**HSE Survival**

**Kostina Anastasia & Sdobnov Vasiliy**

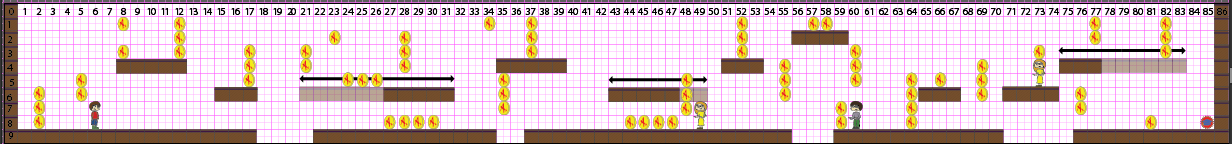
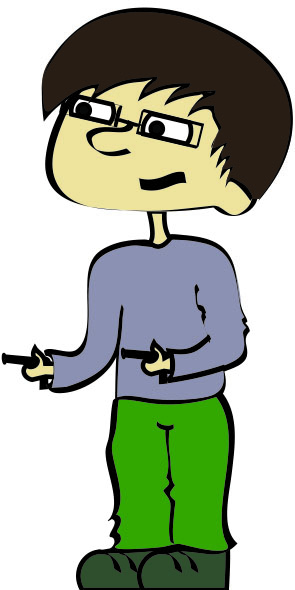
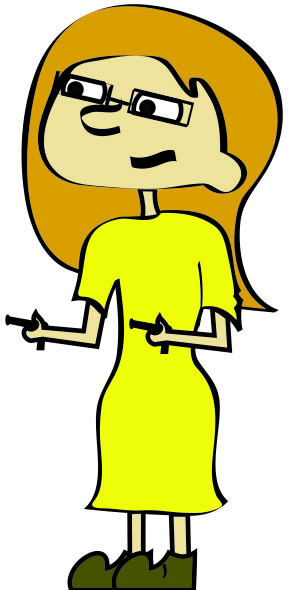
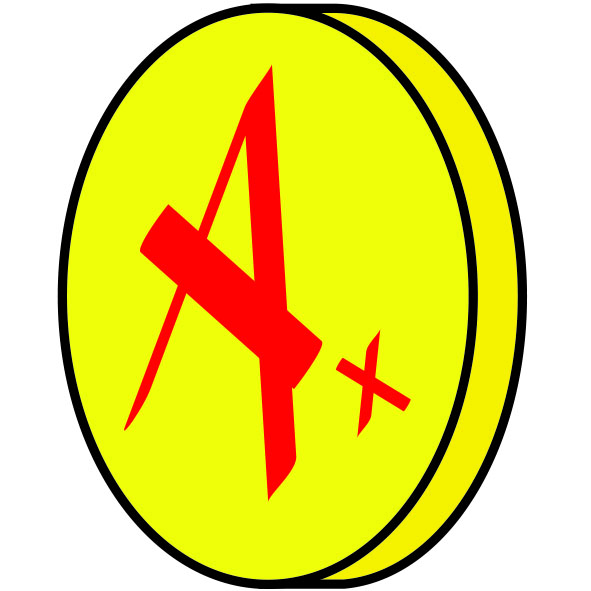
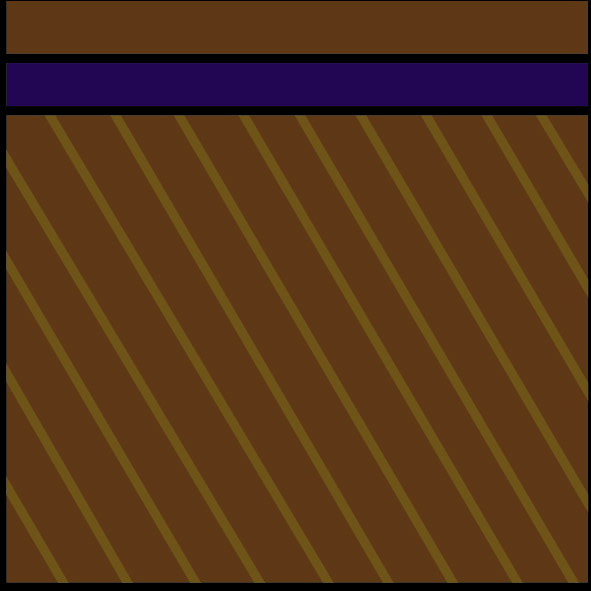
Platform game “HSE Survival” realized the idea of university survival. By the keyboard you can control character movements: “space” – jump, cursor control keys “left” “right”. From the top of the screen bomb-questions are falling. In case of collision with bombs or with a teacher character returns on an initial position and his scores set to null. However, game-time continue to go, so chances to have high result decreases. Bomb falling is turned off, if player kills all teachers. He can do that by jumping on them. The aim of the game is to reach “graduated hat”. In this case results window is opened, so you can fix your results in online database.

<https://github.com/NastyaKostina/HSE_Survival>

**Kostina Anastasia:** UI, general logic and testing. Also I helped my partner with implementation of some classes.

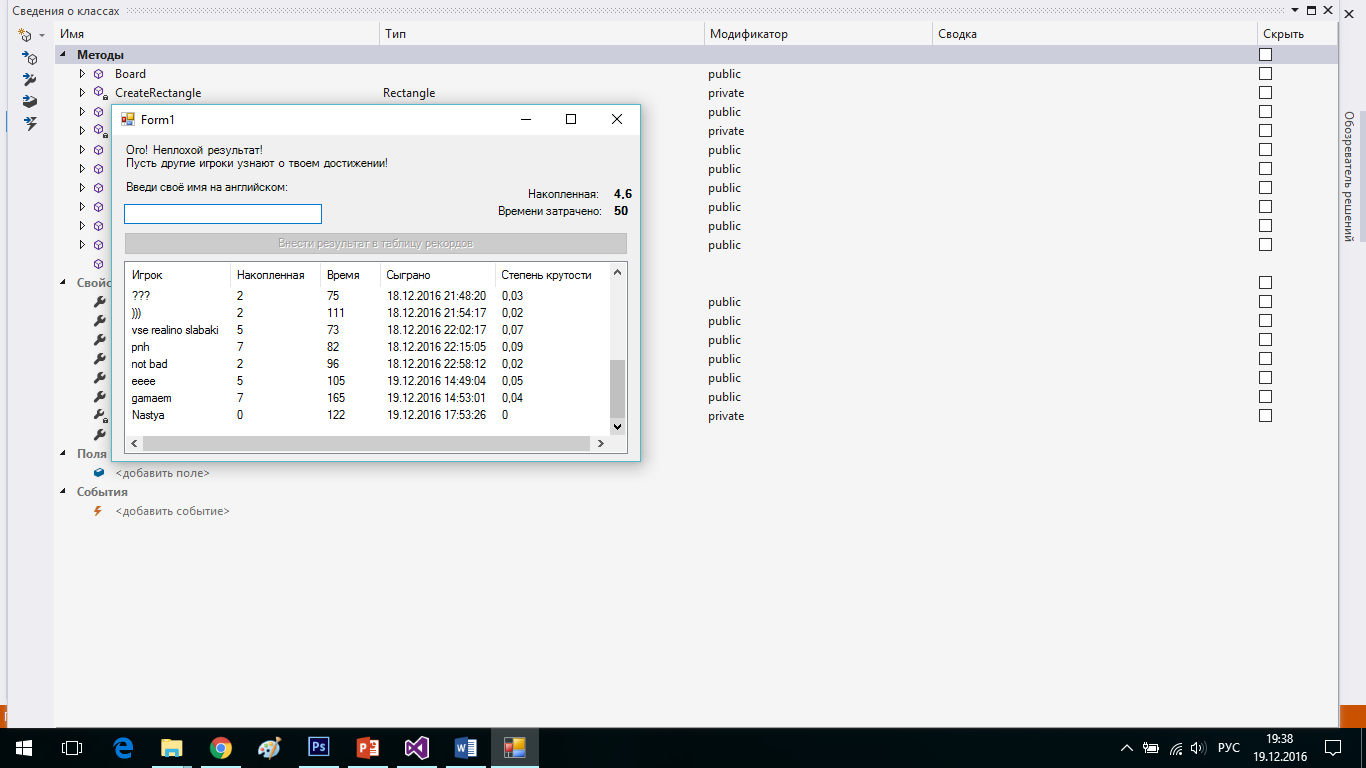
**Sdobnov Vasiliy:** Secondary logic and database management.

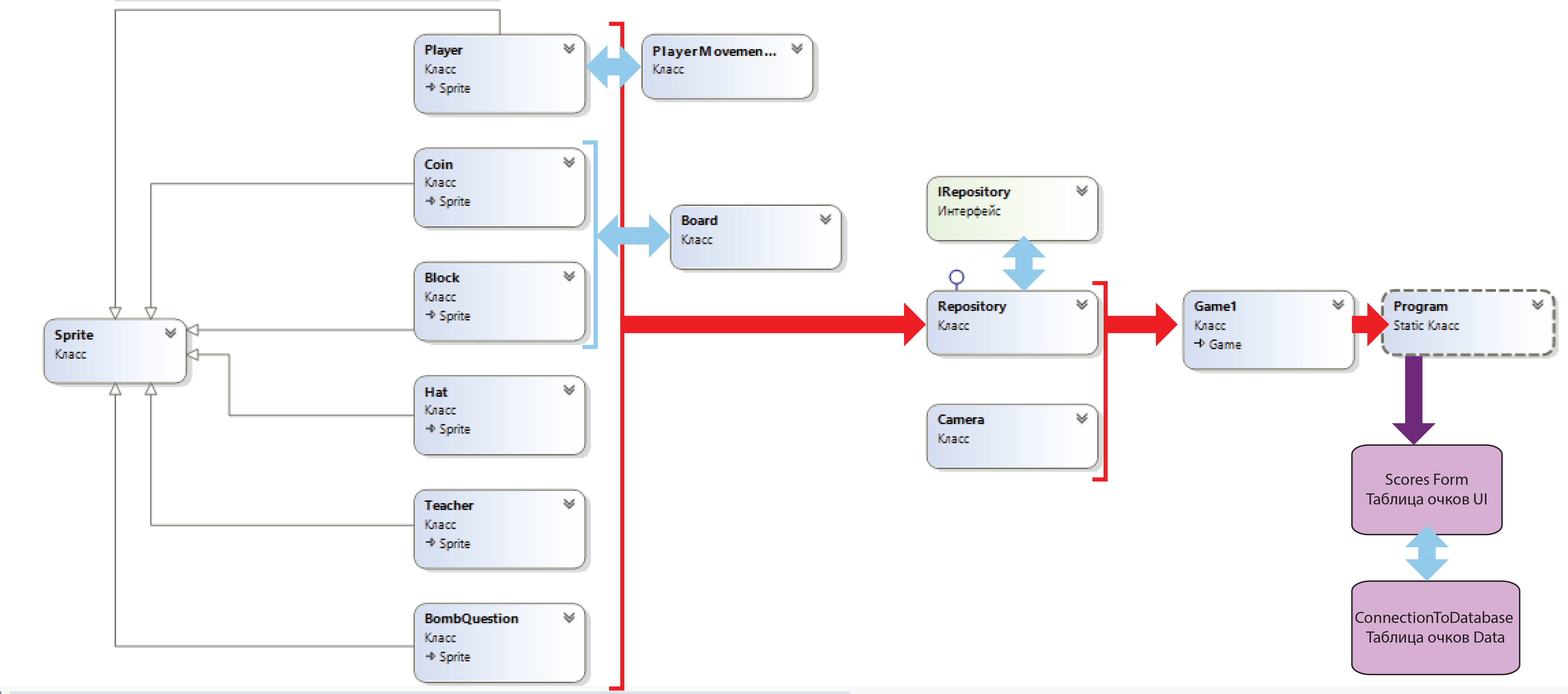




 **Screen of the game**

**Here you can write down your results**





“Sprite” – is a parent-class. It keeps general description of entities. There are six child-classes which hold individual implementations of each game entity. In order to separate UI and logic intermediate class “Repository” was created to store objects, draw them, set their values and call their collisions methods.

“Sprite helpers” classes were added to simplify the logic of child-classes. Thus “PlayerMovementSystem” class contains player movements logic. Board class is an initial map of our game. Actually it contains massives of inanimate objects: coins and blocks. Player’s interaction with a game world has been implemented through that class. Сoins have to be disappeared after collision with the student, so a list of them was added to the Repository class. Repository class always has to process collisions, draw objects, and set values, so “IRepository” interface was added. Game class is an implementation is an actual game process.

* *Interface IRepository - Collisisons(GameTime gametime); SetValues(Dictionary<string, Texture2D> Values, SpriteBatch spriteBatch); Draw(SpriteBatch spriteBatch, SpriteFont Font, GameTime gametime);*
* *Board – map-class, which is responsible for all inanimate objects, their drawing and interaction between player and object of the map.*
* *PlayerMovementSystem – class, responsible for movements of player, his gravitation and death-event (in the other words, if the player was moved – it happened in this class).*
* *Block – a class describing an essence of moving and static blocks.*
* *BombQuestion – a class describing an essence of questions falling from the sky.*
* *Coin – a class describing an essence of coins which equal to 0.1 point of player’s average grade.*
* *Hat – a class describing an essence of a graduated hat, which is a point of finish in the game.*
* *Player – a class describing an essence of a player.*
* *Sprite – a parent class for all essences.*
* *Teacher – a class describing teachers as an analogue of enemies IN THE GAME.*
* *Camera – a class, responsible for “point of view” in its original meaning.*
* *Game1 – a main class of MonoGame-projects which include methods of updating and drawing the game.*
* *Program – class, opening MonoGame-form and Windows-form and transmitting information from one form to another.*
* *Repository – class, storing all “non-block”-objects of the game and checking player’s actions of collisions with them.*

**ConnectionToDatabase**

* *Operations – a class, responsible for making requests to our Azure-database.*

**ScoresForm**

* *Program – start-point of running a “score table”-form.*
* *EnterYourNameForm – class, responsible for displaying data from database.*