-

**Х А З А Р Т Н А И Г Р А**

**Р У Л Е Т К А**

Финален Проект

**­**

**Участници:**

Виктор Цонев

Васил Какевски

Петър Ефимов

Росица

**Съдържание:**

1. Описание на играта
2. Основни състояния на играта
3. Модули и екрани на играта
4. Recovery / Възстановяване на играта
5. Имплементация
6. UML диаграма на модела на играта

1. **Описание на играта**

* Хазартна игра, състояща се от различни модули и екрани, бутони, зареждане/настойка на кредити, чипове със залози, въртящо се колело и топче, печалби, история, зареждане от прекъсната игра, записване в XML файл, звуци, текстове и др.
* Целта е участникът да зареди първоначален кредит, с който да прави залози върху игралната маса, след което да завърти рулетката. При приключване въртене на рулетата, топчето се намира на печелившата позиция, обявява се печалбата на играча в отделен екран, като печалбата е изчислена едновремено в кредити и пари.
* Хазартна игра Рулетка за един играч. Всички автоматизирани действия се извършват от компютъра.
* В екрана за залагания, играчът може да залага желани пулове върху избрани числа, както и да избира типа (четни/нечетни) числа или да заложи върху цвят (червен/черен) на числото.
* Позиции за залог на рулетката:
  + 36 числа [1 – 36]
  + 0 - Нулата
  + Четно/нечетно число
  + Черно/червено число

1. **Основни състояния на играта**

Играта Рулетка се състои от пет основни състояния:

***Intro*** – начално състояние на играта - представлява начален екран с бутони, които препращат към други модули/екрани/състояния или настойки



Бутони в състояние Intro ( Начален екран):

* Insert Credit – с този бутон се добавят кредити в играта ( стойността, която ще се

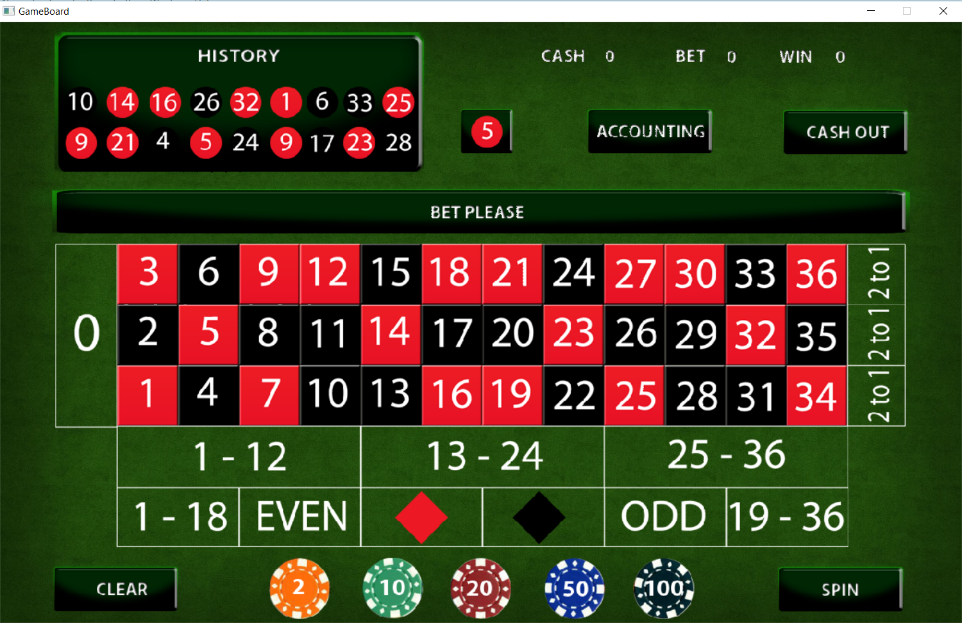
добави може да бъде фиксирана или произволна )

* Start new game – чрез този бутон се нулират всички текущо запазени състояние и/или настройки в XML файла, стартира се нова ( чиста ) игра и записването започва отново.

ТОДО: да се опишат другите бутони, след като им слежи ttf + картинки +

...

***Game*** – основно състояние със същинската логика за играта рулетка. Съдържа зелена маса за залози, със числа и други възможни позиции за залог. Има избор на различни чипове. Има възможност за проследяване на историята – пазят се последните 18 печеливши числа. Има бутон край на залозите и завъртане на рулетката. Има различни екрани, произлизащи от този модул при натискане на някой от бутоните му.

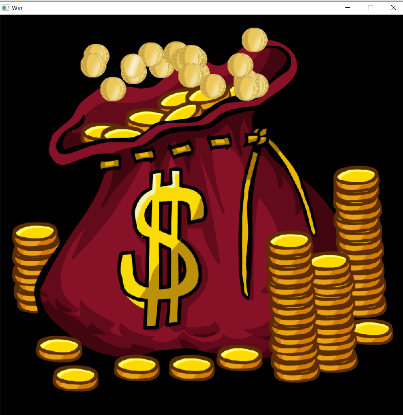


При натискане на бутона „Start New Game“ се отваря нов прозорец, който представлява стандартната игрална маса на играта Рулетка. На зелена фон, има числата от 0 до 36 и възможност за залагане и на друг тип позиции, освен конкретни числа, например: четно , нечетно или по цвят. Играчът има избор на залог от 5 различни пула, със стойности: 2, 10, 20, 50 и 100. На базата на първоначално зададения кредит, играчът може да избира и реди/залага пулове на масата, докато има наличен кредит по-голям от 0. При приключване на залаганията на играча, той се очаква да натисне бутон „Spin“, при което се отваря нов прозорец, където е видима анимацита – завъртане на два обекта: рулетка и топче. За допълнителен ефект, топчето ...

След финално спиране на топчето в определена позиция спрямо число се показва печалбата ...

***Win*** – състоянието, в което се намира играта при печалба

ТОДО: ...



***Bonus*** – състояние на допълнителна печалба

ТОДО:

...

***Outro*** – състояние, в което може да с влезе по желание на играча, след натискане на бутона *CashOut*. Така се изплащат текущите пари на играча.

ТОДО: ...

1. **Модули и екрани на играта**

**Екрани:**

* Задаване на начален кредит – става чрез натискане на бутона за настройка на кредит в Началния Екран на играта.

Тодо: Screenshot от 1ви екран, след като се сложи ttf on the button и се избере финална картинка за Началния екран

* Модули в състояние Game (екран GameBoard)

В състояние ‘Game’, на екрана ‘GameBoard’ има различни модули: кредити, бутони, тапи, история, въртящо се колело с топче.

ТОДО: за редакция след финална реализация + добави снимка

* модул с кредити - показва стойностa на текущите кредити, стойността на

заложените кредити по тапито (ако има такива) и печалбата, ако играта е печеливша

* модул с различни бутони:
  + бутон за смяна стойността и вида на чипа



* + бутон за изчистване на залозите по тапито



* + бутони за залагане на червено/черно



* + четно и нечетно

* + бутон за показване на стрaница счетоводство/accounting



* + бутон за край на залози и завъртане на рулетката



* модул тапи - игралната маса със сектори от 0 до 36. Залаганията могат да бъдат само на целите числа. Не може да се залага меджду две, три или четири числа с един чип. При кликане някъде из определен квадрант/клетка, пулът се позиционира централно автоматично.



* модула с история - показва последните 18 печеливши паднали се числа



* модул с въртящото се колело – обект разделен на равни части, съдържащ числата от 0 до 36, който се върти в една посока, а топчето в обратната посока докато не се падне някое число



ТОДО: ...

1. **Recovery / Възстановяване на играта**

Способност за възстановяване на играта от неочаквано състояние.

Имплементирана функция за запазване текущото си състояния и настройки в XML файл, за да може при непредвидени аварийни ситуации, да има опция/бутон за възстановяване/продължаване на играта от там откъдето е прекъсната.

..

ТОДО: Детайл как е реализирано(четен / писане от XML, къде в кой клас се случва и как е оптимизирано), screenshot.

1. **Имплементация**

**Класове в играта:**

**Class Application**

Кратко описание: ТОДО...

Методи на класа:

**void** **initIntro**(); - препраща към началния екран за избор на модул/екран

**void** **initInfo**(); - препраща към екран с информация и правила на играта

**void** **initGameBoard**(); - препраща към екран с игрална маса, чипове за залания, рулетка и е свързан с основната логика

**void** **initRoulette**(); - препраща към екран с въртяща рулетка с топче

**void** **initOutro**();

**void** **initWin**();

**void** **initSpin**();

**void** **Free**(); - освобождава ресурсите ( освобождава паметта на създадените с new обекти и др.)

**void** **GamePlay**();

**void** **DisplayBets**(**int** x , **int** y , **int** color, vector<Point> v\_allBetPoints); - рисува/позиционира заложените пулове върху позволените числа/позиции

**int** **CalcQuadrandClicked**(**int** x, **int** y); - спрямо координатите в/у които кликнато с мишката, изчислява поредността на клетките/числата започвайки да ги брой по ред и колона и функцията връща число, което е част от редицата [0-43]. На базата на това число се определя къде е направил залога си играчът.

**void** **SpinAnimation** (); - ТОДО

**void** **SpinRoulette**(LTexture &ball,**int**); - ТОДО

**int** **spinBall**();- ТОДО

**bool** **WinAnimation**();- ТОДО

**virtual** **~Application**();-

GameState **getMenuState**() **const**; - ТОДО

**Class Background**

Кратко описание: ТОДО...

Методи на класа:

**Background** (string title= "",**int** width=0, **int** height=0,string path="");

**virtual** **~Background**();

**void** **Clear**();

**void** **Show**();

**bool** **loadMedia**();

**bool** **Init**();

**int** **getHeight**() **const**;

**void** **setHeight**(**int** height);

**const** string& **getPath**() **const**;

**void** **setPath**(**const** string& path);

**int** **getWidth**() **const**;

**void** **setWidth**(**int** width);

**void** **setTitle**(**const** string& title);

**Class Ball**

Кратко описание: ТОДО...

Методи на класа:

**Ball**(**int** \_x,**int** \_y ,**int** \_speed);

**virtual** **~Ball**();

**bool** **spin**();

**bool** **isBallStoped**(){**return** isStoped;}

**int** **getSpeed**() **const**;

**void** **setSpeed**(**int** speed);

**Class Button**

Кратко описание: ТОДО...

Методи на класа:

**Button**(**int** x = 0, **int** y = 0);

**virtual** **bool** **isHover**();

**virtual** **bool** **isClicked**( SDL\_Event \*);

**virtual** **~Button**();

**Class Credits**

Кратко описание: ТОДО...

Методи на класа:

**Credits**(**int** = 0);

**Credits**(Credits &);

**virtual** **~Credits**();

**void** **AddBet**(**int**);

**void** **ChangeCredits**(**int**);

**void** **SetBet**(**int**); //needless

**int** **GetBet**() **const**;

**int** **GetCredit**() **const**;

**int** **GetPossibleWin**() **const**;

**void** **SetPossibleWin**(**int**);

**bool** **isCashedOut**();

**static** **int** **NumberInCell**(**int**);

**Class Globals**

Кратко описание: Съдържа константни, статични и промениливи от тип enum.

ТОДО...

**Class LTexture**

Кратко описание: ТОДО...

Методи на класа:

**LTexture**(**int** x = 0, **int** y = 0);

**virtual** **bool** **loadFromFile**(SDL\_Renderer\* gRenderer, std::string path,

**int** width = -1,

**int** height = -1);

**virtual** **~LTexture**();

**virtual** **void** **render**(SDL\_Renderer \* gRenderer, SDL\_Rect \*clip,

**double** angle = 0,

**int** w = 0, **int** h = 0);

**virtual** **void** **free**();

**virtual** **void** **Show**();

**int** **getHeight**() **const**;

**void** **setHeight**(**int** height);

**int** **getWidth**() **const**;

**void** **setWidth**(**int** width);

**int** **getX**() **const**;

**void** **setX**(**int** x);

**int** **getY**() **const**;

**void** **setY**(**int** y);

**virtual** **void** **setPosition**(**int** \_x, **int** \_y);

**Class Main**

Кратко описание: ТОДО...

Методи на класа:

**Class Point**

Кратко описание:

Представлява обект, който съдържа 4 пропъртита:

* int X (координатът на кликване с мишката)
* int Y (координатът на кликване с мишката)
* string m\_PullColor (цвета на заложения пул)
* int m\_value (стойността на заложения пул: 2,10,20,50 или 100)

ТОДО...

**Class Pools**

Кратко описание: ТОДО...

Методи на класа:

**Pools**();

Credits cr;

**Pools**(Credits& ,**int** x=0,**int** y=0,**int** number=0);

**virtual** **~Pools**();

**bool** **isBet**();

**int** **GetNumber**();

**int** **PossibleWin**();

**Class RouletteWheel**

Кратко описание: ТОДО...

Методи на класа:

**RouletteWheel**(**int** \_x,**int** \_y,**double** angleSpeed);

**virtual** **~RouletteWheel**();

**void** **Spin**();

1. **UML диаграма на модела на играта**

ТОДО: С помощта на UML tool да се създаде визуално създадения модел на играта. Ако визуално има място , да се добавят properties and methods на класовете.

**Информация от Заданието, която трябва да бъде добавена някъде горе, спрямо съответната имплементация....или ако не е реализирано , да не се добави,**

Intro – всяка игра трябва в това си състояние да има следните бутони:

\* Volume – бутон за регулиране звука на играта

\* Info – бутон с кратка информация за правилата на играта и валутата плюс единствената

деноминация на която ще се играе. Всеки екип трябва да избере една от следните деноминации:

0.01, 0.25, 5.00, 2000.00 ( пример: ако избера деноминация 0.25 BGN и вкарам 300 кредита от

бутона Insert Credit и след това веднага натисна бутона Cashout, сумата, която ще се

изпише на поздравителния екран е 75 BGN ).

\* Resume game – с този бутон се зареждат всички текущо запазени настройки в XML

файла, като играта по този начин се възстановява там от където е спряна.

Win – след всяка печалба по голяма от 0 трябва да се появи поздравителен екран, на

който да се изпише спечелената сума в кредити и пари едновременно. Този екран да се премахва

автоматично след 6 секунди.

Bonus – след всяко изтегляне на число, независимо дали има печалба или не, трябва в

отделен буфер да се натрупват по 13% от всеки залог и на случаен принцип натрупаните пари да

се връщат на играча като мистериозна печалба ( Mystery Progressive Jackpot ).

Outro – трябва да се покаже поздравителен екран със спечелената сума, конвертирана от

кредити в пари, според избраната деноминация, като този екран трябва да остане 10 секунди и

след това да се премине автоматично към екран Intro.

\* Счетоводство/accounting - на тази допълнителна страница трябва да се покаже

информация каква е била печалбата от всяко число поотделно и сбора от всички печалби до

момента, както и общата сума пари вкарани чрез бутона Insert Credit, общата сума пари

изкарани с бутона Cashout и какъв е процента на възвръщаемост RTP – return to player)

\*\*\* Всеки проект трябва да съдържа следните компоненти:

- Source code.

- Документация – техническа документация на проекта, схеми, диаграми и др.

- Инструкции за билдване на проекта, списък на външни библиотеки и т.н.