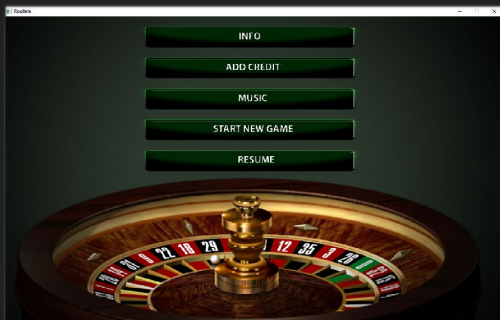


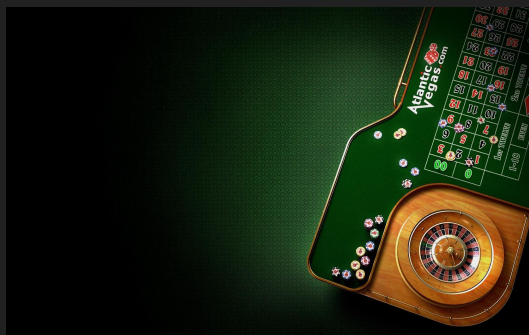
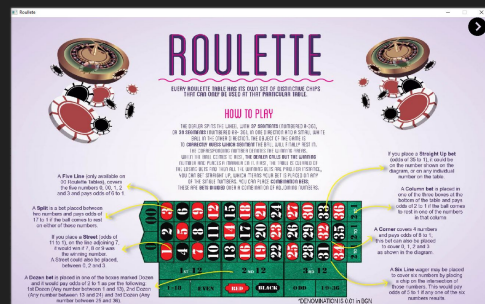
РУЛЕТКА



Направена от
Виктор Цонев
Васил Какевски
Петър Ефтимов
Росица Маджарова



ЕКРАНИ

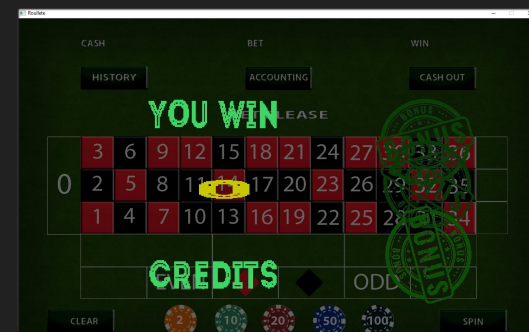


Основни екрани в играта

Интро

Инфо

Игра



Печалба

Изходен

Бонус



ЕКРАНИ

Допълнителни екрани

Въртяща се рулетка

История

Счетоводство



ИНТРО



БУТОНИ:

Инфо

Добави кредити

Музика

Старт

Зареди игра

```
class IntroScreen
```

```
{
```

```
public:
```

```
    IntroScreen();
```

```
    virtual ~IntroScreen();
```

```
    bool Draw();
```

```
    bool Clear();
```

```
    bool getFlag(){return isActive;}
```

```
    void IntroScreenShowCredits(Credits& credits);
```

```
public:
```

```
    LTexture* introBackground;
```

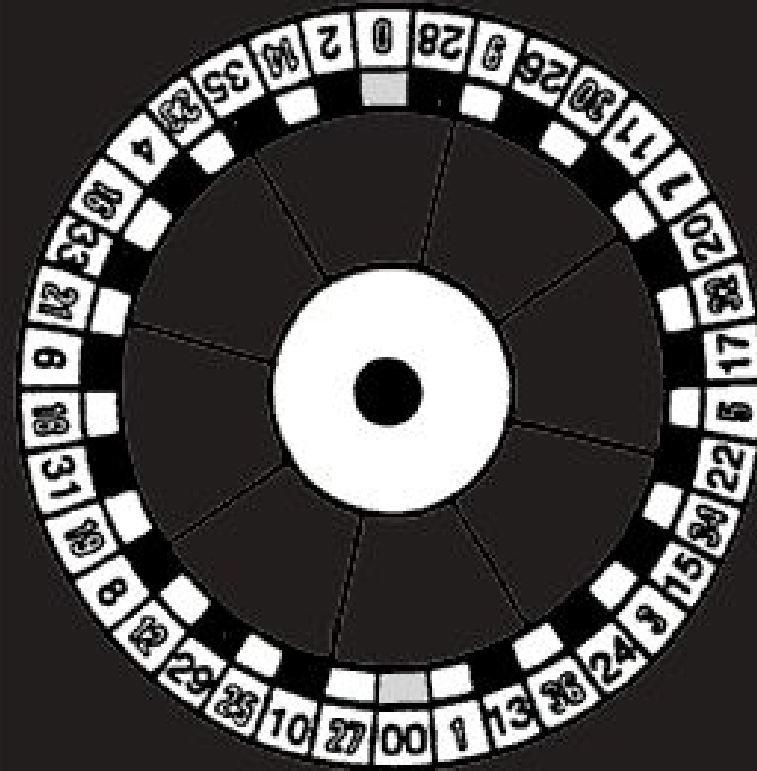
```
    Button* introButtons[INTRO_BUTTONS];
```

```
    //bool for mouseEvents
```

```
    bool isActive;
```

```
};
```

class IntroScreen



class IntroScreen

```
void
IntroScreen::IntroScreenShowCredits(Credits&
credits)
{
    Clear();
    Draw();
    if (credits.GetCredit())
    {
        Text textCredit(SCREEN_W / 2 - 500 / 2 + 50 +
470,
        SCREEN_H / 10 - INTRO_BUTTONS_H / 2
        + (INTRO_BUTTONS_H + 10)
        , 230, 40, 20, "Credits: ", { 100,
        200, 100, 255 });

        Text textCreditsNumber(
        SCREEN_W / 2 - 500 / 2 + 50 + 690,
        SCREEN_H / 10 - INTRO_BUTTONS_H / 2
        + (INTRO_BUTTONS_H + 10)
        , 60, 40, 20, credits.GetCredit(),
        { 100, 200, 100 });
    }
}
```

ИНФО

ROULETTE

EVERY ROULETTE TABLE HAS ITS OWN SET OF DISTINCTIVE CHIPS THAT CAN ONLY BE USED AT THAT PARTICULAR TABLE.

HOW TO PLAY

THE DEALER SPINS THE WHEEL WITH 37 SEGMENTS (NUMBERED 0-36), OR 38 SEGMENTS (NUMBERED 00-36), IN ONE DIRECTION AND A SMALL WHITE BALL IN THE OTHER DIRECTION. THE OBJECT OF THE GAME IS TO CORRECTLY GUESS WHICH SEGMENT THE BALL WILL FINALLY REST IN. THE CORRESPONDING NUMBER DENOTES THE WINNING AREAS. WHEN THE BALL COMES TO REST, THE DEALER CALLS OUT THE WINNING NUMBER AND PLACES A MARKER ON IT. FIRST, THE TABLE IS CLEARED OF THE LOSING BETS AND THEN ALL THE WINNING BETS ARE PAID. FOR INSTANCE, YOU CAN BET STRAIGHT UP, WHICH MEANS YOUR BET IS PLACED ON ANY OF THE SINGLE NUMBERS. YOU CAN PLACE COMBINATION BETS; THESE ARE BETS DIVIDED OVER A COMBINATION OF ADJOINING NUMBERS.

A Five Line (only available on 00 Roulette Tables), covers the five numbers 0, 00, 1, 2 and 3 and pays odds of 6 to 1.

A Split is a bet placed between two numbers and pays odds of 17 to 1 if the ball comes to rest on either of those numbers.

If you place a **Street** (odds of 11 to 1), on the line adjoining 7, it would win if 7, 8 or 9 was the winning number. A Street could also be placed, between 0, 2 and 3.

A Dozen bet is placed in one of the boxes marked Dozen and it would pay odds of 2 to 1 as per the following: 1st Dozen (Any number between 1 and 12), 2nd Dozen (Any number between 13 and 24) and 3rd Dozen (Any number between 25 and 36).

A Straight Up bet (odds of 35 to 1), it could be on the number shown on the diagram, or on any individual number on the table.

A Column bet is placed in one of the three boxes at the bottom of the table and pays odds of 2 to 1 if the ball comes to rest in one of the numbers in that column.

A Corner covers 4 numbers and pays odds of 8 to 1; this bet can also be placed to cover 0, 1, 2 and 3 as shown in the diagram.

A Six Line wager may be placed to cover six numbers by placing a chip on the intersection of those numbers. This would pay odds of 5 to 1 if any one of the six numbers results.

*DENOMINATION IS 0.01 IN BGN

1st 12												2nd 12												3rd 12											
1-10	EVEN		RED		BLACK		ODD		19-36																										

Този екран
има само
един
бутон за
Изход

```
class InfoScreen
```

```
{
```

```
public:
```

```
InfoScreen();
```

```
virtual ~InfoScreen();
```

```
bool Draw();
```

```
bool Clear();
```

```
bool getFlag(){return isActive;}
```

```
public:
```

```
LTexture* infoBackground;
```

```
Button* infoBack;
```

```
bool isActive;
```

```
};
```

class InfoScreen



ИГРАЛЕН ЕКРАН



БУТОНИ:
Завърти рулетката

Изчисти залозите

Изтегли пари

Последните 18
сектора

Счетоводство

class GameBoard

```
class GameBoard
{
public:
    GameBoard();
    virtual ~GameBoard();
    bool Draw();
    bool Clear();
    bool getFlag(){return isActive;}
    void DisplayStatistics(Credits* credits , int
lastWinningNumber);
    int CalcQuadrantClicked(int x, int y);
    void DisplayBets(Credits* credits , int x, int y,
int color,
    bool resume = false);
```

```
public:
    LTexture* gameBoard;
    Button* gameBoardPools[POOLS_BUTTON];
    Button* cashOut;
    Button* spin;
    Button* history;
    Button* accounting;
    Button* clearBets;
    Sound* sound;
    bool isActive;
};
```

class GameBoard

```
if (coordX != -1 && coordY != -1)
{
    if (credits->GetCredit() >= value[j]) //credits logic
    {
        Button gameBoardPools(coordX, coordY);
        gameBoardPools.loadFromFile("Pools.png");
        gameBoardPools.setWidth(POOLS_W);
        gameBoardPools.setHeight(POOLS_H);
        SDL_Rect rec = { j * 111 + 5, 1, 111, 110 };
        gameBoardPools.render(&rec, 0, NULL,
SDL_FLIP_NONE, 60, 60);
        //while piece to write on
        LTexture overPullUnderText(coordX + 17, coordY + 17);
        overPullUnderText.loadFromFile("BALL.png");
        overPullUnderText.setWidth(POOLS_W);
        overPullUnderText.setHeight(POOLS_H);
        overPullUnderText.render(NULL, 0, NULL,
SDL_FLIP_NONE,
        POOLS_W / 3, POOLS_H / 3);
        sound->play(CLICKBUTTON);
```

```
if (!resume)
{
    credits->AddBet(value[j]);
    credits->ChangeCredits(-value[j]);
    credits->betByNumberCell[clickedCell] += value[j];
}
Text textInPool(coordX + 20, coordY + 22, POOLS_W / 2
+ 8,
    POOLS_H / 2 + 8, 15,
    credits->betByNumberCell[clickedCell]
    , { 0, 0, 0, 255 }, "Intro.otf");
Text textBet(SCREEN_BOARD_W / 2 + 25, 40, 45, 35,
15,
    credits->GetBet(),
    { 200, 200, 200, 255 });
recovery->appendToXML(credits->betByNumberCell);
recovery->appendToXMLCredits(credits->GetCredit(),
    credits->GetBet(), credits->GetCreditsCollected());
} //end credits
}
```

class GameBoard

```
GameBoard::~GameBoard()
```

```
{  
    for (int i = 0; i < POOLS_BUTTON; i++)  
    {  
        delete gameBoardPools[i];  
    }  
    cashOut->free();  
    delete cashOut;  
    spin->free();  
    delete spin;  
    history->free();  
    delete history;  
    accounting->free();  
    delete accounting;  
    clearBets->free();  
    delete clearBets;  
    delete sound;  
}
```

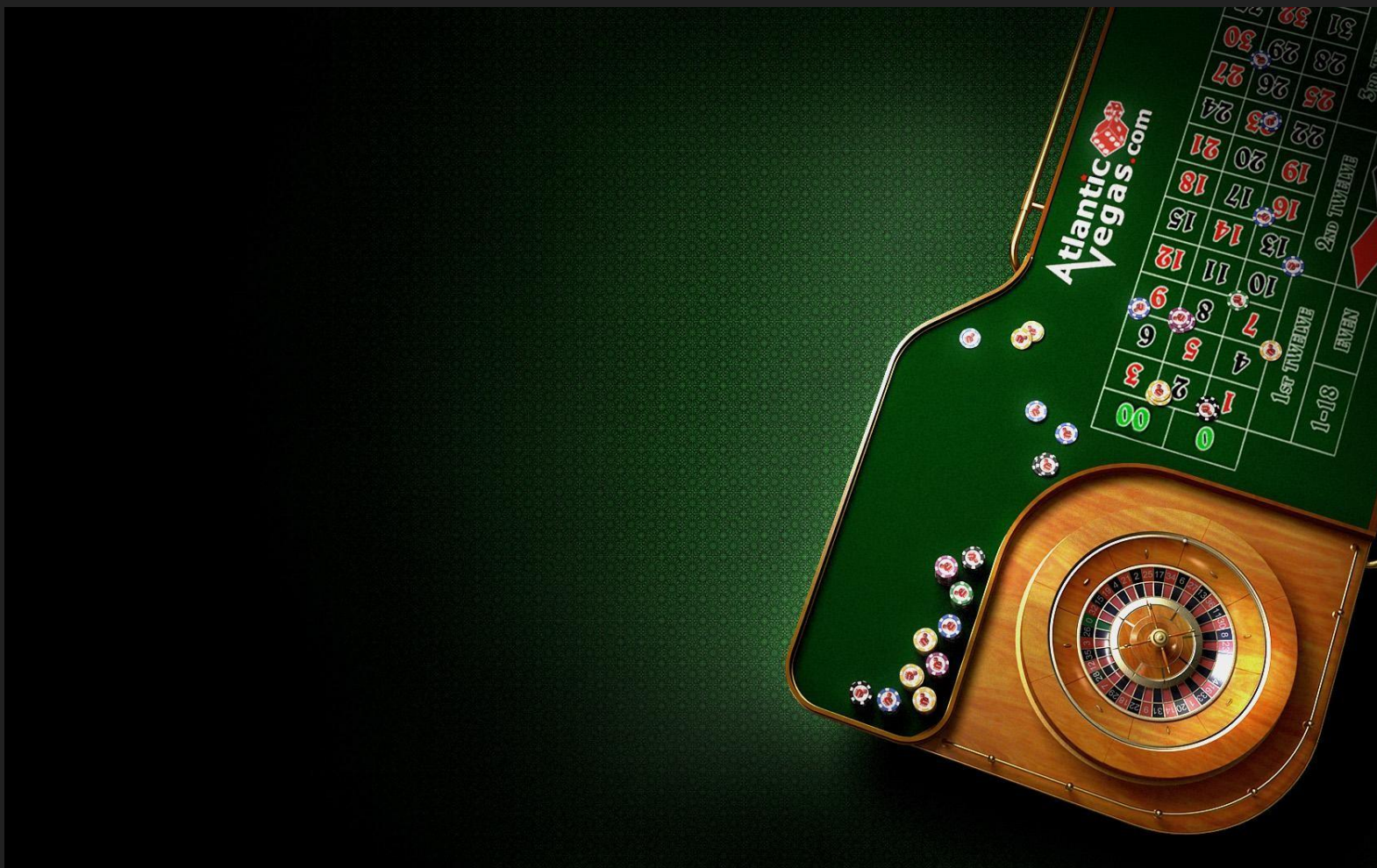
```
bool GameBoard::Draw()
```

```
{  
    background->render(NULL, 0, NULL);  
    isActive = true;  
    return true;  
}
```

```
bool GameBoard::Clear()
```

```
{  
    SDL_RenderClear(LWindow::gRenderer);  
    isActive = false;  
    return true;  
}
```

ИЗХОДЕН ЕКРАН



Без бутони

6 секунди

презентиране

class OutroScreen

```
class OutroScreen: public Screen
{
public:
    OutroScreen();
    virtual ~OutroScreen();
    bool Draw();
    bool Clear();
    void Show(Credits* credits);
};
```

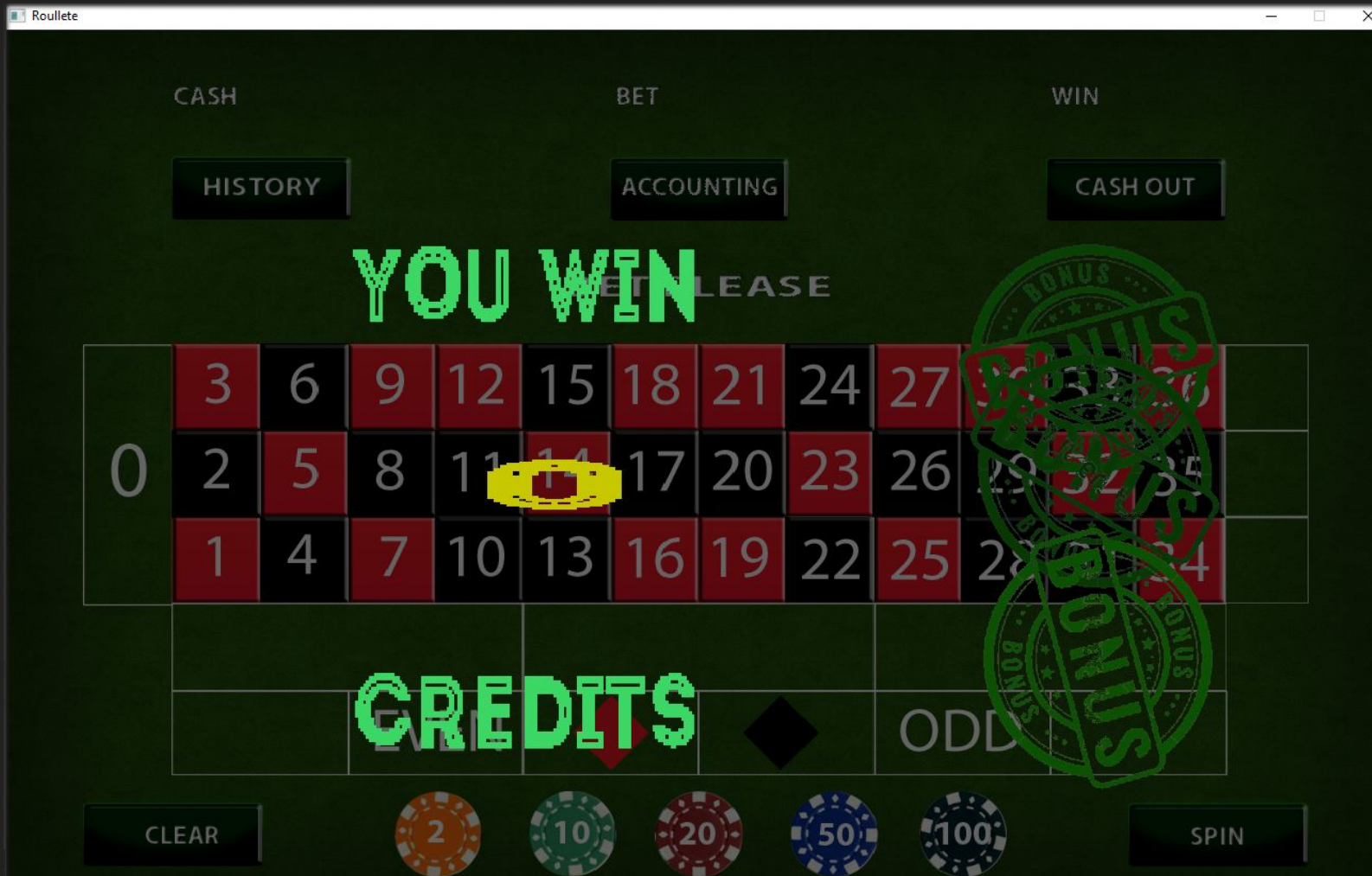
```
OutroScreen::OutroScreen()
    : Screen()
{
    background->loadFromFile("OutroScreen2.jpg");
    background->setWidth(SCREEN_W);
    background->setHeight(SCREEN_H);
    isActive = false;
}
OutroScreen::~~OutroScreen()
{
}
```

class OutroScreen



```
void OutroScreen::Show(Credits* credits)
{
    double scale = 0.6;
    Text textMoneyNumber((SCREEN_W / 8 +
520) * scale, 400 * scale, 180 * scale,
    200 * scale, 15, credits->GetCredit() *
DENOMINATION, { 200, 10, 10,
    255 });
    cout << credits->GetCredit() << "bbbb" <<
endl;
}
bool OutroScreen::Clear()
{
    SDL_RenderClear(LWindow::gRenderer);
    isActive = false;
    return isActive;
}
```

БОНУС ЕКРАН



Без бутони
3 секунди
презентиране
анимация с
бонус
печати.

class BonusScreen

```
class BonusScreen: public Screen
{
public:
    BonusScreen();
    virtual ~BonusScreen();
    bool Draw();
    bool Clear();
    bool getBonusCreditsText(Credits*);
public:
    LTexture* bonusSticker;
    Sound* sound;
};
```



class BonusScreen

```
bool BonusScreen::Draw()
{
    background->render(NULL, 0, NULL);
    SDL_Delay(2000);
    sound->play(WIN);
    srand(time(0));
    SDL_Color col = {
        rand() % 200 + 50,
        rand() % 200 + 50,
        rand() % 200 + 50,
        255 };
}
```

```
Text textYouWin(SCREEN_W / 4, SCREEN_H / 4,
    SCREEN_W / 4,
    SCREEN_H / 8, 30, "You Win", col, "Intro Inline.otf");
Text textCredits(SCREEN_W / 4, SCREEN_H * 3 / 4,
    SCREEN_W / 4, SCREEN_H / 8,
    30, "credits", col, "Intro Inline.otf");
for (int i = 0; i < 3; i++)
{
    SDL_SetTextureColorMod(bonusSticker->getTexture(),
        rand() % 100 + 150, rand() % 70 + 70, rand() % 100 + 150);
    bonusSticker->setY((i + 1) * SCREEN_H / 6 + rand() % 50);
    bonusSticker->render(NULL, i * 45, NULL);
    SDL_Delay(500);
}
isActive = false;
return true;
}
```


class BonusScreen

```
bool BonusScreen::getBonusCreditsText(Credits* credits)
```

```
{  
  
    srand(time(0));  
  
    Text textWinnings(SCREEN_W / 3,  
    SCREEN_H * 2 / 4,  
    SCREEN_W / 8, SCREEN_H / 12, 40,  
    credits->GetCreditsCollected(),  
    { 200, 200, 0 }, "Intro Inline.otf");  
  
    SDL_Delay(1500);  
  
    return true;  
  
}
```



ВЪРТЯЩА СЕ РУЛЕТКА



Допълнителен екран
включващ анимация на
въртящо се колело с топче

```
class SpinScreen: public Screen
{
public:
    SpinScreen();
    virtual ~SpinScreen();
    bool Draw();
    bool Clear();
    int GenerateWinningNumber();
    int GetWinningNumber();
    bool IsReadyForBonus();
private:
    LTexture* roulette;
    LTexture* wheel;
    LTexture* ball;
    Sound* sound;
private:
    int numberOfSpins;
    int winningNumber;
    void FillTheMapsOfRoulette();
};
```

class SpinScreen



```
bool SpinScreen::Draw() {
    int result = GenerateWinningNumber();
    background->render(NULL, 0, NULL);
    sound->play(SPINROULETTE);
    SDL_Delay(2000);
    int mFrame = 0;
    double angleWheel = -3 + 9.7 *
(posissionToNumberInRoulette[result] - 5);
    double stepWheel = 2;
    double maxR = 310, minR = 210,
        currentR = maxR, minAngle = -95,
        step = M_PI / 36, angleBall = 0;
    do {
        angleBall -= (maxR - currentR) / (12 * 200.0 / M_PI);
        currentR -= (maxR - minR) / (10 * 200.0 / M_PI);
        angleBall -= step;
        ball->setX(SCREEN_W / 2 - BALL_W / 2 + cos(angleBall) *
maxR);
        ball->setY(SCREEN_H / 2 - BALL_H / 2 + sin(angleBall) *
maxR);
```

```
if (mFrame % 3 == 0)
{
    roulette->render(NULL, 0);
    wheel->render( NULL, angleWheel);
    ball->render( NULL, 0);
}
angleWheel += stepWheel;
maxR -= 0.1;
mFrame++;
}
while (angleBall > minAngle);
sound->play(WINING_NUMBER);
SDL_Delay(2000);
sound->music(result);
SDL_Delay(1000);
numberOfSpins++;
return true;
}
```

ИСТОРИЯ



Допълнителен екран
показващ последните
18 печеливши числа.

СЧЕТОВОДСТВО

Допълнителен
екран показващ
статистики от
играта.

ЕКРАН ПРИ ПЕЧАЛБА



Допълнителен
екран без
бутони
6 секунди
презентиране
анимация с
въртящи се
монети

class WinScreen

```
class WinScreen: public Screen
{
public:
    WinScreen();
    virtual ~WinScreen();
    bool Draw();
    bool Clear();
    void WinAnimation();
    void ShowCredits(Credits*);
private:
    void fillRectPosition();
    LTexture* coin[COIN_COUNT];
};
```

```
void WinScreen::ShowCredits(Credits* credits)
{
    WinAnimation();
    SDL_Color color { rand() % 255, rand() % 255, rand() % 255, rand()
% 255 };
    Text winAmmount(SCREEN_W / 2 - 300 / 2, SCREEN_H * 3 / 5,
        300, 100, 20, credits->GetWinProfit(), color);
    SDL_Delay(1000);
}
```

class WinScreen

```
void WinScreen::ShowCredits(Credits*
credits)
{
    WinAnimation();
    SDL_Color color { rand() % 255, rand()
% 255, rand() % 255, rand() % 255 };
    Text winAmmount(SCREEN_W / 2 -
300 / 2, SCREEN_H * 3 / 5,
    300, 100, 20, credits->GetWinProfit(),
color);
    SDL_Delay(1000);
}
```


```
void WinScreen::WinAnimation()
{
    vector<SDL_Rect> goldCoins;
    for (int i = 0; i < 10; i++)
        goldCoins.push_back(
            { COIN_W * i, 0, COIN_W, COIN_H });
    for (int i = 0; i < (int) coinFlipz.size(); i++) {
        coin[i]->setX(coinFlipz[i].x);
        coin[i]->setY(coinFlipz[i].y);
        SDL_Color color { rand() % 255, rand() % 255, rand() % 255,
rand() % 255 };
        Text winText(SCREEN_W / 2 - 300 / 2, SCREEN_H / 2 - 10/2,
    300, 100, 20, "YOU WIN", color);
        for (int j = 0; j < 10; j++) {
            SDL_RenderCopyEx(LWindow::gRenderer,
coin[i]->getTexture(),
                &goldCoins[j], &coinFlipz[i], -90, NULL, SDL_FLIP_NONE);
            SDL_RenderPresent(LWindow::gRenderer);
            SDL_Delay(5);
        } } }
```

```
void WinScreen::fillRectPosition()
{
    SDL_Rect rec = { 20, 675, 90, 90 };
    coinFlipz.push_back(rec);
    rec.x = 128;
    rec.y = 690;
    rec.w = 95;
    rec.h = 95;
    coinFlipz.push_back(rec);
    rec.x = 230;
    rec.y = 650;
    coinFlipz.push_back(rec);
    rec.x = 358;
    rec.y = 580;
    coinFlipz.push_back(rec);
    rec.x = 485;
    rec.y = 650;
    rec.w = 105;
    rec.h = 105;
```

```
    coinFlipz.push_back(rec);
    rec.x = 270;
    rec.y = 445;
    rec.w = 95;
    rec.h = 95;
    coinFlipz.push_back(rec);
    rec.x = 340; rec.y = 490;
    rec.w = 90; rec.h = 90;
    coinFlipz.push_back(rec);
    rec.x = 200; rec.y = 575;
    rec.w = 75; rec.h = 75;
    coinFlipz.push_back(rec);
    rec.x = 120; rec.y = 635;
    rec.w = 65; rec.h = 65;
    coinFlipz.push_back(rec);
    for (int i = 0; i < 9; i++){
        rec = coinFlipz[i];
        rec.x = SCREEN_W - rec.x - rec.w + 15;
        coinFlipz.push_back(rec); } }
```


ВЪЗСТАНОВЯВАНЕ НА ИГРАТА

При натискане на бутон RESUME в началният екран играта се възстановява до момента на прекъсване. Възстановяването се реализира с помощта на XML файлове.



RESUME

```
<?xml version="1.0"
<person id="00415"
    <name>Kris
    <address>
        <street>
        <city>80
```

```
if (intro.introButtons[4]->isClicked(&e))
{
    intro.sound->play(CLICKBUTTON);
    intro.Clear();
    if (game.Draw())
    {
        credits = recovery.readXMLWriteCredit(
            "roulette_recovery_credits.xml");
        credits.betByNumberCell = recovery.readXMLWriteMap(
            "roulette_recovery.xml");
        game.DisplayStatistics(&credits,
            recovery.lastWiningNumbers.back());
        for (int y = 285; y < SCREEN_BOARD_H; y += 75)
            for (int x = 77;
                x < SCREEN_BOARD_W - 77; x += 75)
            {
                if (y > 550)
                    x += 75;
                if (credits.betByNumberCell[
                    game.CalcQuadrantClicked(x, y)] > 0)
                    game.DisplayBets(&credits, x, y, 1,
                        true, &recovery);
            }
    }
}
```

ДЕМО

**БЛАГОДАРИМ
ЗА
ВНИМАНИЕТО**