Minesweeper

Оваа мала и едноставно имплементинрана игра е со цел да им направи забава и предизвик на играчите. Многу е слична за оригиналната Minesweeper игра, но овде имаме додадено малку бои и дополнително избор за едно од две нивоа на тежина во играта со тоа што едното е за лесна игра без притисок и предизвици, а другото за малку поголем предизвик и притисок заради времето.

-Објаснување на проблемот и упатство

Играта е доста едноставна, главната цел е полето кое го избирате да не се крие мина под него, ако успеете да ги избегнете сите мини вие сте победник во спротивно губите и добивате шанса повторно да изберите нова игра при што избирате дали сакате да играте без никаков притисок со притискање на копчето Easy или сакате малку предизвик и ке одберете Hard и ке играте под притисок на време односно откако ке ја започнете играта имате 5 секунди после секое откривање на поле за да откриете ново и да не се крие мина под него, во спротивно го губите правото да продолжите со играње и ја губите играта. Можете да застанувате на полињата кои се потполно празни или се крие број под нив. Овие броеви ви помагаат при што ви кажуваат колку од нивните соседни полиња кријат мини под нив, и ваша работа е да процените добро кое поле би можело да крие мина и да не застанете на него. Дополнително погоре имате progress bar кој се вклучува само кога имате одбрано Hard или text box којшто кажува уште колку секунди до губење на правото за играње според кои можете да се ориентирате уште колку време имате преостанато, исто така имате и text box кој во кажува колку полиња сте успеале да откриете во текот на играта и неговата вредност е променлива со секое ново откриено поле.

-Решение на проблемот

Во играта нема класи, но има доста функции, на почеток кога ке ја вклучите играта по default е на Easy и ваш избор е дали сакате да го смените тоа.

Функцијата `GenerateButtons()` се повикува на самото влкучување на играта и генерира матрица од копчиња(полиња) 12 x 12 кои ви дозволуваат да притискате на нив и притоа ви ја откриваат вредноста која ја содржат.

GenerateMines() оваа фукнција исто така се повикува при вклучување на играта и ги генерира мините на рандом позиции во полето.

GenerateNumbers() исто така се повикува при стартување на играта и генерира бројки на полињата каде што нема мини, а некои ги остава празни.

RightClickBtn() оваа функција ни овозможува да си ги означиме со знаменце потенцијалните полиња кои можеби содржат мини.

btnEasy_Click() функција која е директно поврзана со копчето Easy и при кликање целата игра се рестартира и сите финкционалности се прават за на играчот да ми е полесно, се собираат само вредностите на отворените полиња.

btnHard_Click() и оваа функција исто како предходната е дироктно поврзана со копчето Hard и при кликање целата игра се рестартира и почнива да се одбројува време за играчот да почне да притиска на полињата има само по 5 секунди кои се прикажани како progress bar и text box.

OpenAdjacentEmptyTile() со оваа финкција се открива што се крие позади секое поле ако е мина играта завршува изгубивте, ако е празно поле се отвараат и останатите празни полиња кои се до него, а ако е бројка се открива само бројот кој кажува соседи кријат мини.

BtnClick() дозволува повикување на OpenAdjacentEmptyTile() на полето кое е избрано и при тоа при секој клик времето на истекување се рестартира и му дава на играчот уште 5 секунди за размислување пред да го открие следното поле.

timer1_Tick() оваа функција се извршува на секоја секунда се при што проверува дали progress bar е исполнет и времето е поминато и се завршува играта.

-Опис на функција

Како што е спомнато погоре оваа функција е за да ги генерира броевите во полињата со тоа што креираме два for циклуци кои креираат матрица 12 x 12 и за секое поле се проверува дали има вредност 10 ако има вредност 10 тоа значи дека на тоа поле се наоѓа мина и дека таму не треба да се сместува вредност туку само да се продолжи со следното поле од матрицата. Потоа се креира променлива bombCounts којашто во себе носи содржина од тип byte и оваа променлива служи за броење на бројот на мини којшто се наоѓаат во некој од соседните полиња.

```
if (Positions[x, y] == 10)
	continue;
byte bombCounts = 0;
```

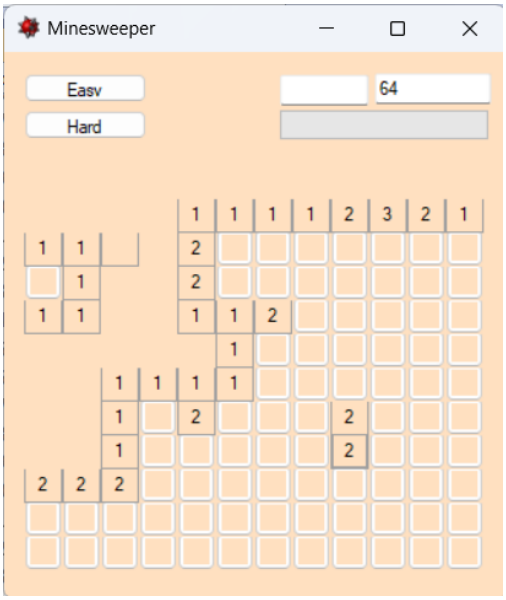
За да можеме да провериме за секое поле колку од соседите содршат мини правиме уште два for циклуси едниот проверува на X-оската(хоризонтално) дали има мини , а другиот на У-ската(вертикално) дополнително се проверува ако полето кое се проверува не е во матрицата да се продолжи понатаму или ако полето кое се проверува е полето кое на кое се наоѓа циклусот во матрицата исто така да се продолжи.

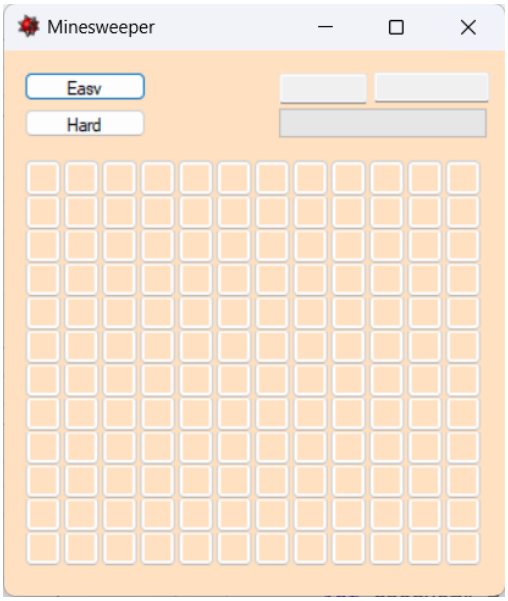
```
for (int counterX = -1; counterX < 2; counterX++)
{
	 int checkerX = x + counterX;
	for (int counterY = -1; counterY < 2; counterY++)
	{
		int checkerY = y + counterY;
		if (checkerX == -1 || checkerY == -1 || checkerX > 11 || checkerY > 11)
			continue;
		if (checkerY == y && checkerX == x)
			continue;
		if (Positions[checkerX, checkerY] == 10)
			bombCounts++;
	}
}
```

Овој if/else проверува и кажува ако на полето на кое е се наоѓа циклусот нема соседи да биде празно поле, во спротивно да се додели вредност која кажува колку соседи кријат мини.

```
if (bombCounts == 0)
	Positions[x, y] = 20;
else
	Positions[x, y] = bombCounts;
```

-Screenshot од играта





Учесници во проектот: Стефан Петров(213288)