Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Алтайский государственный университет»

Школа развития цифровых компетенций «Digital Up» (цифровая кафедра)

Отчет о выполнении производственной практики

Кейс 3. Разработать систему навыков

Исполнители:

Бракк Артём,  
Семенков Никита,

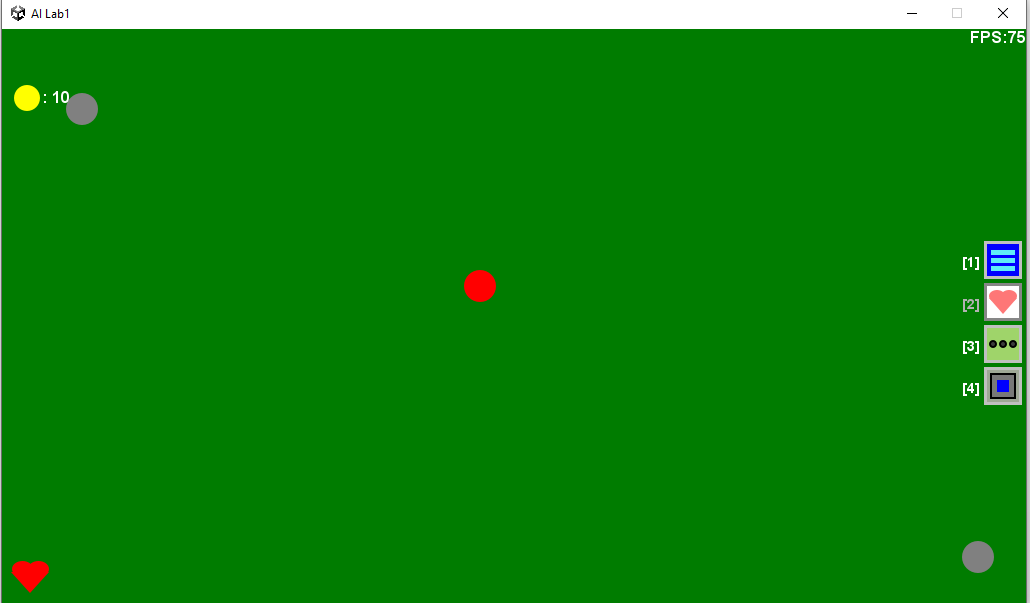
Берлова Ксения

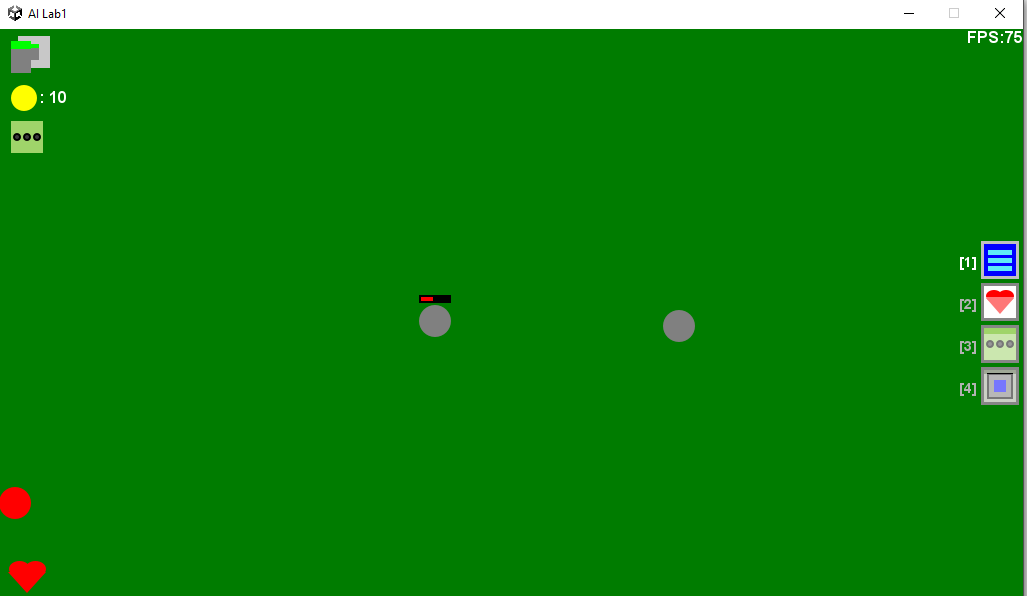
г. Барнаул, 2025

Результаты проекта:

В игре есть 4 навыка, которые можно купить. Персонаж игрока может купить до 4 выбранных навыков. Каждый навык активируется по нажатию кнопок 1-4. Каждое применение навыка отправляет его на перезарядку. Чем больше навыков перезаряжается, тем больше время перезарядки каждого последующего применяемого навыка. После 10 применений навык становится элитным – его время перезарядки сокращается вдвое, а эффект усиливается в три раза. В начале следующего уровня элитный статус сбрасывается. После прохождения уровня игрок может взаимодействовать с торговцем. Торговец предлагает предметы, которые гарантируют появление навыка соответствующей категории.







**Приложение 1.**Программный код

public class Game {

public static final Game game = new Game();

public static JFrame window = new JFrame("AI Lab1");

public static Canvas canvas = new Canvas();

public static int globalFps = 0;

public static int enemyCount = 2;

public static final int W = 1024;

public static final int H = W / 16 \* 9;

public static boolean fpsRender = true;

public static boolean debugRender = false;

public static final byte EASY = (byte)0;

public static final byte MEDIUM = (byte)1;

public static final byte HARD = (byte)2;

public static byte difficulty = EASY;

public static int coins = 90;

public static LinkedList<Skill> buyingSkills = new LinkedList<>();

public static final String MAIN\_MENU\_WORLD\_NAME = "main\_menu";

public static final String MAIN\_PLAYABLE\_WORLD\_NAME = "main\_playable\_world";

public static void main(final String[] args) throws Exception {

window.setDefaultCloseOperation(3);

window.setResizable(false);

canvas.setPreferredSize(new Dimension(W, H));

window.getContentPane().add(canvas);

window.pack();

window.setVisible(true);

window.setLocationRelativeTo(null);

window.setIconImage(ImageIO.read(new java.io.File("res/icon.png")));

final Toolkit toolkit = Toolkit.getDefaultToolkit();

window.setCursor(toolkit.createCustomCursor(

toolkit.getImage("res/cursor\_m.png"),

new Point(0, 0),

""

));

canvas.addMouseListener(new MouseAdapter() {

@Override

public void mousePressed(final MouseEvent event) {

game.mousePressed(event);

}

@Override

public void mouseReleased(final MouseEvent event) {

game.mouseReleased(event);

}

});

canvas.addMouseMotionListener(new MouseMotionAdapter() {

@Override

public void mouseMoved(final MouseEvent event) {

game.mouseMoved(event);

}

});

canvas.addKeyListener(new KeyAdapter(){

@Override

public void keyPressed(final KeyEvent event) {

game.keyPressed(event);

}

@Override

public void keyReleased(final KeyEvent event) {

game.keyReleased(event);

}

});

canvas.requestFocus();

canvas.setFocusable(true);

new Thread(() -> {

double unprocessed = 0d;

long oldTime = System.nanoTime();

long nowTime;

final double targetTime = 1000d / 75d;

game.init();

long millis = System.currentTimeMillis();

int frames = 0;

while(true) {

nowTime = System.nanoTime();

unprocessed += (double)(nowTime - oldTime) / 1\_000\_000d;

oldTime = nowTime;

while (unprocessed >= targetTime) {

game.tick();

game.doDraw();

frames ++;

unprocessed -= targetTime;

}

if (System.currentTimeMillis() - millis >= 1000) {

//window.setTitle("AI Lab 1 FPS:[" + frames + "]");

globalFps = frames;

frames = 0;

millis += 1000;

}

}

}).start();

}

private void doDraw() {

final BufferStrategy strategy = canvas.getBufferStrategy();

if (strategy == null) {

canvas.createBufferStrategy(3);

return;

}

final Graphics2D g = (Graphics2D)strategy.getDrawGraphics();

g.setColor(Color.GREEN.darker().darker());

g.fillRect(0, 0, W, H);

g.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

draw(g);

g.dispose();

strategy.show();

}

private void init() {

WorldManager.putWorld(MAIN\_MENU\_WORLD\_NAME, new MenuWorld());

WorldManager.putWorld(MAIN\_PLAYABLE\_WORLD\_NAME, new PlayableWorld());

WorldManager.setWorld(MAIN\_MENU\_WORLD\_NAME);

}

private void draw(final Graphics2D g) {

WorldManager.world.ifPresent((world) -> world.draw(g));

}

private void tick() {

WorldManager.world.ifPresent(World::tick);

}

public void mousePressed(final MouseEvent event) {

WorldManager.world.ifPresent((world) -> world.mousePressed(event));

}

public void mouseReleased(final MouseEvent event) {

WorldManager.world.ifPresent((world) -> world.mouseReleased(event));

}

public void mouseMoved(final MouseEvent event) {

WorldManager.world.ifPresent((world) -> world.mouseMoved(event));

}

public void keyPressed(final KeyEvent event) {

if (event.getKeyCode() == KeyEvent.VK\_F4) { Game.fpsRender = !Game.fpsRender; }

if (event.getKeyCode() == KeyEvent.VK\_F3) { Game.debugRender = !Game.debugRender; }

WorldManager.world.ifPresent((world) -> world.keyPressed(event));

}

public void keyReleased(final KeyEvent event) {

WorldManager.world.ifPresent((world) -> world.keyReleased(event));

}

}

**Приложение 2.**

Ссылка на презентацию: https://drive.google.com/file/d/1V-tKqw4JfrhJt4Hjph-L3l6lwlNM7o-\_/view?usp=sharing

**Приложение 3.**

Ссылка на файл программы: https://drive.google.com/file/d/1GQHvg3PyUaga8wGrV51Fli1MnLS\_Q6rR/view?usp=sharing