

Snake++: Datatypes and function flowcharts

09.11.2023

custom data types

struct config_t (config.h)

```
const int windowSize;

int mapWidth; //note: fixed 1:1 aspect ratio
int mapHeight;
int blockSizePx; //pixle size of one block
                calculated using windowSize and mapSize

int cycleDurationMs;
int difficulty; //0-3
int snakeDefaultLength;

const char * leaderboardFilename;

bool debug; //enable debug output
```

struct snake_t (snake.h)

```
int length;
int headX, headY;

snakeDirection_t direction;
int tail[MAX_MAP_SIZE * MAX_MAP_SIZE][2] = {0};

bool isAlive;
```

enum snakeDirection_t (snake.h)

```
enum snakeDirection_t {DOWN=0, UP, LEFT, RIGHT};
```

struct gameData_t (game.h)

```
snake_t snake;

SDL_Renderer *sdlRenderer
SDL_Window *sdlWindow

int mapCollisions[MAX_MAP_SIZE][MAX_MAP_SIZE];
portal_t mapPortals[MAX_PORTALS];
int mapPortalCount;

int foodX, foodY;
int lifesRemaining;

int timestampLastCycle;
bool isPaused;

gameState_t gameState;
```

enum gameState_t (game.h)

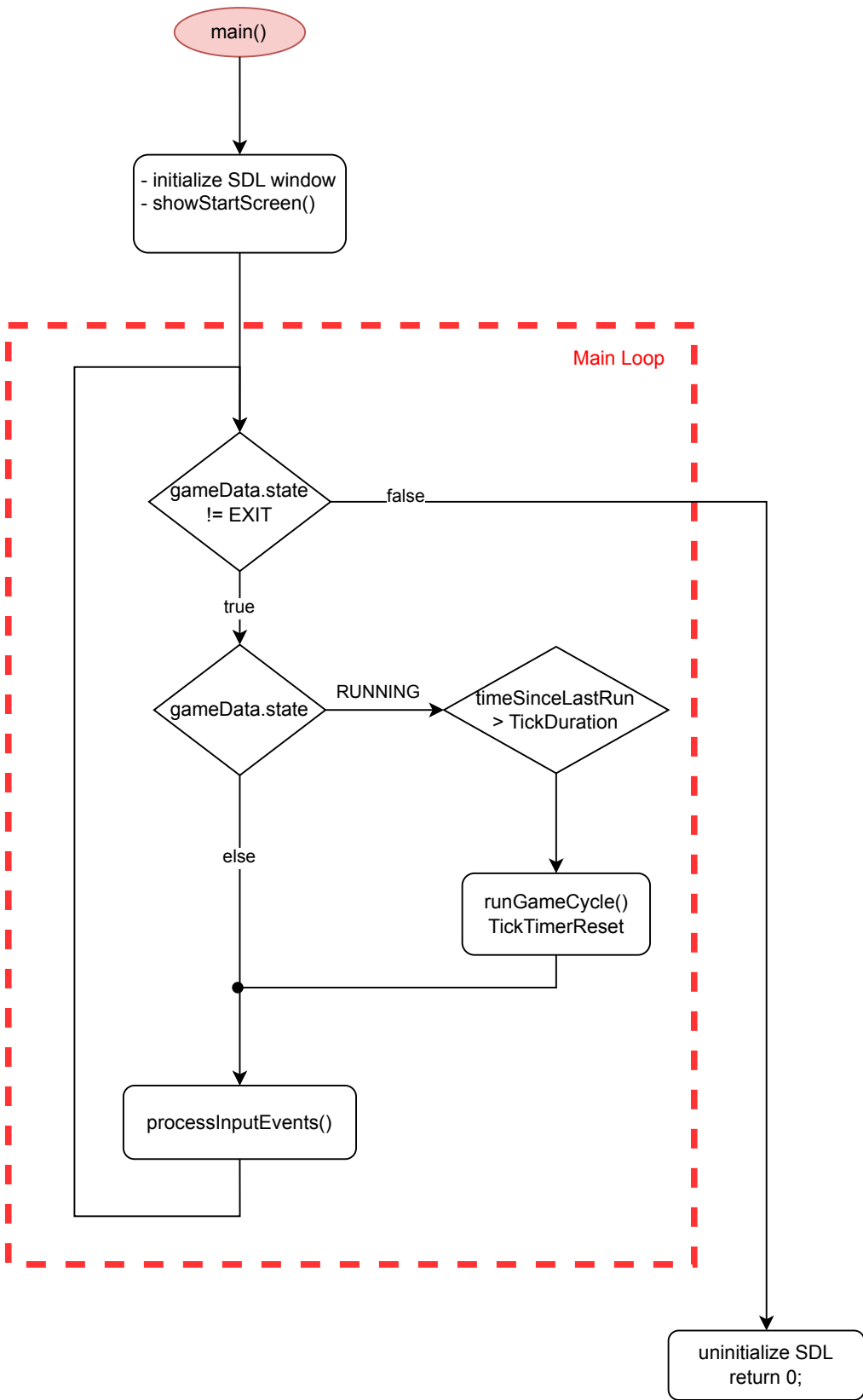
```
typedef enum gameState_t {PAUSED=0, MENU, RUNNING};
```

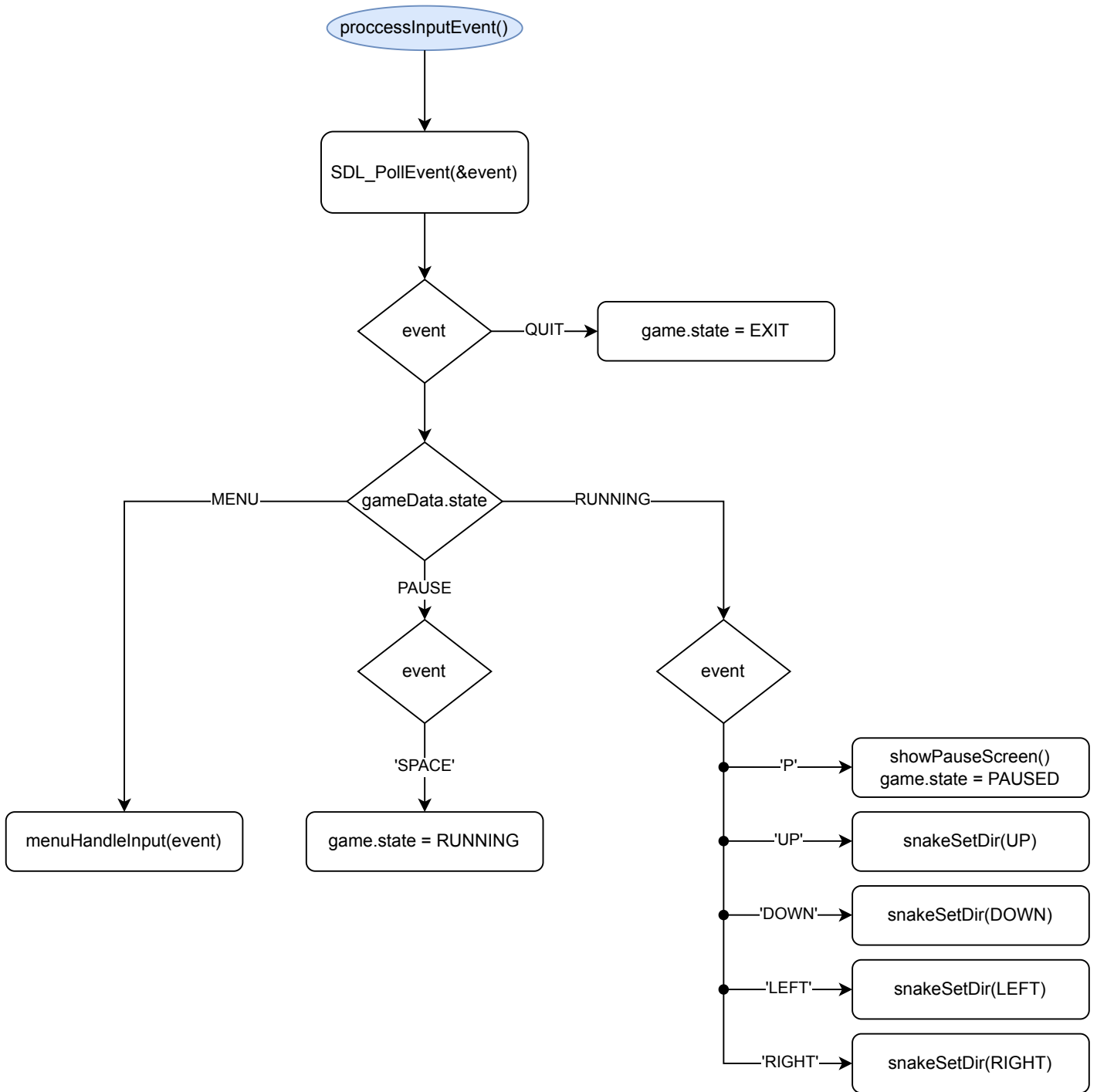
struct portal_t (game.h)

```
int posX;
int posY;

int targetX;
int targetY;

char * color;
```





menu.c

```
enum menus_t = {NONE=0, START, SETTINGS, LEADERBOARD, PAUSE}  
menus_t activeMenu = NONE
```

showStartScreen()

```
game.state = MENU  
activeMenu = START  
[show initial menu frame]
```

showLeaderboard()

```
game.state = MENU  
activeMenu = LEADERBOARD  
[show initial menu frame]
```

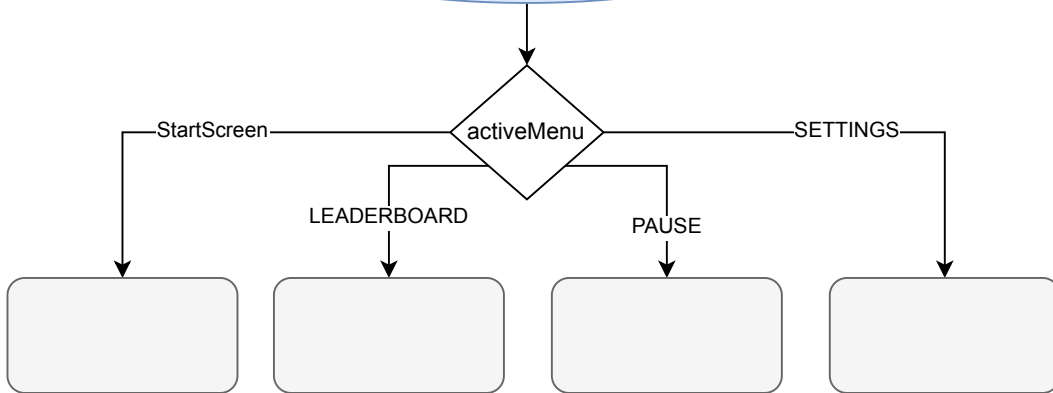
showSettings()

```
game.state = MENU  
activeMenu = SETTINGS  
[show initial menu frame]
```

showPauseScreen()

```
game.state = MENU  
activeMenu = PAUSE  
[show initial menu frame]
```

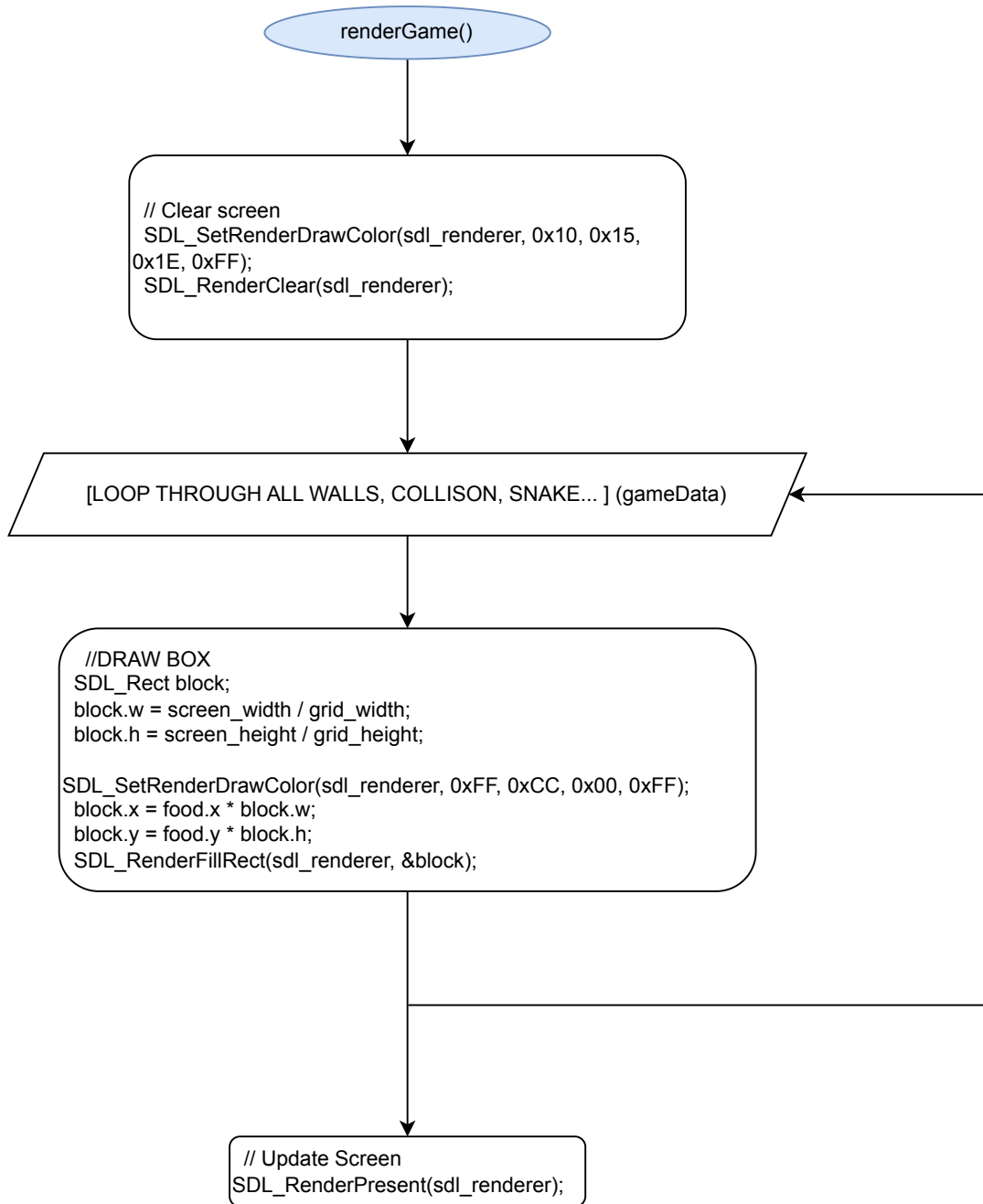
menuHandleInput(SDL_Event)

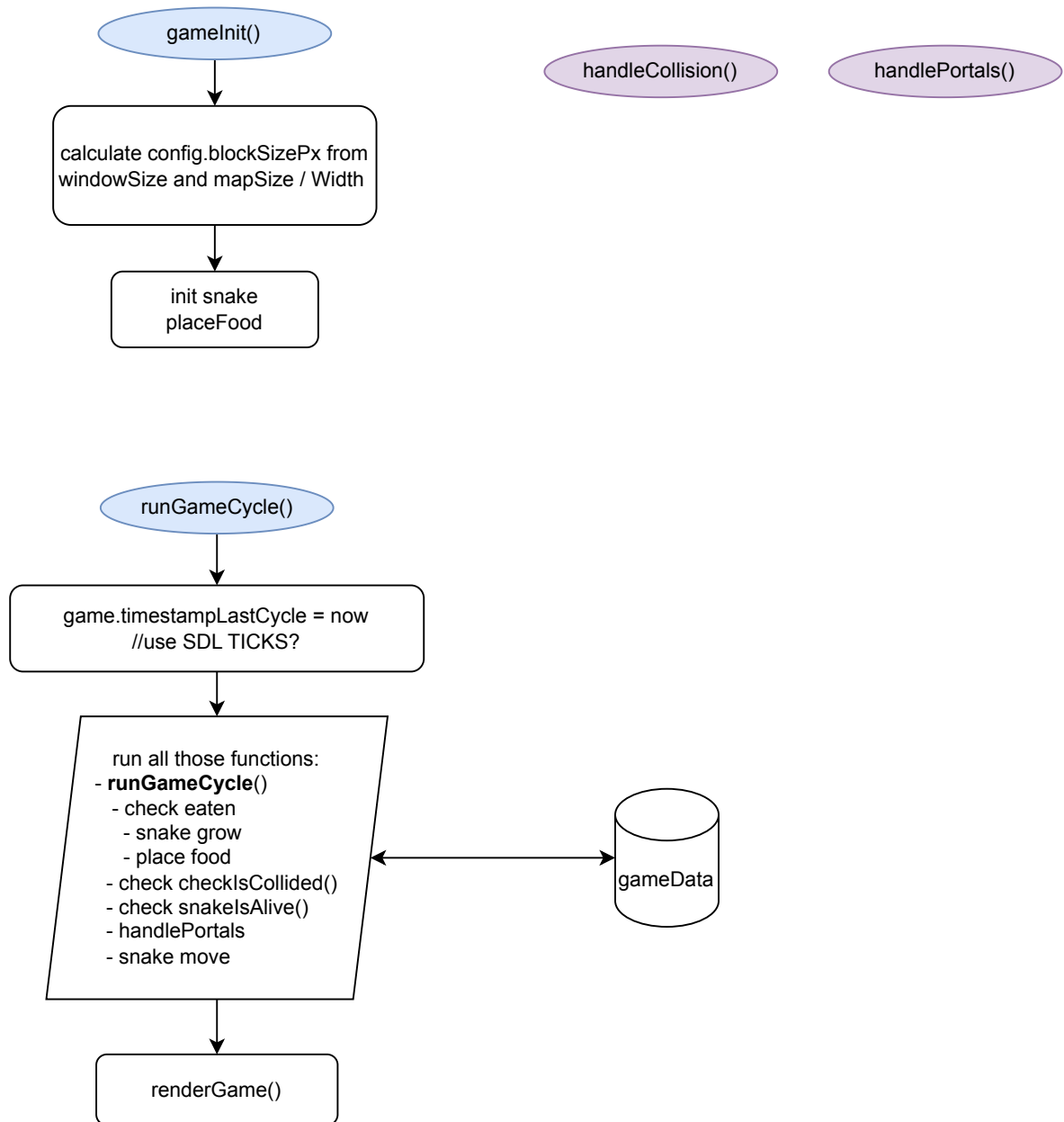


```
switch case for each used key:  
change menu item  
change value  
save to global config_t config
```

```
game.state = RUNNING
```

render.c





snake.c

snakeInit()

define start position
define start length

snakeSetDir()

snakeSetHeadPos()

```
snake.c  
- snakeInit()  
- snakeGrow()  
- snakeMove()  
- snakeSetDir(ENUM dir)  
- snakeSetHeadPos(x y)  
- snakeIsAlive()
```

snakeMove()

rotate array
enlarge array in current dir
remove last segment
...

update game.snake object

snakeGrow()

size++
...

