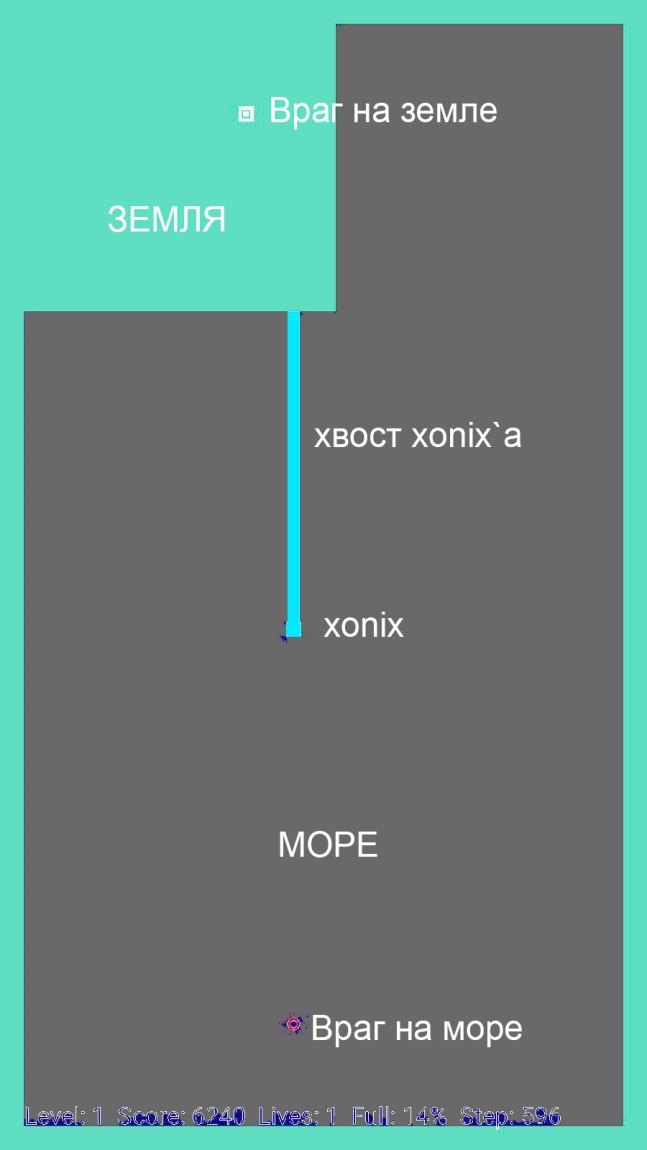
В начале игры Xonix имеет 3 жизни. Он теряет по одной жизни при столкновении с врагом, или если враг пересечет трек от Xonix`a (или даже сам Xonix). Т.к. трек остается только на море, то пересечь его может только враг на море. При переходе на новый уровень прибавляется одна жизнь. Игра заканчивается если жизни закончились.

Так же предусмотрена система бонусов. Когда Xonix увеличивает площадь суши, ему начисляются очки – чем больше добавленная площадь, тем больше очков. Как только количество очков превышает определенную границу – на море появляется бонус. Чтобы собрать его, Xonix должен отрезать часть моря с бонусом. Бонус может либо добавить жизнь, убрать одного врага или замедлить врагов в два раза.

Переход на новый уровень осуществляется тогда, когда Xonix отрежет от моря 75%. На новом уровне добавится один морской враг и прибавится одна жизнь.

В процессе игра в информационной строке отображается: количество очков, количество жизней, номер уровня, процент суши на поле.

Игра заканчивается если кончаются жизни, поэтому если играть аккуратно, то игра может быть бесконечной.



# Глава 2. Проектирование

***Xonix***

Знает:

- свои текущие координаты

- количество жизней

- количество очков

- приращение dx и dy

- не пересек ли он свой трек

Xonix может:

- двигаться

- прибавить жизнь

- отнять жизнь

- проверить, не пересек ли он свой трек

- обновить количество очков

***Враги на море***

Знают:

- свои текущие координаты

- приращение dx и dy

Могут:

- двигаться

- проверить и обновить dx и dy

- проверить столкновение с Xonix`ом или его треком

- добавить еще одного врага

- удалить одного врага

***Враг на суше***

Знает:

- свои текущие координаты

- приращение dx и dy

Может:

- двигаться

- проверить и обновить dx и dy

- проверить столкновение с Xonix`ом

***Поле***

Знает:

- значение каждой ячейки

- размеры поля

- процент заполнения поля

Может:

- превратить часть моря в сушу

- обновить процент заполнения поля

- удалить трек Xonix`а

Как известно, игровое поле Xonix представляет собой сетку из квадратных ячеек. Такая структура игрового поля значительно упрощает реализацию игры. Тот факт, что мы можем принять ячейку за «пиксель», избавляет от большинства проблем с вычислениями, которые обычно встречаются в играх со многими движущимися объектами: расчет движения, отскоков и столкновений и т.п.

В начале игры (уровня) большую часть поля занимает прямоугольная черная область («море»), которую окаймляет со всех сторон рамка («суша»), которая в начале уровня всегда имеет фиксированную ширину в 2 ячейки. Игра происходит строго в пределах поля. Размер одной ячейки определяется по ширине экрана программистом, например, 20px. Тогда поле экрана с шириной 1080px будет иметь 54 ячейки. Каждая ячейка поля может принимать какое-то одно из трех значений: суша, море, трек от Xonix`а.

На поле находятся Xonix, враги (в начале игры один морской (в ходе игры их число увеличивается) и один сухопутный) и бонусы (при достижении определенного результата).

***Взаимодействия поля и Xonix`а:***

Xonix имеет текущие координаты – x и y. При инициализации Xonix`а, которая происходит при включении игры или если Xonix`а задел враг, его текущие координаты меняются на центр верхней границы. В зависимости от направления, заданного органами управления, координаты по х и у будут увеличиваться или уменьшаться на 1. Когда Xonix начинает движение по морю, значения ячеек поля, по которым прошел Xonix, меняется на «трек». Xonix двигается до тех пор, пока не изменится направление движения (при этом Xonix продолжает двигаться) или не закончится поле. Как только текущие координаты указывают на то, что Xonix находится на суше, начинается проверка на наличие врагов в образовавшихся областях. Если в области нет врагов, то эта область становится сушей. Может быть такая ситуация, что в каждой области есть враги, тогда сушей станет только трек. Затем подсчитывается площадь суши. Выполняется вычисление процента суши и перерасчет очков, если площадь суши больше 75%, то выполняется алгоритм перехода на следующий уровень. Если сумма очков превышает заданную границу, то вызывается метод, добавляющий бонус.

***Взаимодействия поля и врагов:***

Враги на море – объекты массива врагов. На первом уровне массив состоит из одного объекта – одного врага на море. На каждом уровне создается новый элемент массива – новый враг. При обновлении экрана у каждого элемента массива вызывается метод, который проверяет - не совпадают ли координаты врага с треком Xonix`а или самим Xonix`ом.

Враг на суше – отдельный объект. На протяжении всей игры он остается один.

Враги двигаются по диагонали. У каждого врага есть две переменные (х, у), хранящие текущее положение, и две переменные (dx, dy), которые хранят значения (1, -1) направления движения по осям. При инициализации им задаются случайные текущие координаты и случайные направления движения (1, -1). Метод move прибавляет к текущим координатам х, у переменные направления dx, dy:

//x += dx; y += dy;

При этом проверяется – не достигнута ли граница, разделяющая море и сушу, или граница поля. Если достигнута, то соответствующая переменная направления меняет значение на противоположное.

***Взаимодействия врагов между собой:***

Столкновения врагов друг с другом можно игнорировать, просто позволяем им «проходить» сквозь друг друга. Поскольку все враги (одного типа) выглядят одинаково, то со стороны это ничем не будет отличаться от столкновения и отскока.

***Бонусы:***

Генерируется случайное число, указывающее на тип бонуса. Затем генерируются его координаты. Бонус может появиться только на море. Чтобы собрать бонус, нужно чтобы он оказался на суше (отрезать часть моря с бонусом). Как только бонус собран – вызывается соответствующий ему метод. Добавление жизни – переменная, хранящая количество жизней, увеличивается на 1. Удаление врага на море (бонус может появиться только если врагов больше 1) – удаляется один объект из массива врагов.

***Игровой цикл:***

В первую очередь следует отметить, что игра бесконечна. Поэтому единственной целью игры является получения максимального числа очков.

1. Обновление экрана
   1. Прорисовка поля в соответствии со значениями ячеек
   2. Прорисовка врагов
   3. Прорисовка Xonix`a
   4. Прорисовка информационной строки
   5. Прорисовка бонусов
2. Обновление координат Xonix`a и врагов и проверка условий:
   1. Самопересечение (Xonix пересек свой трек)
   2. Столкновение с врагом
   3. Если текущие координаты Xonix`a указывают на сушу и при этом на море есть трек, то производится обновление значений поля. Вычисляется новая сумма очков и в зависимости от значения – добавляется бонус.
   4. Если процент заполнения поля больше или равен 75%, то происходит переход на следующий уровень:
      1. Xonix помещается на центр верхней границы
      2. Прибавляется одна жизнь
      3. Добавляется один морской враг
      4. Поле принимает первоначальный вид
   5. Если Xonix`a «ранили», то отнимается одна жизнь и:
      1. Если жизней не осталось, то начинается новая игра – происходит сброс на первый уровень, полная инициализация Xonix`a (сброс жизней, местоположения, счета), уменьшение числа врагов до 1.
      2. Если жизни еще есть, то происходит инициализация Xonix`a – текущие координаты меняются на центр верхней границы.

# Глава 3. Реализация

**Описание классов:**

Класс игрового поля – Field:

Int FIELD\_HEIGHT; //Высота экрана в ячейках

Int FIELD\_WIDTH; //Ширина экрана в ячейках

Int POINT\_SIZE; // Размер одной ячейки в пикселях

Int[FIELD\_WIDTH][ FIELD\_HEIGHT] fieldArray; // Массив игрового поля

Int currentPercent; // Процент заполнения поля

Методы:

* Void initFieldArray() – заполнение массива игрового поля значениями

Если координаты ячейки – два внешних кольца, то 1(суша), иначе – 2(море)

* Void clearTrack() – удаление трека Xonix`a

Обход всего массива игрового поля. Если значение равно 3(трек), то меняем его на 2(море)

* Int getCurrentPercent() – возвращает число – сколько суши занимает игровое поле в процентах

Обход массива игрового поля (только той части, которая первоначально была морем). Подсчитывается сколько ячеек занято сушей. Возвращается отношение суши к морю.

**Игровой цикл:**

В мобильной версии управление Xonix`ом осуществляется «свайпами», в компьютерной – «стрелочками».