

Razrada projektnog zadatka sa zaduženjima članova tima

Za implementaciju 3D labirint igre razvit ćemo navedene strukture i funkcije koje će kako je naznačeno biti smještene u sljedeće fajlove:

Block.h: [Feđa – termin 2]

// Struktura bloka

struct _Block{ }; i njen konstruktor

// Varijabla za definisanje da li je zid ili staza

// Pokazivač na teksturu – jer nema razloga da trošimo memoriju svaki put kad inicijaliziramo novi blok ako se većina tekstura ponovno koristiti.

Color.h: [Feđa – termin 2]

// Struktura za boje

struct _Color{ }; i njen konstruktor

// Baggood displej ima 18-bitni color space. Svaka komponenta (R, G, B) je definisana sa 6 bita

// stoga koristimo char za smještanje vrijednosti. Dva najznačajnija bita ćemo ignorisati.

Definitions.h: [Amer – termin 2]

// Enumeracija pravca

enum _Direction { };

// Koristimo char za istinitosne vrijednosti jer se ne isplati trošiti resurse na int ili short

Game.h: [Feđa – termin 2]

// Struktura za igrača

struct _Player { }; i njen konstruktor

// Definira poziciju igrača u labirintu

// njegovu trenutnu poziciju unutar bloka

// te smjer u kojem igrač gleda (0-360)

// Struktura za igru

struct _Game { }; i njen konstruktor

// Posjeduje informacije veličine pojedinačnog bloka

// Prati informaciju da li je igrač došao do kraja labirinta

// Informacije o igraču držimo u odvojenoj strukturi, pošto se one uvijek zajedno koriste

Generator.h: [Amer – termin 2]

// Generiše labirint uz pomoć recursive backtracker algoritma i smješta ga u Grid strukturu

Grid generateMaze(int height, int width);

Graphics.h: [Amer – termin 2]

// Renderuje trenutnu scenu na osnovu pozicije igrača u svijetu i izgleda labirinta

void renderScene(Game *game, int fieldOfView, int width);

//Raycast vraća row-major koordinate tačke s kojom se sudario

int raycast(Grid *maze, Player *player, float angle);

Grid.h: [Feđa – termin 2]

// Struktura Grid-a

// Grid je osnovni koordinatni sistem koji određuje pozicije i karakteristike zidova ili manjka istih

struct _Grid{;

Movement.h: [Amer – termin 2]

// Uzima se vrijednost ulaznog uređaja I braća se koliko se treba pomjeriti po x i y koordinatama.

Movement getMovement();

Stack.h: [Feđa – termin 2]

// Definisanje strukture steka

// Služi za postavljanje steka na odgovarajuću vrijednosti

void stackInit(Stack *stack, int size);

//Praktično destruktor, služi da oslobodimo memoriju nakon što završimo sa stekom

void stackDestroy(Stack *stack);

//Osnovne operacije sa stekom

void push(Stack *stack, int element);

int pop(Stack *stack);

int peek(Stack *stack);

BOOL isEmpty(Stack *stack);

Texture.h: [Amer – termin 2]

// Definiše strukturu teksture

// Informacije o visini, širini u matričnoj reprezentaciji strukture

struct _Texture{;

// Postavljanje teksture iz mape boja i njenih dimenzija

Texture setTexture(Color *colorMap, int height, int width);