

EngineAI Bomberman

28 декабря 2021 г.

Глава 1

Package model

<i>Package Contents</i>	<i>Page</i>
Classes	
Action 2 Класс, содержащий действие игрока	2
Bomb 2 Класс, описывающий бомбу на игровом поле	2
Bonus 4 Класс, описывающий бонус на игровом поле	4
BonusType 5 Перечисление, описывающее типы бонусов	5
Explosion 5 Класс, описывающий взрыв на игровом поле	5
Obstacle 6 Класс, описывающий препятствие на игровом поле	6
Player 7 Класс, описывающий игрового персонажа на игровом поле	7
Rules 11 Класс, содержащий постоянные, описывающие игровые правила	11
Strategy 14 Класс, который должна расширять стратегия игрока	14
World 15 Класс описывающий состояние игрового мира	15
WorldObject 17 Класс, содержащий информацию об объекте на игровом поле	17
WorldObjectType 19 Перечисление, описывающее типы объектов на игровом поле	19

1.1 Class Action

Класс, содержащий действие игрока

1.1.1 Объявление

```
public class Action
    extends java.lang.Object
```

1.1.2 Список полей

- drop** Размещать или нет в начале тика бомбу.
- speedX** Скорость перемещения вдоль осей координат.
- speedY** Скорость перемещения вдоль осей координат.

1.1.3 Поля

- **public double speedX**
 - Скорость перемещения вдоль осей координат. Длина вектора (speedX,speedY) не должна превышать World.PLAYER_MAX_SPEED.
- **public double speedY**
 - Скорость перемещения вдоль осей координат. Длина вектора (speedX,speedY) не должна превышать World.PLAYER_MAX_SPEED.
- **public boolean drop**
 - Размещать или нет в начале тика бомбу.

1.2 Class Bomb

Класс, описывающий бомбу на игровом поле

1.2.1 Объявление

```
public class Bomb
    extends model.WorldObject
```

1.2.2 Список полей

timeout Количество тиков, через которые бомба взорвётся

1.2.3 Список методов

- getOwner()** Получить номер игрока, разместившего бомбу
- getWalkableFor()** Получить копию списка номеров игроков, которые могут перемещаться по бомбе
- isWalkableFor(int)** Проверить, что игрок player может перемещаться по бомбе

1.2.4 Поля

- `public final int timeout`
 - Количество тиков, через которые бомба взорвётся

1.2.5 Методы

- `getOwner`

```
public int getOwner()
```

- **Описание**
Получить номер игрока, разместившего бомбу
- **Возвращает** – номер владельца

- `getWalkableFor`

```
public java.util.List<Integer> getWalkableFor()
```

- **Описание**
Получить копию списка номеров игроков, которые могут перемещаться по бомбе
- **Возвращает** – список номеров игроков

- `isWalkableFor`

```
public boolean isWalkableFor(int player)
```

- **Описание**
Проверить, что игрок `player` может перемещаться по бомбе
- **Parameters**
 - * `player` – номер игрока
- **Возвращает** – может ли игрок перемещаться по бомбе

1.2.6 Методы, унаследованные от класса `WorldObject`

`model.WorldObject` (in 1.11, page 17)

- `public double getX()`
- `public double getY()`
- `public boolean isCircle()`
- `public boolean isSquare()`
- `public void setX(double x)`
- `public void setY(double y)`
- `public final size`
- `public final type`

1.3 Class Bonus

Класс, описывающий бонус на игровом поле

1.3.1 Объявление

```
public class Bonus
    extends model.WorldObject
```

1.3.2 Список полей

bonusType Тип бонуса, который игрок получит, подняв его

1.3.3 Список методов

getImmuneDuration() Сколько ходов бонус нельзя подорвать бомбой

isImmune() Является ли бонус неуязвимым в данный тик

1.3.4 Поля

- **public final BonusType bonusType**
 - Тип бонуса, который игрок получит, подняв его

1.3.5 Методы

- **getImmuneDuration**

```
public int getImmuneDuration()
```

- **Описание**
Сколько ходов бонус нельзя подорвать бомбой
- **Возвращает** – количество тиков

- **isImmune**

```
public boolean isImmune()
```

- **Описание**
Является ли бонус неуязвимым в данный тик
- **Возвращает** – неуязвим ли бонус

1.3.6 Методы, унаследованные от класса `WorldObject`

`model.WorldObject` (in 1.11, page 17)

- `public double getX()`
- `public double getY()`
- `public boolean isCircle()`
- `public boolean isSquare()`
- `public void setX(double x)`
- `public void setY(double y)`
- `public final size`
- `public final type`

1.4 Class `BonusType`

Перечисление, описывающее типы бонусов

1.4.1 Объявление

```
public final class BonusType
    extends java.lang.Enum
```

1.4.2 Список полей

BonusBombs Бонус, увеличивающий количество размещаемых бомб

BonusPower Бонус, увеличивающий мощность размещаемых бомб

NoBonus Нет бонуса

1.4.3 Поля

- `public static final BonusType NoBonus`
 - Нет бонуса
- `public static final BonusType BonusBombs`
 - Бонус, увеличивающий количество размещаемых бомб
- `public static final BonusType BonusPower`
 - Бонус, увеличивающий мощность размещаемых бомб

1.5 Class `Explosion`

Класс, описывающий взрыв на игровом поле

1.5.1 Объявление

```
public class Explosion
    extends model.WorldObject
```

1.5.2 Список методов

`getCreators()` Возвращает игроков, создавших взрыв напрямую или косвенно
`getTimeout()` Получить количество тиков, через которые взрыв исчезнет с игрового поля

1.5.3 Методы

- `getCreators`

```
public java.util.Set getCreators()
```

- **Описание**

Возвращает игроков, создавших взрыв напрямую или косвенно

- **Возвращает** – множество игроков, создавших взрыв на игровом поле

- `getTimeout`

```
public int getTimeout()
```

- **Описание**

Получить количество тиков, через которые взрыв исчезнет с игрового поля

- **Возвращает** – количество тиков

1.5.4 Методы, унаследованные от класса `WorldObject`

`model.WorldObject` (in 1.11, page 17)

- `public double getX()`
- `public double getY()`
- `public boolean isCircle()`
- `public boolean isSquare()`
- `public void setX(double x)`
- `public void setY(double y)`
- `public final size`
- `public final type`

1.6 Class `Obstacle`

Класс, описывающий препятствие на игровом поле

1.6.1 Объявление

```
public class Obstacle  
    extends model.WorldObject
```

1.6.2 Список методов

`getContents()` Возвращает тип бонуса, который находится внутри препятствия
`setContents(BonusType)`

1.6.3 Методы

- `getContents`

```
public BonusType getContents ()
```

– **Описание**

Возвращает тип бонуса, который находится внутри препятствия

– **Возвращает** – тип бонуса

1.6.4 Методы, унаследованные от класса `WorldObject`

`model.WorldObject` (in 1.11, page 17)

- `public double getX()`
- `public double getY()`
- `public boolean isCircle()`
- `public boolean isSquare()`
- `public void setX(double x)`
- `public void setY(double y)`
- `public final size`
- `public final type`

1.7 Class `Player`

Класс, описывающий игрового персонажа на игровом поле

1.7.1 Объявление

```
public class Player  
    extends model.WorldObject
```

1.7.2 Список полей

`playerId` Номер игрока

`playerName` Имя игрока

1.7.3 Список методов

- canPerformActions()** Проверяет, будет ли стратегия совершать действия в текущий тик.
- clone()**
- getBombsCount()** Возвращает количество бомб, которые может разместить игрок
- getBombsLeft()** Возвращает количество бомб, которые остались у игрока с учетом уже расставленных
- getBombsPower()** Возвращает силу взрыва бомб игрока
- getHasCrashed()** Возвращает, была ли аварийно завершена стратегия игрока
- getImmuneDuration()** Получить количество тиков, которые игрок не подвержен действию взрывов
- getLivesLeft()** Получить количество жизней, оставшихся у игрока
- getScore()** Получить количество набранных игроком баллов
- getStunDuration()** Возвращает количество тиков, которые игрок не может совершать действия
- isImmune()** Проверяет, является ли игрок неуязвимым в данный тик.
- isStunned()** Проверяет, оглушен ли игрок на данный тик.

1.7.4 Поля

- **public final int playerId**
 - Номер игрока
- **public final java.lang.String playerName**
 - Имя игрока

1.7.5 Методы

- **canPerformActions**

```
public boolean canPerformActions()
```

- **Описание**

Проверяет, будет ли стратегия совершать действия в текущий тик. Она не может совершать действия, если оглушена или упала

- **Возвращает** –

- **getBombsCount**

```
public int getBombsCount()
```

- **Описание**

Возвращает количество бомб, которые может разместить игрок

- **Возвращает** – количество бомб

- **getBombsLeft**

```
public int getBombsLeft ()
```

- **Описание**

Возвращает количество бомб, которые остались у игрока с учетом уже расставленных

- **Возвращает** – количество бомб

- **getBombsPower**

```
public int getBombsPower ()
```

- **Описание**

Возвращает силу взрыва бомб игрока

- **Возвращает** – силу взрыва

- **getHasCrashed**

```
public boolean getHasCrashed ()
```

- **Описание**

Возвращает, была ли аварийно завершена стратегия игрока

- **Возвращает** – была ли аварийно завершена стратегия

- **getImmuneDuration**

```
public int getImmuneDuration ()
```

- **Описание**

Получить количество тиков, которые игрок не подвержен действию взрывов

- **Возвращает** – количество тиков

- **getLivesLeft**

```
public int getLivesLeft ()
```

- **Описание**

Получить количество жизней, оставшихся у игрока

- **Возвращает** – количество жизней

- **getScore**

```
public int getScore()
```

- **Описание**
Получить количество набранных игроком баллов
- **Возвращает** – количество баллов

- **getStunDuration**

```
public int getStunDuration()
```

- **Описание**
Возвращает количество тиков, которые игрок не может совершать действия
- **Возвращает** – количество тиков

- **isImmune**

```
public boolean isImmune()
```

- **Описание**
Проверяет, является ли игрок неуязвимым в данный тик. Игрок неуязвим несколько тиков после того, как его задело бомбой, или если у него закончились жизни
- **Возвращает** – неуязвим ли игрок

- **isStunned**

```
public boolean isStunned()
```

- **Описание**
Проверяет, оглушен ли игрок на данный тик. Игрок оглушен несколько тиков после того, как его задело бомбой, или если у него закончились жизни
- **Возвращает** – оглушен ли игрок

1.7.6 Методы, унаследованные от класса WorldObject

model.WorldObject (in 1.11, page 17)

- public double getX()
- public double getY()
- public boolean isCircle()
- public boolean isSquare()
- public void setX(double x)
- public void setY(double y)
- public final size
- public final type

1.8 Class Rules

Класс, содержащий постоянные, описывающие игровые правила

1.8.1 Объявление

```
public class Rules
    extends java.lang.Object
```

1.8.2 Список полей

BOMB_DELAY Количество тиков, через которые бомба взрывается

BOMB_SIZE Диаметр окружности, обозначающей бомбу а игровом поле

BOMBS_INITIAL_AMOUNT Начальное количество бомб игрока

BOMBS_INITIAL_POWER Начальная сила взрыва бомб игрока

BONUS_BOMB_COUNT Количество бонусов с дополнительными бомбами на игровом поле

BONUS_IMMUNE_DURATION Количество тиков, которые бонус нельзя взорвать

BONUS_PICKUP_SCORE Количество очков, которые игрок получит за поднятие бонуса

BONUS_POWER_COUNT Количество бонусов с увеличением мощности бомб на игровом поле

BONUS_SIZE Длина стороны квадрата, обозначающего бонус на игровом поле

BRICK_COUNT Количество разрушаемых препятствий на игровом поле

BRICK_SIZE Длина стороны квадрата, обозначающего препятствие на игровом поле

EXPLOSION_DURATION Количество тиков, которые взрыв остаётся на игровом поле

EXPLOSION_SIZE Длина стороны квадрата, обозначающего взрыв на игровом поле

GAME_DURATION Длительность игры в тиках

HEIGHT Высота игрового поля

PLAYER_AFTER_HIT_IMMUNE_DURATION Количество тиков, которые игрок неуязвим после того, как он был оглушен

PLAYER_AFTER_HIT_STUN_DURATION Количество тиков, которые игрок оглушен после взрыва

PLAYER_COUNT Количество игроков

PLAYER_HIT_SCORE Количество очков, которые игрок получает за подрыв игрока

PLAYER_LIVES_COUNT Начальное количество жизней игрока

PLAYER_MAX_SPEED Максимальная скорость перемещения игрока

PLAYER_SELF_HIT_PENALTY Количество очков, которые игрок теряет за подрыв самого себя

PLAYER_SIZE Диаметр окружности, обозначающей игрока в игровом поле

WALL_HIT_SCORE Количество очков, которые игрок получает за разрушение стены

WIDTH Ширина игрового поля

WORLD_HEIGHT_IN_BRICKS Высота игрового поля, измеренная в препятствиях

WORLD_WIDTH_IN_BRICKS Ширина игрового поля, измеренная в препятствиях

1.8.3 Список методов

snapToGrid(double) Получает ближайшую координату центра сетки, к которой привязано размещение бомб

1.8.4 Поля

- `public int PLAYER_COUNT`
 - Количество игроков
- `public double PLAYER_SIZE`
 - Диаметр окружности, обозначающей игрока в игровом поле
- `public int PLAYER_LIVES_COUNT`
 - Начальное количество жизней игрока
- `public double PLAYER_MAX_SPEED`
 - Максимальная скорость перемещения игрока
- `public int PLAYER_AFTER_HIT_IMMUNE_DURATION`
 - Количество тиков, которые игрок неуязвим после того, как он был оглушен
- `public int PLAYER_AFTER_HIT_STUN_DURATION`
 - Количество тиков, которые игрок оглушен после взрыва
- `public double BOMB_SIZE`
 - Диаметр окружности, обозначающей бомбу в игровом поле
- `public int BOMB_DELAY`
 - Количество тиков, через которые бомба взрывается
- `public int BOMBS_INITIAL_POWER`
 - Начальная сила взрыва бомб игрока
- `public int BOMBS_INITIAL_AMOUNT`
 - Начальное количество бомб игрока
- `public int BONUS_SIZE`
 - Длина стороны квадрата, обозначающего бонус на игровом поле

- `public int BONUS_IMMUNE_DURATION`
 - Количество тиков, которые бонус нельзя взорвать
- `public int BONUS_BOMB_COUNT`
 - Количество бонусов с дополнительными бомбами на игровом поле
- `public int BONUS_POWER_COUNT`
 - Количество бонусов с увеличением мощности бомб на игровом поле
- `public double EXPLOSION_SIZE`
 - Длина стороны квадрата, обозначающего взрыв на игровом поле
- `public int EXPLOSION_DURATION`
 - Количество тиков, которые взрыв остаётся на игровом поле
- `public double BRICK_SIZE`
 - Длина стороны квадрата, обозначающего препятствие на игровом поле
- `public int BRICK_COUNT`
 - Количество разрушаемых препятствий на игровом поле
- `public int WORLD_WIDTH_IN_BRICKS`
 - Ширина игрового поля, измеренная в препятствиях
- `public int WORLD_HEIGHT_IN_BRICKS`
 - Высота игрового поля, измеренная в препятствиях
- `public int WALL_HIT_SCORE`
 - Количество очков, которые игрок получает за разрушение стены
- `public int PLAYER_HIT_SCORE`
 - Количество очков, которые игрок получает за подрыв игрока
- `public int PLAYER_SELF_HIT_PENALTY`
 - Количество очков, которые игрок теряет за подрыв самого себя
- `public int BONUS_PICKUP_SCORE`
 - Количество очков, которые игрок получит за поднятие бонуса
- `public double WIDTH`
 - Ширина игрового поля
- `public double HEIGHT`
 - Высота игрового поля
- `public int GAME_DURATION`
 - Длительность игры в тиках

1.8.5 Методы

- `snapToGrid`

```
public double snapToGrid(double x)
```

- **Описание**

Получает ближайшую координату центра сетки, к которой привязано размещение бомб

- **Parameters**

- * `x` – заданная координата

- **Возвращает** – координата, ближайшая к центру клетки

1.9 Class Strategy

Класс, который должна расширять стратегия игрока

1.9.1 Объявление

```
public abstract class Strategy  
    extends java.lang.Object
```

1.9.2 Список методов

`getAction(Player, World)` Получить действия игрока на текущий тик

`getRules()` Получить игровые правила, сохранённые в классе.

`init(Rules)` Проинициализировать стратегию, задав ей правила.

1.9.3 Методы

- `getAction`

```
public abstract Action getAction(Player player, World world)
```

- **Описание**

Получить действия игрока на текущий тик

- **Parameters**

- * `player` – текущий игрок

- * `world` – состояние мира в данный тик

- **Возвращает** – действие игрока в данный тик

- `getRules`

```
public final Rules getRules()
```

– **Описание**

Получить игровые правила, сохранённые в классе. Можно использовать после вызова `init`.

– **Возвращает** – сохранённые игровые правила

• **init**

```
public final void init(Rules rules)
```

– **Описание**

Проинициализировать стратегию, задав ей правила. Всегда вызывается перед первым запросом действия игрока.

– **Parameters**

* `rules` – класс с игровыми правилами

1.10 Class World

Класс описывающий состояние игрового мира

1.10.1 Объявление

```
public class World  
    extends java.lang.Object
```

1.10.2 Список полей

`tick` Текущий номер тика

1.10.3 Список методов

`getBombs()` Получить бомбы на игровом поле
`getBonuses()` Получить бонусы на игровом поле
`getExplosions()` Получить взрывы на игровом поле
`getObstacles()` Получить препятствия на игровом поле
`getPlayers()` Получить игроков на игровом поле

1.10.4 Поля

- `public final int tick`
 - Текущий номер тика

1.10.5 Методы

- `getBombs`

```
public Bomb[] getBombs()
```

- **Описание**
Получить бомбы на игровом поле
- **Возвращает** – массив бомб

- `getBonuses`

```
public Bonus[] getBonuses()
```

- **Описание**
Получить бонусы на игровом поле
- **Возвращает** – массив бонусов

- `getExplosions`

```
public Explosion[] getExplosions()
```

- **Описание**
Получить взрывы на игровом поле
- **Возвращает** – массив взрывов

- `getObstacles`

```
public Obstacle[] getObstacles()
```

- **Описание**
Получить препятствия на игровом поле
- **Возвращает** – массив препятствий

- `getPlayers`

```
public Player[] getPlayers()
```

- **Описание**
Получить игроков на игровом поле
- **Возвращает** – массив игроков

1.11 Class WorldObject

Класс, содержащий информацию об объекте на игровом поле

1.11.1 Объявление

```
public class WorldObject
    extends java.lang.Object
```

1.11.2 Все известные наследники

Bonus (in 1.3, page 4), Obstacle (in 1.6, page 6), Bomb (in 1.2, page 2), Player (in 1.7, page 7), Explosion (in 1.5, page 5)

1.11.3 Список полей

size Размер объекта: диаметр окружности или длина стороны квадрата
type Тип объекта

1.11.4 Список методов

getX() Возвращает координату x центра объекта
getY() Возвращает координату y центра объекта
isCircle() Возвращает, является ли объект кругом на игровом поле
isSquare() Возвращает, является ли объект квадратом на игровом поле
setX(double)
setY(double)

1.11.5 Поля

- `public final WorldObjectType type`
 - Тип объекта
- `public final double size`
 - Размер объекта: диаметр окружности или длина стороны квадрата

1.11.6 Constructors

- `WorldObject`

```
public WorldObject(WorldObjectType type, double x, double y, double
    size)
```

1.11.7 Методы

- **getX**

```
public double getX()
```

- **Описание**
Возвращает координату x центра объекта
- **Возвращает** – координату x

- **getY**

```
public double getY()
```

- **Описание**
Возвращает координату y центра объекта
- **Возвращает** – координату y

- **isCircle**

```
public boolean isCircle()
```

- **Описание**
Возвращает, является ли объект кругом на игровом поле
- **Возвращает** – является ли объект кругом

- **isSquare**

```
public boolean isSquare()
```

- **Описание**
Возвращает, является ли объект квадратом на игровом поле
- **Возвращает** – является ли объект квадратом

- **setX**

```
public void setX(double x)
```

- **setY**

```
public void setY(double y)
```

1.12 Class WorldObjectType

Перечисление, описывающее типы объектов на игровом поле

1.12.1 Объявление

```
public final class WorldObjectType
    extends java.lang.Enum
```

1.12.2 Список полей

Bomb Бомба
Bonus Бонус
BrickWall Разрушаемое препятствие
ConcreteWall Неразрушаемое препятствие
count
Explosion Взрыв
None Нет типа
Player Игрок

1.12.3 Поля

- `public static final WorldObjectType None`
 - Нет типа
- `public static final WorldObjectType Player`
 - Игрок
- `public static final WorldObjectType Bomb`
 - Бомба
- `public static final WorldObjectType Explosion`
 - Взрыв
- `public static final WorldObjectType ConcreteWall`
 - Неразрушаемое препятствие
- `public static final WorldObjectType BrickWall`
 - Разрушаемое препятствие
- `public static final WorldObjectType Bonus`
 - Бонус

1.12.4 Методы

- `getById`

```
public static WorldObjectType getById(int id)
```

- `valueOf`

```
public static WorldObjectType valueOf(java.lang.String name)
```

- `values`

```
public static WorldObjectType[] values()
```

1.12.5 Методы, унаследованные от класса Enum

```
java.lang.Enum
```

- `public final int compareTo(Enum arg0)`
- `public final boolean equals(Object arg0)`
- `public final Class getDeclaringClass()`
- `public final int hashCode()`
- `public final String name()`
- `public final int ordinal()`
- `public String toString()`
- `public static Enum valueOf(Class arg0, String arg1)`