

## Modulo NewGame

**Descrizione del modulo:** Modulo per preparare il campo da gioco di una nuova partita

### INTERFACCIA

newGame.....	1
loadVerticalAxis .....	2
loadHorizontalAxis.....	4
buildPlayerNavy.....	5

### newGame

#### INPUT

Nome	Descrizione	Tipo	Vincoli
numFile	Numero file di salvataggio	Intero	numFile >= FIRST_SLOT AND numFile <= LAST_SLOT
c	Carattere letto da tastiera	Carattere	/

#### OUTPUT

Nome	Descrizione	Tipo	Vincoli
File	File di salvataggio della partita	File binario	/

#### LAVORO

Nome	Descrizione	Tipo	Vincoli
round	Round della partita	Round	/
whoPlay	Indica il player che attacca	Intero	1 = Attacca il player 1 2 = Attacca il player 2
turn	Indica il turno della partita	Intero	turn > 0
Pause	Indica lo stato di pausa	Intero	1 = Uscire dalla partita 0 = Continuare a giocare
id	Indica l'id del giocatore	Intero	1 = id player 1 2 = id player 2
player1	Primo player	Player	/
player2	Secondo player	Player	/
Map	Mappa di gioco	Array bidimensionale di caratteri	Dimensione 16x16 Caratteri consentiti:

			Da 'a' a 'o' per le navi '~' per il mare 'x' nave colpita '*' nave affondata
--	--	--	---

**ALGORITMO**

```

turn := 1
whoPlay := 1
pause := 0
round := setGameRound(round, turn)
round := setWhoPlay(round, whoPlay)
round := setPause(round, pause)
StampaAVideo("PIAZZA LE NAVI IL GIOCATORE 1")
id := 1
player1 := createPlayer(player1, id)
player1 := buildPlayerNavy(player1)
StampaAVideo("PLAYGROUND FINALE")
map := getPlayground(player1)
showMap(map)
StampaAVideo("Ora tocca al tuo avversario piazzare le navi!")
c := clickToContinue()
cleanScreen()
StampaAVideo("PIAZZA LE NAVI IL GIOCATORE 2")
id := 2
player2 := createPlayer(player2, id)
player2 := buildPlayerNavy(player2)
StampaAVideo("PLAYGROUND FINALE")
map := getPlayground(player2)
showMap(map)
StampaAVideo("La partita sta per cominciare!")
c := clickToContinue()
cleanScreen()
round := setActivePlayer(player1)
round := setPassivePlayer(player2)
playGame(round, numFile)

```

**loadVerticalAxis****INPUT:**

Nome	Descrizione	Tipo	Vincoli
------	-------------	------	---------

player	Giocatore a cui inserire una nave	Player	\
startColumn	Colonna di partenza della nave	intero	startColumn >= TABLE_MIN AND startColumn <= TABLE_MAX
startRow	Riga di partenza della nave	intero	startRow >= TABLE_MIN AND startRow <= TABLE_MAX
label	Etichetta della nave	carattere	label >= 'a' AND label <= 'o'
shipSize	Dimensioni della nave	intero	shipSize >= MIN_SHIP_SIZE AND shipSize <= MAX_SHIP_SIZE

**OUTPUT:**

Nome	Descrizione	Tipo	Vincoli
player	Giocatore aggiornato con all'interno del suo playground una nave	Player	\

**LAVORO:**

Nome	Descrizione	Tipo	Vincoli
i	Contatore utilizzato per inserire correttamente la nave nel suo range di coordinate	intero	i > 0
playground	Campo da gioco del giocatore player	Array di caratteri a due dimensioni	Dimensione 16x16 Caratteri consentiti: da 'a' o 'o' per le navi '~' per il mare 'x' per nave colpita '*' per nave affondata

**ALGORITMO:**

i := 1

playground := getPlayground(player)

MENTRE (i <= shipSize)

Carattere di playground situato nella riga startRow e colonna startColumn := label

startRow := startRow + 1

$i := i + 1$

FINE

player := setPlayground (player, playground)

### loadHorizontalAxis

#### INPUT:

Nome	Descrizione	Tipo	Vincoli
player	Giocatore a cui inserire una nave	Player	\
startColumn	Colonna di partenza della nave	intero	startColumn >= TABLE_MIN AND startColumn <= TABLE_MAX
startRow	Riga di partenza della nave	intero	startRow >= TABLE_MIN AND startRow <= TABLE_MAX
label	Etichetta della nave	carattere	label >= 'a' AND label <= 'o'
shipSize	Dimensioni della nave	intero	shipSize >= MIN_SHIP_SIZE AND shipSize <= MAX_SHIP_SIZE

#### OUTPUT:

Nome	Descrizione	Tipo	Vincoli
player	Giocatore aggiornato con all'intero del suo playground una nave	Player	\

#### LAVORO:

Nome	Descrizione	Tipo	Vincoli
i	Contatore utilizzato per inserire correttamente la nave nel suo range di coordinate	intero	$i > 0$
playground	Campo da gioco del giocatore player	Array di caratteri a due dimensioni	Dimensione 16x16 Caratteri consentiti: da 'a' a 'o' per le navi '~' per il mare

			'x' per nave colpita '*' per nave affondata
--	--	--	--

**ALGORITMO:**

i := 1

playground := getPlayground(player)

MENTRE (i <= shipSize)

    Carattere di playground situato nella riga startRow e colonna startColumn := label

    startColumn := startColumn + 1

    i := i + 1

FINE

player := setPlayground(player, playground)

**buildPlayerNavy****INPUT**

Nome	Descrizione	Tipo	Vincoli
player	Player a cui far inserire le navi nel suo playground	Player	/
MAX_SHIP_AMOUNT	Numero massimo di navi in un playground	Numero intero	15
MAX_SHIP_SIZE	Dimensione massima di una nave	Numero intero	5

**OUTPUT**

Nome	Descrizione	Tipo	Vincoli
player	Player con le navi inserite nel suo playground	Player	/

**LAVORO**

Nome	Descrizione	Tipo	Vincoli
i	Contatore del numero di navi da creare mancanti	intero	i > 0
j	Contatore del numero di navi da creare di dimensione sizeModifier	intero	j > 0
ship	Nave del giocatore	ship	/
shipDirection	Direzione della nave	carattere	V = Verticale O = Orizzontale

label	Etichetta della nave	carattere	label >= 'a' AND label <= 'o'
integerColumn	Colonna di partenza di una nave	numero intero	integerColumn >= TABLE_MIN AND integerColumn <= TABLE_MAX
integerRow	Riga di partenza di una nave	numero intero	integerRow >= TABLE_MIN AND integerRow <= TABLE_MAX
sizeModifier	Dimensioni della nave da inserire	numero intero	sizeModifier >= MIN_SHIP_SIZE AND sizeModifier <= MAX_SHIP_SIZE
amountModifier	Quantità di navi da inserire per una specifica dimensione	numero intero	amountModifier >= 1 AND amountModifier <= MAX_SHIP_AMOUNT
Index	Posizione che indica quale nave leggere	Numero Intero	index ≥ 1 AND index ≤ MAX_SHIP_AMOUNT
playground	Campo da gioco del giocatore player	Array di caratteri a due dimensioni	Dimensione 16x16 Caratteri consentiti: da 'a' a 'o' per le navi '~' per il mare 'x' per nave colpita '#' per nave affondata

**ALGORITMO**

sizeModifier := MAX\_SHIP\_SIZE

amountModifier := 1

index := 0

i := 1

j := 1

MENTRE (i <= MAX\_SHIP\_AMOUNT)

  j := 1

  MENTRE (j <= amountModifier)

    playground := getPlayground(player, playground)

    ship := createShip(sizeModifier, index, playground)

    player := setShip(player, (index + 1), ship)

    shipDirection := getDirection(ship)

    label := getLabel(ship)

## Modulo NewGame

```
integerColumn := pullColumn(ship)
integerRow := pullRow(ship)
SE (shipDirection = 'V')
  ALLORA
    player := loadVerticalAxis(player, integerColumn, integerRow, label, sizeModifier)
  ALTRIMENTI
    SE (shipDirection = 'O')
      ALLORA
        player := loadHorizontalAxis(player, integerColumn, integerRow, label, sizeModifier)
      FINE
    FINE
  j := j + 1
  i := i + 1
  index := index + 1
FINE
amountModifier := amountModifier + 1
sizeModifier := sizeModifier - 1
FINE
```