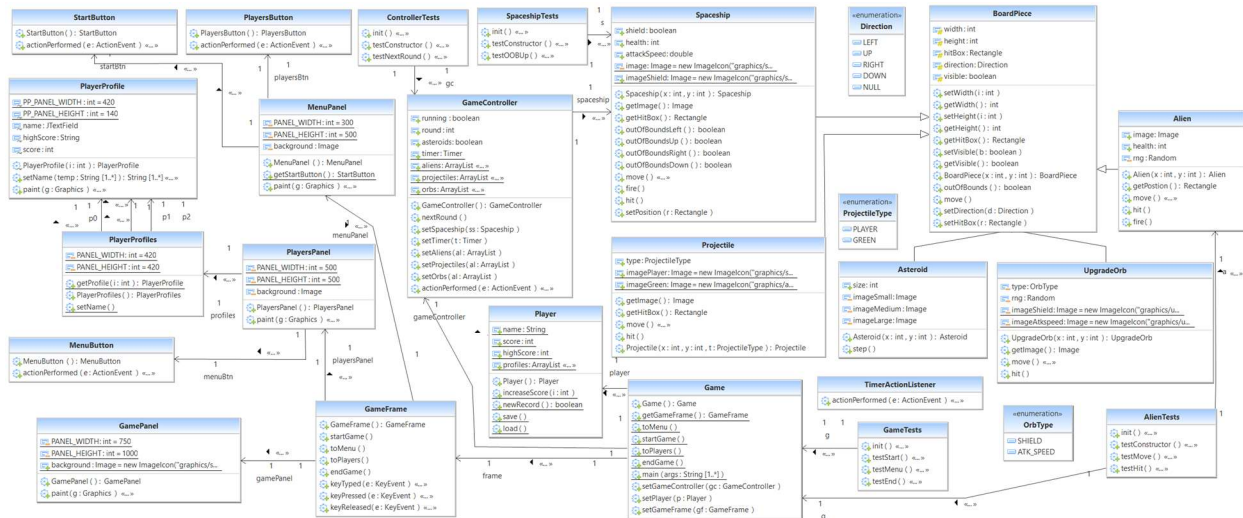


Prog3 házi dokumentáció

Class-diagram (with UML Lab Eclipse tool):



Függvények dokumentációja:

Megjegyzés: az alapvető setterek (void setter(param) { member = param; }) és getterek (returnType getter() { return member; }) nélkül.

Package Base:

Direction:

Enumeration, a szereplők lehetséges mozgási irányait nevezi meg.

LEFT, UP, RIGHT, DOWN, NULL

Game:

Adatok:

- static GameController gameController
- static Player player
- static GameFrame frame

Game():

Paraméter nélküli konstruktor.

static void toMenu():

A menübe lépésért felelős statikus függvény.

static void startGame():

A játék elindításáért felelős statikus függvény. Létrehozza az űrlényeket, beállítja a helyzetüket, átadja a GameController-nek, alaphelyzetbe állítja az űrhajót.

static void toPlayers():

A profilokat listázó menümenet hozza be.

static void endGame():

Leállítja a játékot, elszámolja a játékos pontjait, a GameController-ben maradt űrlényeket, lövedékeket és orb-okat eltávolítja.

static void main(String[] args):

A main függvény.

GameController implements ActionListener:

Adattagok:

- boolean running
- int round
- boolean asteroids
- static Timer timer
- static ArrayList<Alien> aliens
- static ArrayList<Projectile> projectiles
- static ArrayList<UpgradeOrb > orbs
- static Spaceship spaceship

GameController():

Paraméter nélküli konstruktor. A boolean adattagokat hamisra inicializálja, a round értéke 1 lesz, a timer 20 ms-ként lép, az űrhajó kezdőpontja (345; 900).

Void nextRound():

Növeli a round értékét 1-gyel.

void actionPerformed(ActionEvent e):

Override

Ha a startgombra kattintottak, akkor elindítja a timer-t.

class TimerActionListener implements ActionListener:

Belső osztály, a timer ActionListener-je.

void actionPerformed(ActionEvent e):

Override

A játék szereplőinek mozgását, tüzelését és sebződését valósítja meg, újrafesti ablakot minden alkalommal.

Player:

Adattagok:

- static String name
- static int score
- static int highScore
- static ArrayList<String[]> profiles

Player():

Paraméter nélküli konstruktor. Inicializálja profiles-t, meghívja a load() metódust.

static void increaseScore(int i):

A score értékét i-vel növeli.

static boolean newRecord():

Ha az aktuális pontszám nagyobb a játékos rekordjánál, átírja a highScore-t, majd beírja a profiles-ba. Ekkor visszatérési értéke igaz, különben hamis.

Static void save():

A profiles.txt fájlba kiírja a profiles tartalmát soronként. A sorok formátuma:

<name>:<highScore>.

Static void load():

A profiles.txt tartalmát betölti a profiles-ba.

Package BoardPieces:

Alien extends BoardPiece:

Adattagok:

- Image image
- int health
- Random rng

Alien(int x, int y):

Konstruktor, az ős konstruktorát meghívja x és y paraméterekkel, betölti az űrlény képét, beállítja az ős width és height adattagjait a kép megfelelő paramétereivel, a health értékét 2-re állítja.

Rectangle getHitBox():

Visszaadja az ős getHitBox() metódusa által kapott Rectangle-t.

void move():

Override

Használja a Direction-t. Az űrlény ősének iránya alapján módosítja az x és y koordinátákat, ha kilépne a játéktérrel, irányt vált.

LEFT: x+=6

UP: x-=6; y-=4

RIGHT: x-=6

DOWN: x+=6; y+=4

void hit():

Csökkenti a health-et 1-gyel, és ha a health értéke 0, akkor az űrlény láthatóságát hamisra állítja, és az rng következő véletlen egésze alapján 1:10 eséllyel létrehoz egy UpgradeOrb-ot az űrlény helyén.

void fire():

Az rng következő véletlen egésze alapján 1:500 eséllyel létrehoz egy ProjectileType.GREEN Projectile-t az űrlény helyén.

BoardPiece:

Adattagok:

- Int width
- Int height

- Rectangle hitBox
- Direction direction
- Boolean visible

BoardPiece(int x, int y):

Konstruktor, a hitBox x és y koordinátáját a paraméterül kapott x és y értékeknek állítja be.

boolean outOfBounds():

Ha a BoardPiece elhagyná a játékkeret, visszatérési értéke igaz, különben hamis.

void move():

Megfelelő irányban 6-tal változtatja a BoardPiece koordinátáit. Ha outOfBounds() igaz, ellentétes irányra vált.

OrbType:

Enumeration, az UpgradeOrb-ok lehetséges fajtáit nevezi meg.

SHIELD, ATK_SPEED

Projectile extends BoardPiece:

Adattagok:

- ProjectileType type
- static Image imagePlayer
- static Image imageGreen

Projectile(int x, int y, ProjectileType t):

Konstruktor, az ős konstruktorát meghívja x és y paraméterekkel, betölti az lövedék képeit, beállítja az ős width és height adattagjait a kép megfelelő paramétereivel, a lövedék típusa a t paraméter.

Image getImage():

A type alapján visszaadja a megfelelő Image-t.

Rectangle getHitBox():

Visszaadja az ős getHitBox() metódusa által kapott Rectangle-t.

void move():

Override

A type alapján mozgatja a lövedéket felfelé vagy lefelé. Ha a játékos lőtte ki, akkor felfelé, különben lefelé. Ha elhagyná a játékkeret, megszűnik (a GameController projectiles listájából remove() metódussal kikerül).

void hit():

A láthatóságot hamisra állítja.

ProjectileType:

Enumeration, a Projectile-ok lehetséges fajtáit nevezi meg.

PLAYER, GREEN

Spaceship extends BoardPiece:

Adattagok:

- boolean shield
- int health
- double attackSpeed
- static Image image
- static Image imageShield

Spaceship(int x, int y):

Konstruktor, az ős konstruktorát meghívja x és y paraméterekkel, betölti az űrhajó képeit, beállítja az ős width és height adattagjait a kép megfelelő paramétereivel, az health értéke 3, a shield false.

Image getImage():

Ha a shield igaz, visszaadja az imageShield-et, különben image-t.

void setHitBox(Rectangle r):

Az ős setHitBox(Rectangle r) metódusát hívja meg r paraméterrel.

Rectangle getHitBox():

Visszaadja az ős getHitBox() metódusa által kapott Rectangle-t.

boolean outOfBoundsLeft():

Ha az űrhajó balra elhagyná a játékteret, igazzal tér vissza, különben hamissal.

boolean outOfBoundsUp():

Ha az űrhajó felfelé elhagyná a játékteret, igazzal tér vissza, különben hamissal.

boolean outOfBoundsRight():

Ha az űrhajó jobbra elhagyná a játékteret, igazzal tér vissza, különben hamissal.

boolean outOfBoundsDown():

Ha az űrhajó lefelé elhagyná a játékteret, igazzal tér vissza, különben hamissal.

void move():

Override

Ha az űrhajó nem hagyja el a játékteret, akkor megfelelő irányba mozgatja 10-zel.

void fire():

A GameController projectiles listájához hozzáad egy PLAYER típusú lövedéket az űrhajó orránál található ponton.

void hit():

Ha shield igaz, hamisra állítja, különben csökkenti a health értékét 1-gyel. Ha health 0, meghívja a Game.endGame() metódust.

UpgradeOrb extends BoardPiece:

Adattagok:

- OrbType type
- Random rng
- static Image imageShield
- static Image imageAtkspeed

UpgradeOrb(int x, int y):

Konstruktor, az űs konstruktorát meghívja x és y paraméterekkel, az rng következő véletlen egésze alapján 2:3 valószínűséggel a type-ot OrbType.SHIELD-re állítja, 1:3 valószínűséggel pedig OrbType.ATK_SPEED-re, beállítja az űs width és height adattagjait a megfelelő kép megfelelő paramétereivel.

Image getImage():

Visszaadja a type-nak megfelelő Image adattagját.

void move():

Override

Lefelé mozgatja az orb-ot 2-vel, ha elhagyná a játékteret, a láthatóságát hamisra állítja.

void hit():

A láthatóságát hamisra állítja. Ha a type SHIELD, akkor az űrhajón meghívja a setShield(boolean b) metódust true paraméterrel.

Package GUI:

GameFrame extends JFrame implements KeyListener:

Adattagok:

- MenuPanel menuPanel
- GamePanel gamePanel
- PlayersPanel playersPanel

GameFrame():

Paraméter nélküli konstruktor, inicializálja az adattagokat, valamint egy WindowListener-t ad az ablakhoz, melynek a windowClosing(WindowEvent e) metódusa menti a játékos profilját, majd bezárja az ablakot.

void startGame():

Hozzáadja az ablakhoz a gamePanel-t és elindítja a GameController timer-jét.

void toMenu():

Hozzáadja az ablakhoz a menuPanel-t.

void toPlayers():

Hozzáadja az ablakhoz a playerPanel-t.

void endGame():

Hozzáadja az ablakhoz a menuPanel-t és leállítja a GameController timer-jét.

void keyPressed(KeyEvent e):

Override

A nyilak lenyomására beállítja az űrhajó irányát.

void keyReleased(KeyEvent e):

Override

A nyilak felengedésére Direction.NULL-ra állítja az űrhajó irányát. A space felengedésére meghívja az űrhajó fire() metódusát.

GamePanel extends JPanel:

Adattagok:

- static final int PANEL_WIDTH = 750
- static final int PANEL_HEIGHT = 1000
- static Image background

GamePanel():

Paraméter nélküli konstruktor.

void paint(Graphics g):

Override

Kirajzolja a játék szereplőit.

MenuButton extends JButton implements ActionListener:

MenuButton():

Paraméter nélküli konstruktor, a gomb felirata MENU, mérete (70, 26).

void actionPerformed(ActionEvent e):

Meghívja a PlayerProfiles.setName(), Player.save() és Game.toMenu() metódusokat.

MenuPanel extends JPanel:

Adattagok:

- static final int PANEL_WIDTH = 300
- static final int PANEL_HEIGHT = 500
- static Image background
- JPanel btnHolder
- Static StartButton startBtn
- Static PlayersButton playersBtn

MenuPanel():

Paraméter nélküli konstruktor, hozzáadja az adattagként tárolt elemeket a panelhez.

void paint(Graphics g):

Override

Feketére festi a hátteret.

PlayerProfiles extends JPanel:

Adattagok:

- static final int PANEL_WIDTH = 420
- static final int PANEL_HEIGHT = 420
- static PlayerProfile p0
- static PlayerProfile p1
- static PlayerProfile p2

PlayerProfiles():

Paraméter nélküli konstruktor, hozzáadja az adattagként tárolt elemeket a panelhez.

class PlayerProfile extends JPanel:

Belső osztály, tárol egy, az adott profilhoz tartozó nevet kiíró JTextField-et, és kiírja a profilhoz tartozó rekord pontszámot HIGHSCORE: <highscore> formátumban.

String[] setName(String[] temp):

A paraméter 0 indexű elemét a JTextField szövegére írja át és visszaadja.

void paint(Graphics g):

A háttérrel sötétszürkére festi, a betűtípust Consolas-ra állítja 20-as betűmérettel, kirajzolja a highscore-t.

static PlayerProfile getProfile(int i):

Visszaadja a pi adattagot (p0, p1 vagy p2, ahol i = 0, 1 vagy 2).

PlayersButton extends JButton implements ActionListener:

PlayersButton():

Paraméter nélküli konstruktor, a gomb felirata PLAYERS, mérete (87, 26).

void actionPerformed(ActionEvent e):

Meghívja a Game.toPlayers() metódust.

PlayersPanel extends JPanel:

Adattagok:

- static final int PANEL_WIDTH = 500
- static final int PANEL_HEIGHT = 500
- PlayerProfile-es profiles
- JButton menuBtn

PlayersPanel():

Paraméter nélküli konstruktor, hozzáadja az adattagként tárolt elemeket a panelhez.

void paint(Graphics g):

Feketére festi a háttérrel.

StartButton extends JButton implements ActionListener:

StartButton():

Paraméter nélküli konstruktor, a gomb felirata START GAME, mérete (108, 26).


```
void actionPerformed(ActionEvent e):  
    Meghívja a Game.startGame() metódust.
```