ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Отчет по лабораторной работе №4

по курсу «Профессиональный практикум программной инженерии»

на тему «Разрешение конфликтов объединения веток»

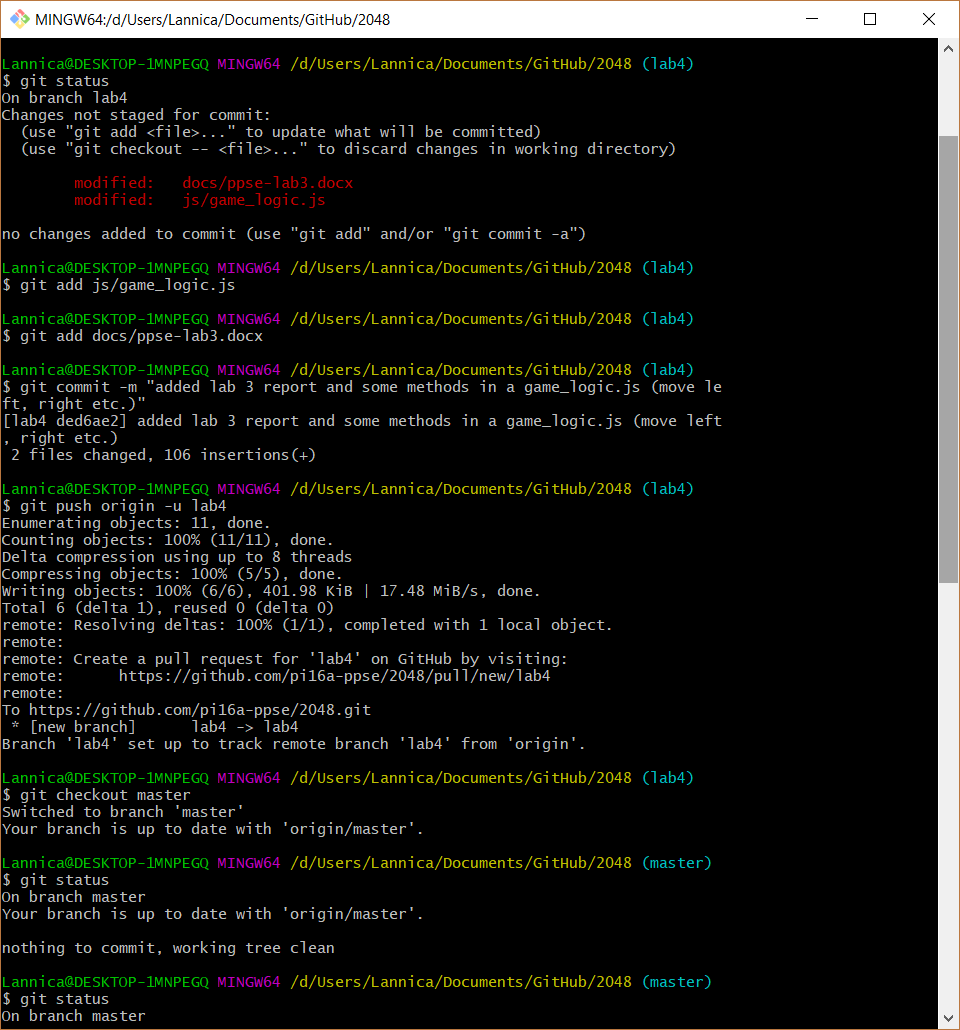
Выполнил:

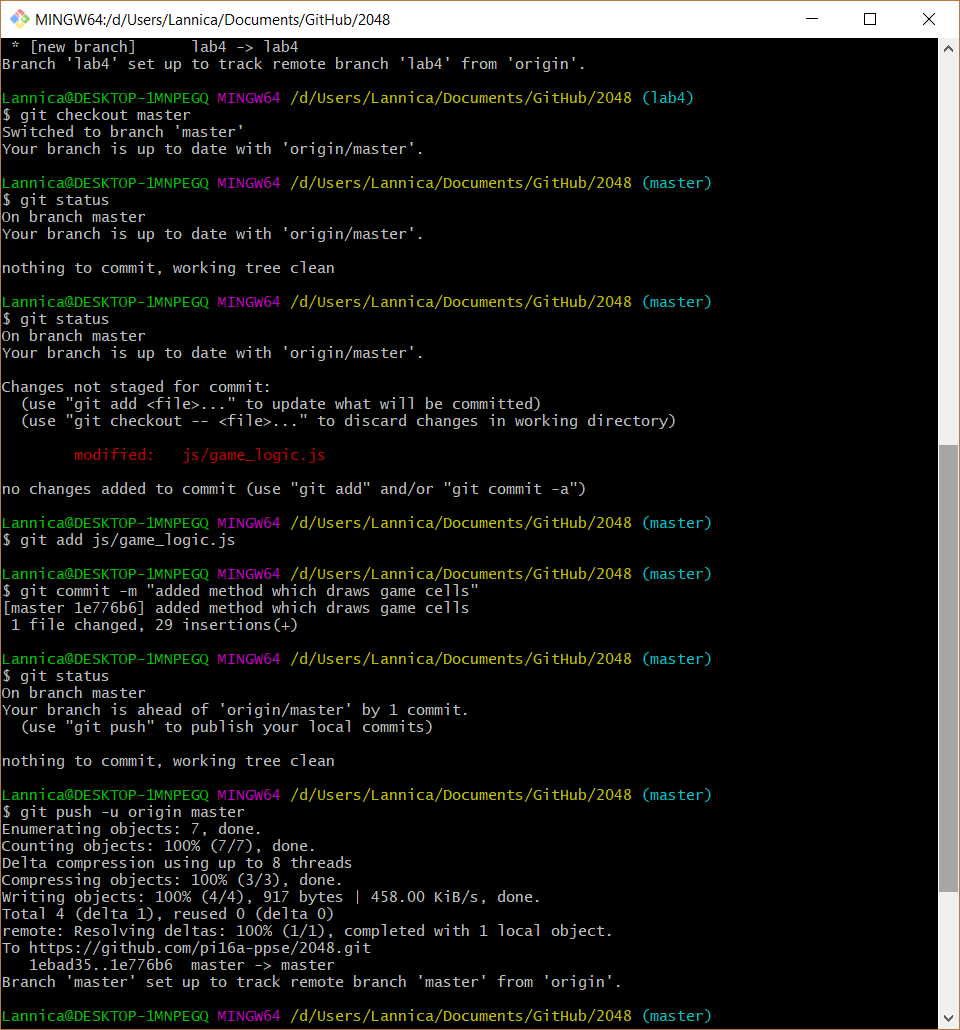
Ст. гр. ПИ-16а

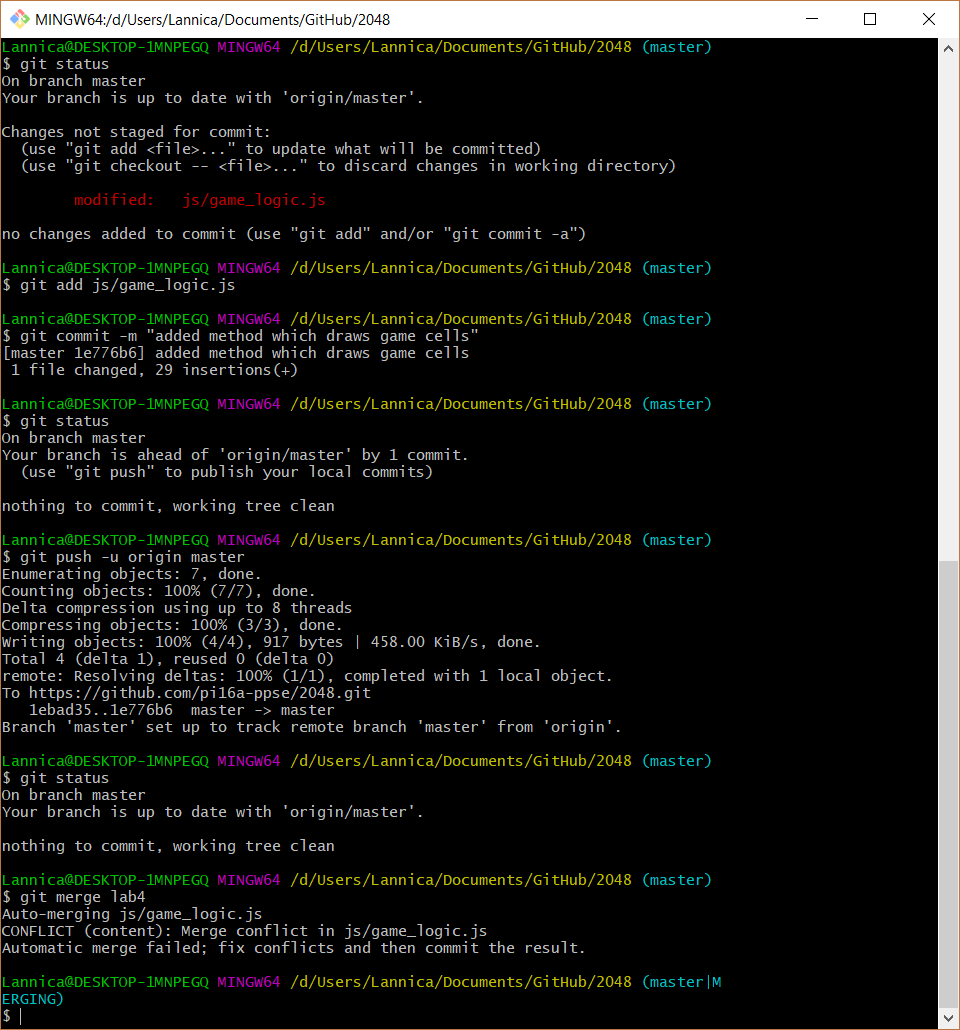
Бердюкова С.С.

Донецк 2019

Цель лабораторной работы: научиться решать конфликтные ситуации в Git, возникающие после объединения веток.







Конфликт в файле:

<<<<<<< HEAD

function drawCell(cell) {

ctx.beginPath();

ctx.rect(cell.x, cell.y, width, width);

switch (cell.value){

case 0 : ctx.fillStyle = '#A9A9A9'; break;

case 2 : ctx.fillStyle = '#D2691E'; break;

case 4 : ctx.fillStyle = '#FF7F50'; break;

case 8 : ctx.fillStyle = '#ffbf00'; break;

case 16 : ctx.fillStyle = '#bfff00'; break;

case 32 : ctx.fillStyle = '#40ff00'; break;

case 64 : ctx.fillStyle = '#00bfff'; break;

case 128 : ctx.fillStyle = '#FF7F50'; break;

case 256 : ctx.fillStyle = '#0040ff'; break;

case 512 : ctx.fillStyle = '#ff0080'; break;

case 1024 : ctx.fillStyle = '#D2691E'; break;

case 2048 : ctx.fillStyle = '#FF7F50'; break;

case 4096 : ctx.fillStyle = '#ffbf00'; break;

default : ctx.fillStyle = '#ff0080';

}

ctx.fill();

if (cell.value) {

fontSize = width / 2;

ctx.font = fontSize + 'px Arial';

ctx.fillStyle = 'white';

ctx.textAlign = 'center';

ctx.fillText(cell.value, cell.x + width / 2, cell.y + width / 2 + width/7);

}

=======

function moveRight () {

var i, j;

var coll;

for(i = 0; i < size; i++) {

for(j = size - 2; j >= 0; j--) {

if(cells[i][j].value) {

coll = j;

while (coll + 1 < size) {

if (!cells[i][coll + 1].value) {

cells[i][coll + 1].value = cells[i][coll].value;

cells[i][coll].value = 0;

coll++;

} else if (cells[i][coll].value == cells[i][coll + 1].value) {

cells[i][coll + 1].value \*= 2;

score += cells[i][coll + 1].value;

cells[i][coll].value = 0;

break;

} else {

break;

}

}

}

}

}

pasteNewCell();

}

function moveLeft() {

var i, j;

var coll;

for(i = 0; i < size; i++) {

for(j = 1; j < size; j++) {

if(cells[i][j].value) {

coll = j;

while (coll - 1 >= 0) {

if (!cells[i][coll - 1].value) {

cells[i][coll - 1].value = cells[i][coll].value;

cells[i][coll].value = 0;

coll--;

} else if (cells[i][coll].value == cells[i][coll - 1].value) {

cells[i][coll - 1].value \*= 2;

score += cells[i][coll - 1].value;

cells[i][coll].value = 0;

break;

} else {

break;

}

}

}

}

}

pasteNewCell();

}

function moveUp() {

var i, j, row;

for(j = 0; j < size; j++) {

for(i = 1; i < size; i++) {

if(cells[i][j].value) {

row = i;

while (row > 0) {

if(!cells[row - 1][j].value) {

cells[row - 1][j].value = cells[row][j].value;

cells[row][j].value = 0;

row--;

} else if (cells[row][j].value == cells[row - 1][j].value) {

cells[row - 1][j].value \*= 2;

score += cells[row - 1][j].value;

cells[row][j].value = 0;

break;

} else {

break;

}

}

}

}

}

pasteNewCell();

}

function moveDown() {

var i, j, row;

for(j = 0; j < size; j++) {

for(i = size - 2; i >= 0; i--) {

if(cells[i][j].value) {

row = i;

while (row + 1 < size) {

if (!cells[row + 1][j].value) {

cells[row + 1][j].value = cells[row][j].value;

cells[row][j].value = 0;

row++;

} else if (cells[row][j].value == cells[row + 1][j].value) {

cells[row + 1][j].value \*= 2;

score += cells[row + 1][j].value;

cells[row][j].value = 0;

break;

} else {

break;

}

}

}

}

}

pasteNewCell();

>>>>>>> lab4

}

