

# SHOOTER GAME

## GAME LOGIC:

### MAIN GAME CLASS

#### FUNCTION CALLS:

- NEW ENEMY
- SHOW SCORE
- KEY DOWN
- KEY UP
- CHECK COLLISION
- CHECK PLAYER COLLISION

INIT

HEALTH  
SCORE  
LIFE

ENEMY SHIP ARRAY  
ENEMY BULLET ARRAY  
PLAYER SHIP  
PLAYER BULLET ARRAY

EVENT LISTENERS INPUT KEYBOARD  
EVENT LISTENERS ENTER FRAME / CHECK COLLISION  
EVENT LISTENERS. ENTER FRAME CHECK PLAYER COLLISION

## GLOBAL VARIABLES

- ENEMY SPAWN TIMER : TIMER
- PLAYER SHIP : PLANE SHIP → CLASS
- ENEMY SHIP : ARRAY ○ UP, DOWN, LEFT, RIGHT : BOOLEAN
- ENEMY BULLET : ARRAY
- PLAYER BULLET : ARRAY ○ PLAYER SCORE

NEW ENEMY

ENEMY SPAWN TIMER = NEW TIMER (INTERVALS, 1);

" . ADD EVENT LISTENER (TIMER EVENT. TIMER\_COMPLETE, NEW ENEMY SHIP);  
" . START ();

FUNCTION CALLS:  
○ NEW ENEMY SHIP

NEW ENEMY SHIP

COMING FROM DIRECTION (LEFT OR RIGHT)  
HOW FAR FROM TOP/BOTTOM BOUNDARY  
MOVEMENT SPEED  
ENEMY SHIP.ADDCHILD  
ENEMY SHIP ARRAY (ENEMY SHIP)

FUNCTION CALLS:  
○ NEW ENEMY

KEY DOWN

EVENT.keyCode

by event: KEYBOARD EVENT

FIRE PLAYER BULLET

→ LEFT 37  
→ RIGHT 39  
→ UP 38  
→ DOWN 40  
→ SPACE 32 → FIRE PLAYER BULLET } TRUE

KEY UP

EVENT.keyCode → FALSE

FIRE PLAYER BULLET

playerBullet.addCHILD  
playerBulletArray(playerBullet).push

ON LOCATION OF PLAYER SHIP .x.y & BULLET SPEED

↳ MAYBE SET IN CLASS

# SHOOT EM UP GAME

## GAME LOGIC

MAIN GAME CLASS  
CONTINUED

CALLED FROM ENEMY SHIP CLASS

**FIRE ENEMY BULLET**

FIRE MULTIPLE BULLETS AT INTERVAL  
ENEMY BULLET.ADDCHILD  
ENEMY BULLET ARRAY (ENEMY BULLET).PUSH  
ON LOCATION OF FIRING ENEMY SHIP

**CHECK PLAYER COLLISION**

SAME AS COLLISION CHECK

FUNCTION CALLS:

PLAYER SHIP HIT  
DELETE ENEMY BULLET

REVERSE LOOP OF ENEMY BULLET ARRAY

```
↳ IF (ENEMY PLAYER BULLET [i]. HIT TEST OBJECT . player SHIP)  
  {  
    player SHIP . player SHIP HIT ();  
    DELETE ENEMY BULLET ();  
    break;
```

**PLAYER SHIP HIT**

IN PLAYER SHIP CLASS

REMOVE PLAYER HEALTH

CHECK IF HEALTH < 0

↳ USE A LIFE

↳ PLAY EXPLODE ANIMATION → DELETE PLAYER SHIP ();

↳ IF LIFE = 0 ⇒ GAME OVER

↳ ELSE ADD PLAYER SHIP AGAIN

ELSE

↳ UPDATE HEALTH METER ();

VIA: MOVE (UP (PARENT)). UPDATE HEALTH ();

**REMOVE ENEMY BULLET**

( MOVE (UP (PARENT)). REMOVE ENEMY BULLET (THIS);

FUNCTION CALLS:

UPDATE HEALTH (SHIP HEALTH)

# SHOOTEmUp GAME

## GAME LOGIC

MAIN GAME CLASS  
CONTINUED

**COLLISION CHECK**

FUNCTION CALLS:

ENEMY SHIP HIT  
DELETE PLAYER BULLET  
SHOW SCORE

REVERSE  
IF  
LOOP & HITTEST OBJECT

↳ FIND THE BULLET NUMBER IN PLAYER BULLET ARRAY

↳ FIND THE ENEMY SHIP NUMBER IN ENEMY SHIP ARRAY

IF (playerBulletArray[i].hitTestObject, enemyShipArray[j])

enemyShip.enemyShipHit();

delete Player Bullet ();

enemyShip.score → playerScore

SHOW SCORE();

BREAK ON SUCCESS

**ENEMY SHIP HIT**

IN ENEMY SHIP CLASS



PLAYS A

EXPLODE ANIMATION

THEN CALLS A

REMOVE FROM ARRAY FUNCTION

FUNCTION

ON LAST FRAME:

parent.removeChild(this);

stop();

MovieClip (parent)

removeEnemyShip (this);

CLASS: SHIPHit function

↳ REMOVE ENEMY SHIP

= SAME AS REMOVE PLAYER BULLET

**REMOVE ENEMY SHIP**

by SHIP: ENEMY SHIP CLASS

LOOP ENEMY SHIP ARRAY

IF enemyShipArray[i] = ship

" . splice(i, 1)

break;

DELETES THE BULLET

AND REMOVE IT

FROM THE PLAYER BULLET ARRAY

FUNCTION CALLS:

MovieClip (parent). removePlayerBullet (this);

**REMOVE PLAYER BULLET**

by BULLET: PLAYER BULLET CLASS

LOOP PLAYER BULLET ARRAY

IF playerBulletArray[i] = bullet

bullet;

. splice(i, 1)

# SHOOT UP GAME

## GAME LOGIC

MAIN GAME CLASS

CONTINUED

**SHOW SCORE**

→ VIA PLACED TEXTFIELDS ON STAGE OF FLA

SHOW SCORE TEXT.TEXT = STRING ("SCORE: " + playerScore);

**GAME OVER**

→ REMOVE ALL OBJECTS ON STAGE

→ ALL ENEMY SHIPS - FOR ENEMYSHIPARRAY[i]. DELETE ENEMY SHIP ();  
IN ENEMYSHIPCLASS

SET ENEMYSHIPARRAY TO NULL;

→ PLAYER SHIP. DELETE PLAYER SHIP ();  
IN PLAYERSHIPCLASS

SET PLAYER SHIP TO NULL;

REMOVE EVENT LISTENER ON STAGE KEY UP / KEY DOWN  
" ENTER FRAME CHECK COLLISION

STOP THE ENEMY SPAWN TIMER. STOP ();  
" = NULL;

GO TO AND STOP ("GAME OVER"); SCREEN

# SHOOT EM UP GAME

## GAME LOGIC

MAIN GAME CLASS

CONTINUED

UPDATE HEALTH

→ VIA PLACED TEXTFIELDS ON STAGE OF FLA

```
SHOW SHIP HEALTH.TEXT = STRING("HEALTH: " + playerShip.health);
```

# SHOOTER GAME

- ① PLAYER SHIP CLASS
- ② PLAYER BULLET CLASS
- ③ ENEMY SHIP CLASS
- ④ ENEMY BULLET CLASS

① INIT : X, Y

SPEED → GLOBAL

INIT : EVENT LISTENER (ENTER FRAME, MOVE SHIP);

SHIP TIME → GLOBAL

MOVE SHIP TIME PASSED = getTimer() - SHIP TIME;

SHIP TIME += TIME PASSED;

VAR NEW X, NEW Y = this.x/y

IF (MOVIE CLIP (parent). LEFT ARROW ⇒ NEW X = - SPEED \* TIME PASSED / 1000  
- RIGHT ARROW ⇒ " = + "  
- UP ARROW ⇒ NEW Y = - "  
- DOWN ARROW ⇒ " = + "

BOUNDBRIES

STAGE 0/0 / WIDTH / HEIGHT ± 20

REPOSITION

this.x/y = NEW X/Y

# ShootEmUp GAME

## GAME OBJECTS:

playerShip

○ MOVES: UP/DOWN/LEFT/RIGHT

○ BOUNDARY: ALL SIDES

○ CURRENT WEAPON

○ SPAWN LOCATION

enemyShip

○ MOVES: LEFT/RIGHT

○ SPAWN LOCATION

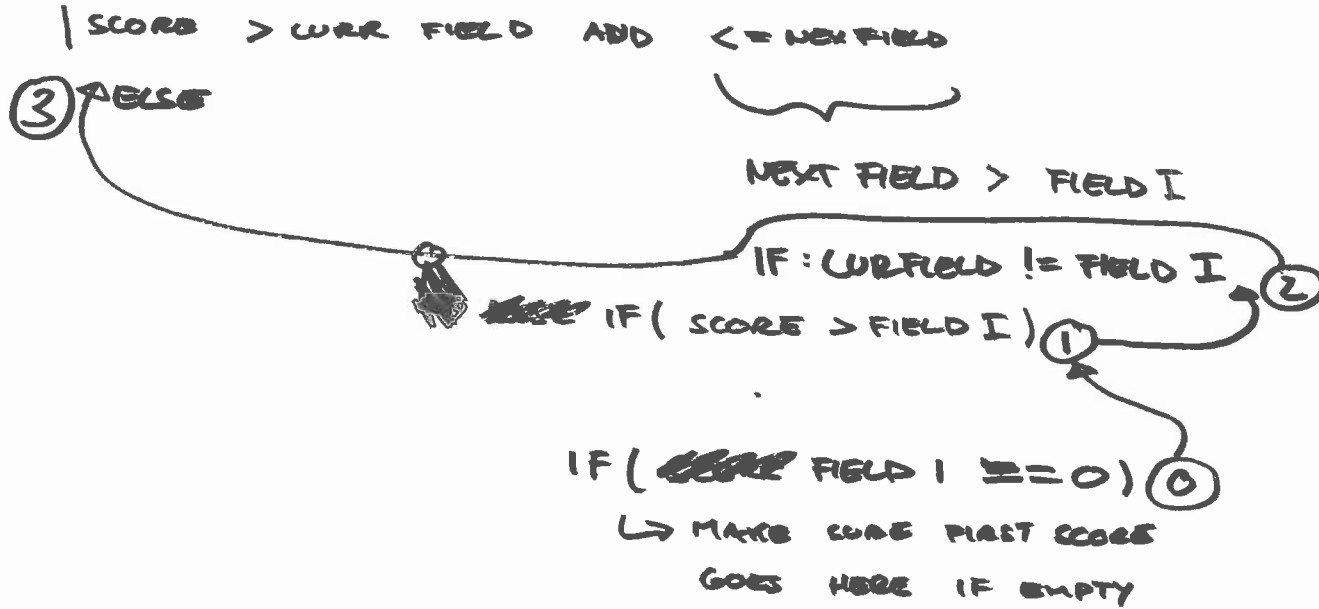
○ CURRENT WEAPON

bulletLaser

○ MOVES: UP/DOWN

○ SPAWN LOCATION

# SCORE PLACING



?

COPY

8	to	9
7	to	8
6	to	7
5	to	6
4	to	5
3	to	4
2	to	3
1	to	2
0	to	1

```

FOR J=1
UNTIL J==LAST
  J++
  (J+1=>)
  COPY 4 to 5
i=4: 5 to 6
      6 to 7
      7 to 8
      8 to 9
  
```

then PUT SCORE

FIND RANK POSITION → KEEP ⇔ SHIFT ALL LOWER AND SAME RANK ONE POSITION DOWN

GET PLAYER NAME ←

↓

BASED ON RANK  
PASTE THE NAME & SCORE

GET ALL SCORES FROM XML

→ ARRAY

→ NEW HIGHSCORE . PUSH

↓

ARRAY SORT

↓

REMOVE LAST  
VALUE

↓

GET INDEX OF SCORE ⇒ +1 = RANK