**Class Player**

Player number (int)

Wagon (int)

Bottom row (bool) *// или row(int) с 0 или 1, тк нужен индекс для карты*

Position in wagon row (int)

Is dead (bool)

Is shot (bool)

Facing forward (bool)

Methods

If shot()

If on last wagon()

If on first wagon()

Get indexForMap(return [wagon][row])

**Class Map**

Number of players (int) *// может оставить только array.lenght?*

Players left (array)

Round (int)

Players turn at round start (int)

Players turn at turn start (int)

Methods

Next round()

* Delete [0] wagon
* Players in [0] wagon become dead
* Players turn at round start = Players turn at round start (next in array), else first in array
* Count the number of players left, change array of players
* Round++
* Pick up all shot players

Next turn()

* Players turn at turn start = Players turn at turn start (next in array), else first in array

**Variables**

Map (arr) *// [ [ ][ ] ],[ [ ][ ] ],[ [ ][ ] ] (number of players +2)*

Var map = []

For i in map.Number of players-1 => map.push(new Player(data, data, data));

// each turn changes the array

01 11 21 31 41

00 10 20 30 40

**Functions**

Climb()

Move()

TurnAround()

Shoot()

**To do:**

~~Очередность игроков~~

Подсчет ходов до конца раунда, следующий раунд, следующий живой игрок

Убирание вагона в конце раунда + объявление мертвыми игроков

Добавить случайную сумму денег ближнему игроку в предпоследнем вагоне(?)