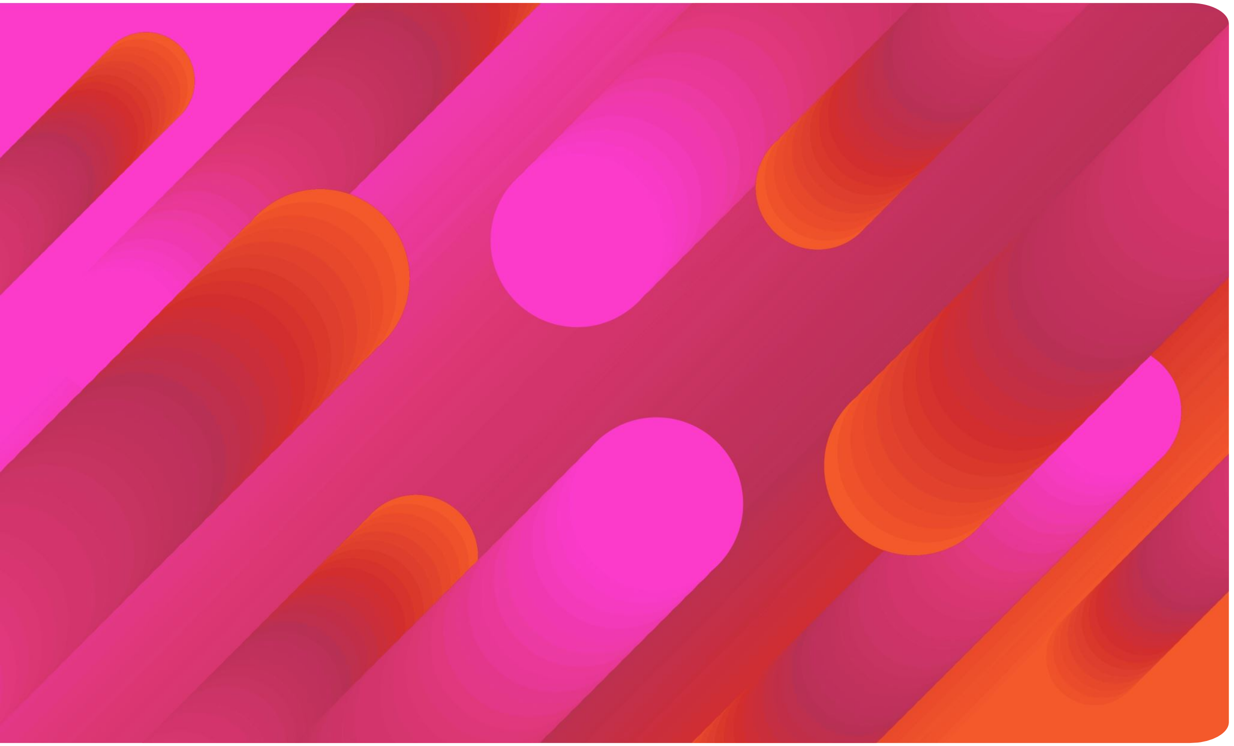


BATTLESHIP GAME

Nikita Zhukov & Diogo Gomez

Introduction

- Organization
- Game Logic

Organization

main.c

#Game logic -
battleship.*
ship_setup.*

#Structures –
user.*
board.*
ship.*

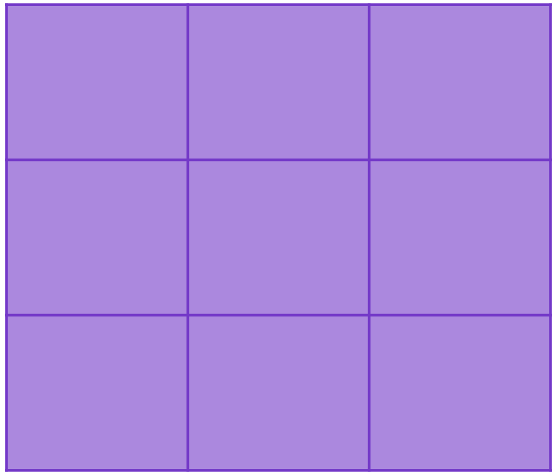
#Others –
io_handler.*
utilities.h

Ship

```
{        int hp
         int type
         int size
         char bitmap[3][3]        }
```

hp = health
type {0..4}

Ship’s bitmap



3x3

Bitmap

Bitmap is static

#	#	#
#	M	#
#	#	#

Monomino

#	D	#
#	D	#
#	#	#

Domino

#	T	#
#	T	#
#	T	#

Tromino

#	T	#
#	T	T
#	T	#

T Tetromino

#	L	#
#	L	#
#	L	L

L Tetromino

Ship's methods

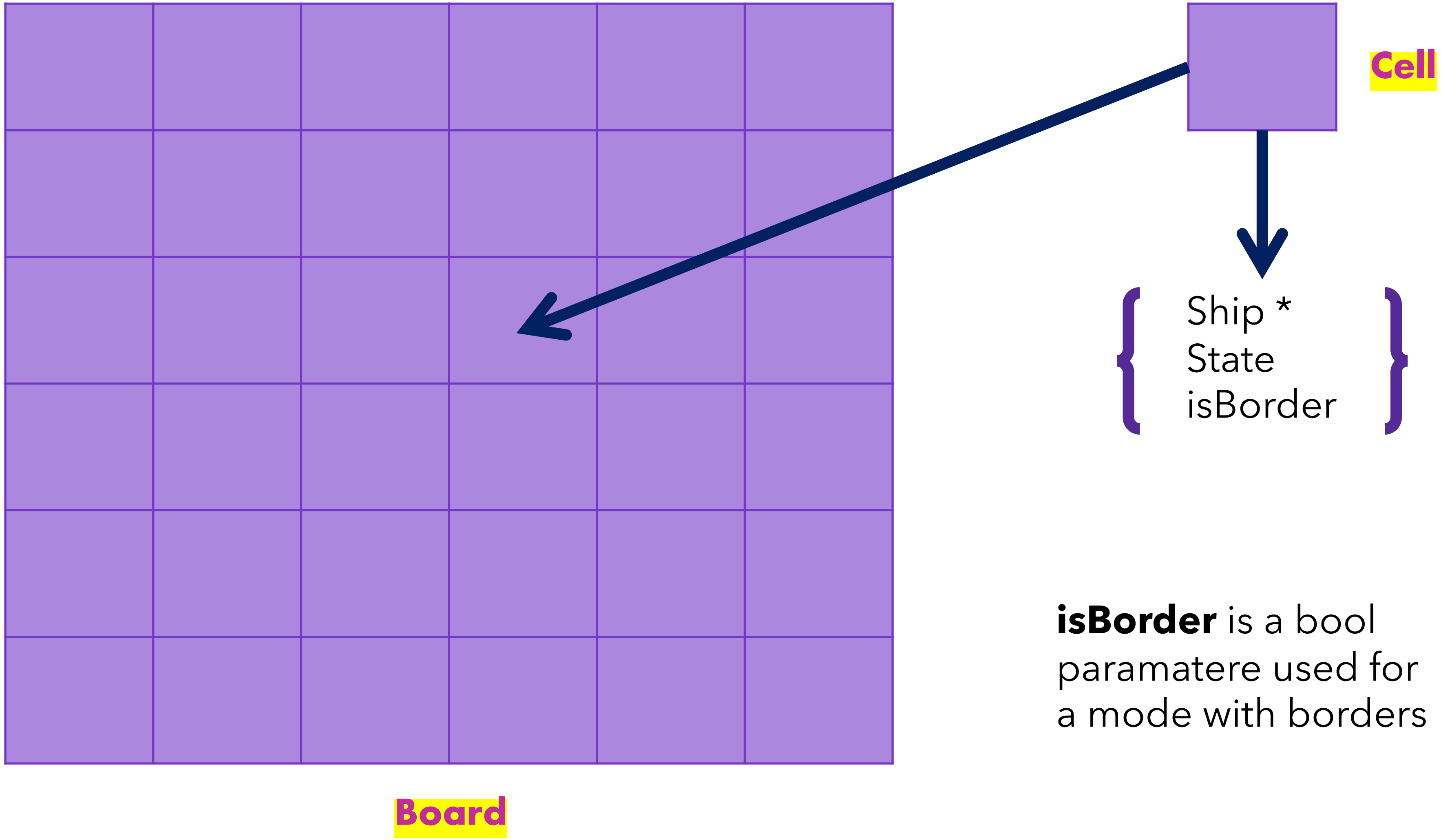
- shipCreate
- shipRotate (in clockwise direction)
- shipPrint
- destroyShip

Board

Every Cell contains the pointer to the ship.

After the initialization of a board the Board[i][j].ship = **NULL**

Once the ship is inserted
NULL changes to the **address** of the inserted ship



{
Ship *
State
isBorder
}

isBorder is a bool paramatere used for a mode with borders

Organization/board.*

Board’s methods

- boardInit
- boardPrint
- boardDestroy

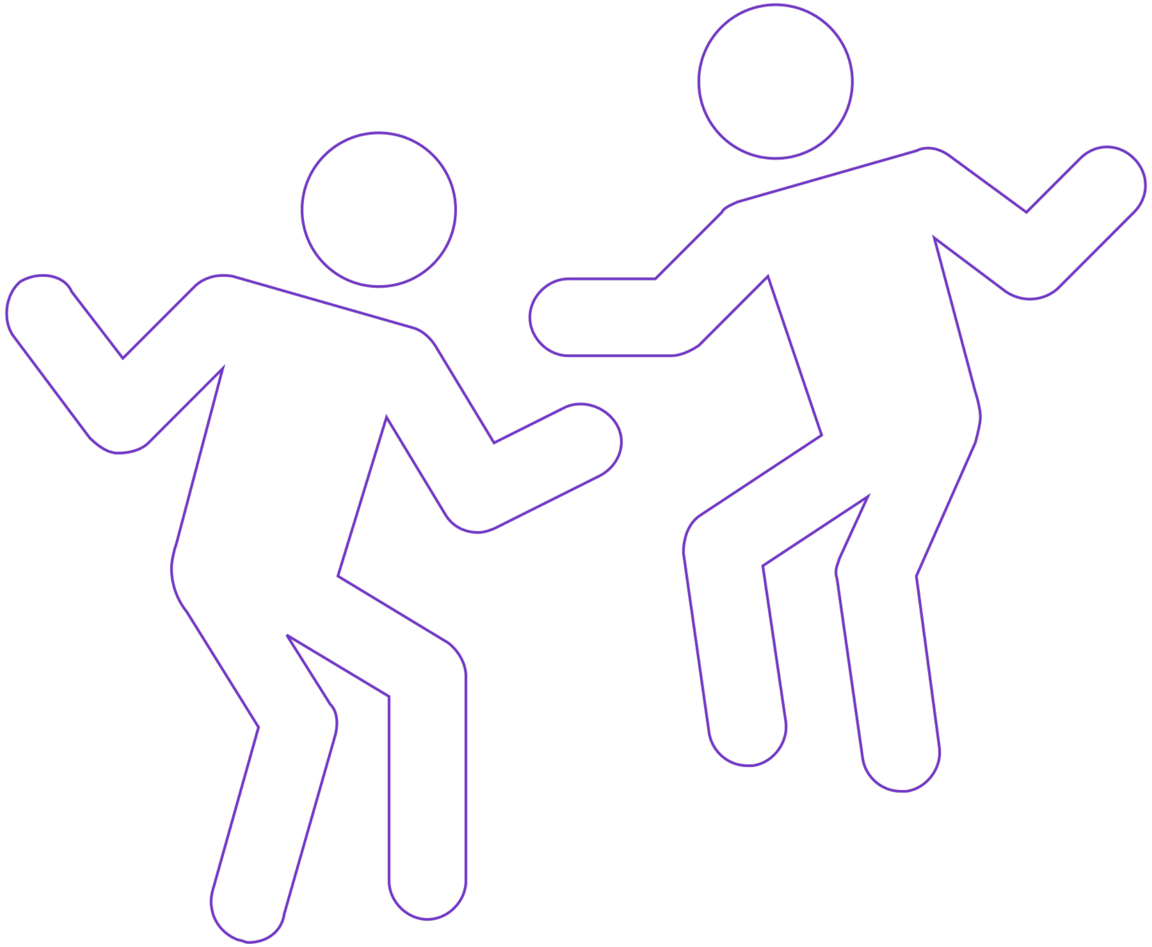
User

User encapsulates board and ship

```
{
        int id
        Cell** board
        int ships
}
```

id = number of user (0,1)
ships = number of ships that user has

Organization/user.*



Game Logic

We have **2** Users

Initialization/ship_setup.*

There are two types of initialization:

- **Manual**
- **Random**

1. **Manual mode** allows to user to choose where he wants to place his/her ship. During insertion can happen fail, if user put coordinates out of board or this coordinate is already occupuied. For that case we user simple array called **coordToFree** that contains failed coordinates. It sets **ship*** to **NULL**.

2. **Random mode** uses **<time.h>** for generation of random numbers. Also we use two functions that keeps random number in a rational range called **getRandomNumber** and **initShipsRandomly**.

Once both boards are **initiliazed** the game starts

Idea

- User shoots until he misses, once he misses another user is in charge

- Active user (user who makes shoots) see only opponent's board

- Once all opponent's ship are sunk game is finished

Shoots

Once user manage to hit the opponent’s ship its **hp** decreases by one. If ship had one hp before the shoot and was shooted the **number of ships** for opponent decreases by one. Ship is **sunk**

Others

I/O Handler and Utilities

- Debugging
- Modes (***Mode with borders***)
- I/O: boardStatus + input for initialization