1. Introduction

This document describes interface of the simple frames and units – based game engine which allows user to know only positions and images to draw graphics part of game. We try make this API maximally programming language independent and abstract to allow implement it in any one.

This API doesn't describe what kind of game it should be, and how it should be implemented, it describes only way to connect graphical part of game with its engine.

2. Data types description

GameAPI uses some data types which could be implemented in different ways in different languages. Some of these types can be native types for some languages.

Coordinates – it's data structure which represents some rectangle unit position and size. It must have at least 4 numeric values which represents two points: bottom left angle (x_1,y_1) and top right angle (x_2,y_2).

Possible implementation:

```
struct Coordinates
{
  int blx, bly, trx, try;
};
```

List<type> – it's usual list which appears in most of programming languages. But it also can be usual dynamic array or some other similar data structure.

Pair<type1, type2> – it's pair of values of two types, e.g. std::tuple in C++.

It can be implemented in C like this:

```
struct
  Pair_type1_type2
{
  type1 t1;
  type2 t2;
};
```

Note that there is no template requirement by API, types required by API's function can be hardcoded in structure.

Image – it's language and framework dependent type which represents image which can be drawn by GUI of game. e.g. Image class in C# Windows Forms.

3. API functions

void nextFrame();

This function must cause matching of the next frame (next state) of the game. It is something like send synchronization impulse to the game's object.

List<Coordinates> getAllUnitsCoordinates();

It must return list of coordinates of all items which user should know about in the current game state.

List<Tuple<Coordinates, Image>> getAllUnitsCoordinatesImages();

It's similar to **getAllUnitsCoordinates** function, but returns positions of game items to draw paired with images associated with each unit. It means pairs of position and size – image to draw at this position with this size.

bool playerIsAlive();

Returns true if player is alive. If there are more than one players, returns false if no one is alive.

Coordinates getPlayerPosition(int playerNumber = 0);

Returns position of the player. If there are more than one players, this function must have parameter with index of player asked for position.