**Документација за проектот по визуелно програмирање**

Изработиле:

Стефан Стеафновски индекс бр. 151028

Борис Танчовски индекс бр. 151004

Тема на проектот: 3D Пинг Понг симулатор

Темата ја одбравме со цел да ги усовршиме техниките за цртање графика кои ги работевме по предметот. Целта беше да се нацрта пинг понг табла заедно со мрежа, палки и топче во 3 димензионална перспектива на играчот. Ова го постигнавме со дефинирање на сите рабови и темиња на објектите во 3Д координатен систем и потоа користејќи трансформациона матрица за камера во просторот која ја надгледува таблата, ги проектиравме овие 3Д координати во перспектива на камерата и потоа користејќи едноставна перспективна трансформација ги цртаме на екран пред камерата.

Решението се состои од класите Ball, PlayerPaddle, EnemyPaddle и Table, кои содржат информации за објектите како и методи за меѓусебна интеракција и класите SoundFX (за озвучување на удар на топчето) и Camera (за методите за трансформација).

Во продолжение подетално ќе ја објасниме класта Ball.  
Таа ги содржи позицијата на топчето, радиусот, информации за движењето (насока, брзина, гравитација) и други помошни информации како дали се тркала или паѓа.  
Тука се дефинирани и методите за:

* Цртање на топчето кој исто така ја црта и сенката зависно од тоа дали е врз масата,
* Движење на топчето кој проверува дали се удрило од масата или од палките, дали излегло надвор од границите на маста, ја следи видливоста на топчето за дали би било цртано пред или зад масата и палките.
* Помошни методи ‘void CheckPaddleHit’, ‘void CheckEnemyPaddleHit’, ‘int findZIndex’, ‘bool IsOut’ кои помагаат за остварување на проверките на Move методот и повикуваат други методи за менување на состојбата.
* Методите PaddleHit и EnemyPaddleHit (кои се повикуваат од CheckPaddleHit и CheckEnemyPaddleHit доколку има контакт помеѓу топчето и соодветната палка) и методот BounceUp (кој се повикува доколку има контакт помеѓу топчето и масата) ја менуваат насоката и брзината на движење на топчето и го повикуваат методот Pong() од класата SoundFX кој потоа случајно репродуцира еден од 3 различни звуци на удар на топче.

Следно ќе анализираме дел од кодот на Move() методот на оваа класа.

public void Move()

{

center.X += Math.Cos(angle) \* velocityAngled;

center.Z += Math.Sin(angle) \* velocityAngled;

CheckEnemyPaddleHit();

CheckPaddleHit();

zIndex = findZIndex();

CheckEnemyPaddleHit();

CheckPaddleHit();

zIndex = findZIndex();

if (rolling && velocityAngled > 0)

{

velocityAngled -= 0.01;

if(velocityAngled < 0)

{

velocityAngled = 0;

}

if (IsOut() && center.Y <= 0)

{

falling = true;

rolling = false;

}

}

if (!rolling || falling)

{

velocityUp -= gravity;

center.Y += velocityUp;

if (!falling && velocityUp < 0 && center.Y <= radius)

{

if (center.Y < radius) center.Y = radius;

if (!IsOut())

{

if (prevZIndex == zIndex)

{

awardPoint(zIndex);

}

BounceUp();

}

else

{

falling = true;

if (center.Z > 1590/2)

PlayerPaddle.points++;

else

EnemyPaddle.points++;

parentForm.startTimer();

}

}

}

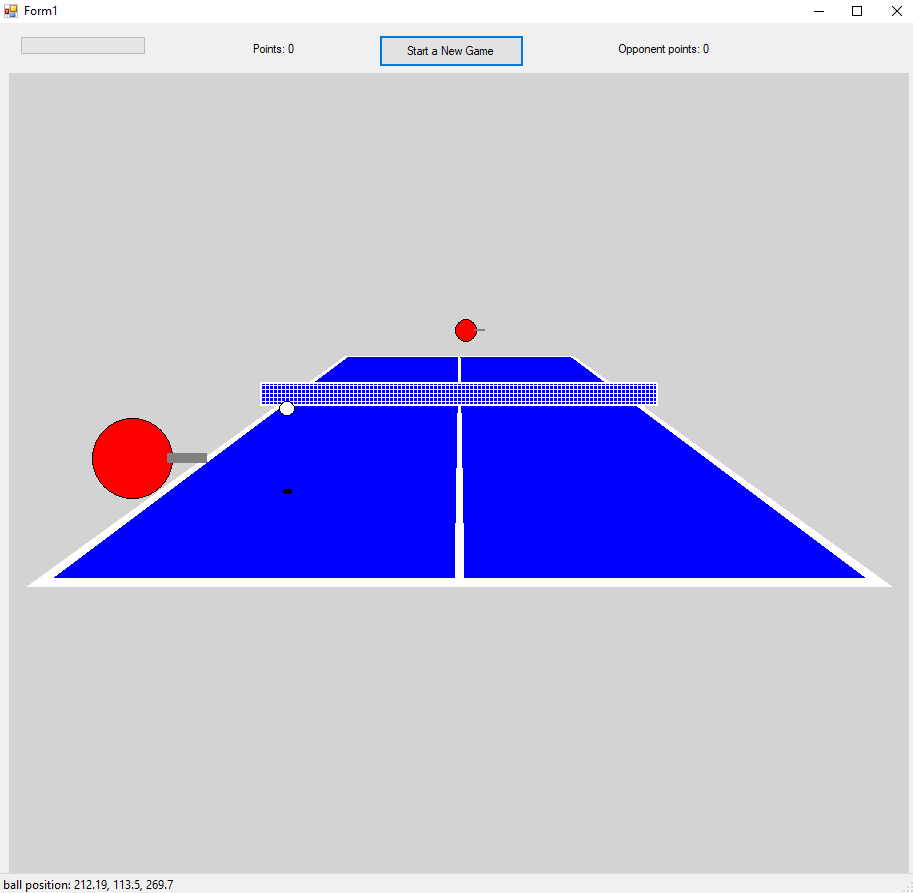
}

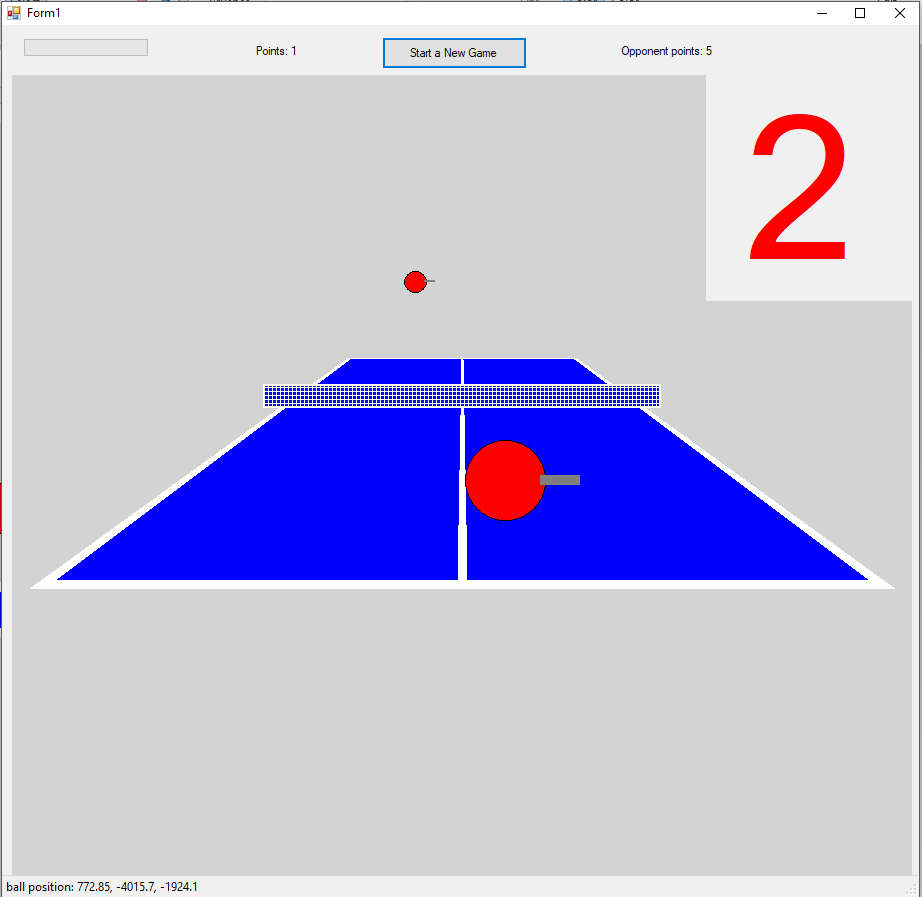
Првите 2 реда го поместуваат центарот на топчето за соодветна вредност зависно од аголот и брзината на движење. Потоа ги повикува методите CheckEnemyPaddleHit(), CheckPaddleHit() и zIndex = findZIndex() кои се справуваат со претходно објаснетите проверки.

Следниот if блок се извршува само ако топчето е во состојба на тркалање и не стои во место. Внатре во блокот при секое вакво движење ја намалуваме брзината на топчето за да симулираме триење и проверуваме дали топчето паднало од масата користејќи го методот isOut(), доколку тоа се случило, топчето запира со тркалање и паѓа од масата.  
  
Во следниот if блок, кој се извршува доколку топчето паѓа или не се тркала, најпрво ја менуваме неговата состојба во однос на висината (Y оска), и симулираме дејство на гравитација, со тоа што ја намалуваме нагорна брзина (velocityUp). Потоа правиме проверка дали топчето се движи надолу и дали топчето доаѓа до висина еднаква на висината на масата. Доколку да, проверуваме дали а внатре или надвор од масата. Ако е внатре, тогаш го повикуваме методот BounceUp, кој ја превртува неговата нагорна брзина, дејствувајќи топчето да се одбие од масата и придвжи нагоре. Доколку не, тоа значи дека топчето паднало надвор од масата, и проверуваме на која страна од масата се случило тоа и соодветно додаваме поен на соодветниот играч, и почнуваме нова игра, користејќо го методот StartTimer од parentForm.

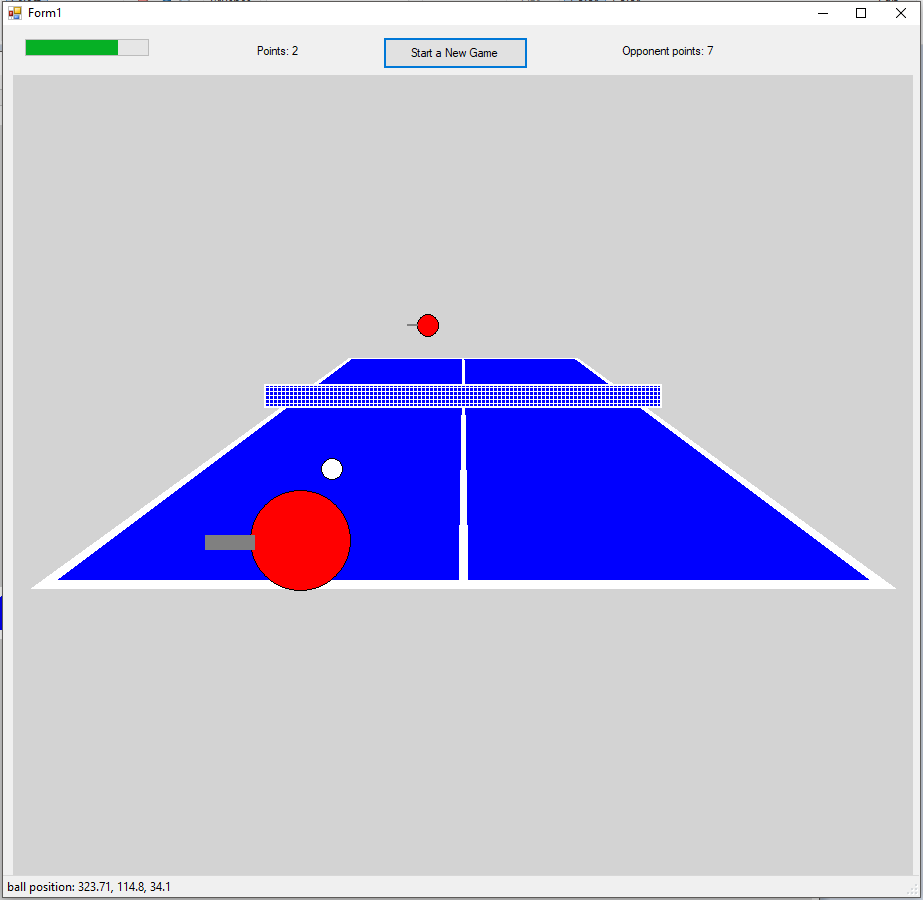
Во продолжение ќе го опишеме начинот на играње преку слики и текст.

Со двужење на маусот се придвижува палката на игачот. Со клик и задржување на маусот се зголемува јачината на ударот и соодветно се зголемува палката, со изпуштање на маусот се враќа во нормална состојба.

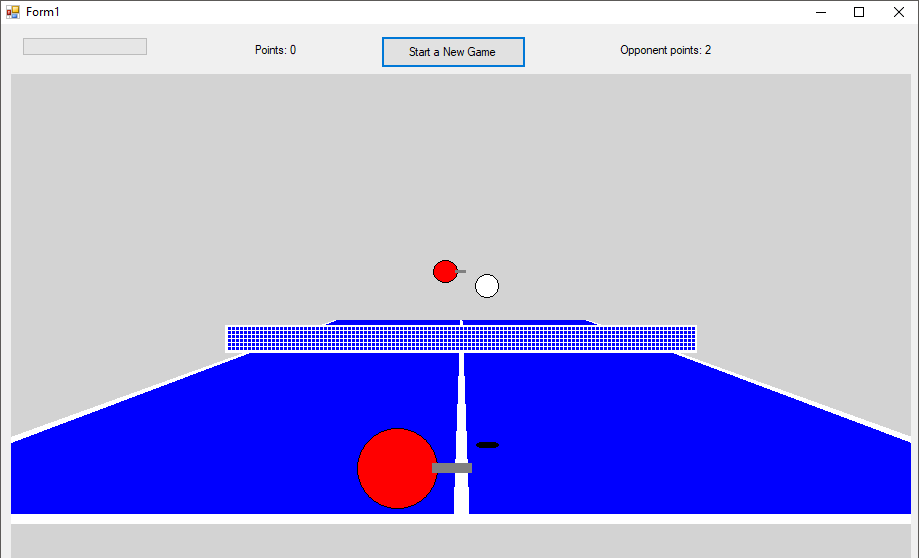
  
Слика 1. Нова игра



Слика 2. Одбројување на секунди до почеток на нова игра по добивање/губење поен.



Слика 3. Држење на маусот за зголемување на моќноста на ударот (соодветно полнење на progressBar во горе левиот агол).



Слика 4. Можност за дефинирање на различна позиција на камерата.