COP 290 Assignment 3

Space Invaders

Faran Ahmad
2013CS10220

Kabir Chhabra
2013CS50287

Kartikeya Gupta
2013CS10231

Prateek Kumar Verma
2013CS10246

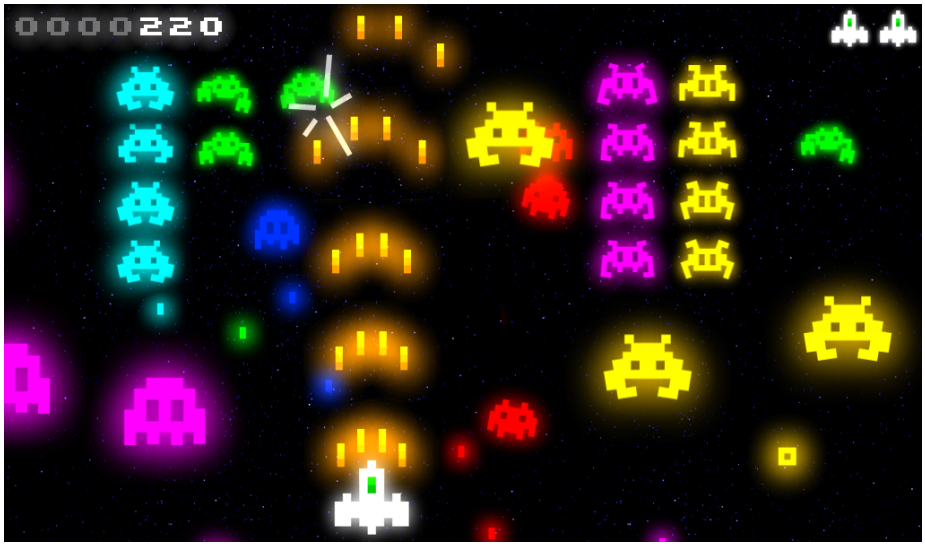
March 2015

Contents

1 Objectives

Design a game which is :

- Multi-player on-line without a central server.
- Has a artificial intelligence component.
- Is an action game and not a simple board game.



2 Overall Design

The game which we will build is space invaders. It involves the player controlling a space ship and shooting down aliens. The aliens will fight back with bullets and missiles. The player has a limited number of lives and has to score the maximum in them.

1. The application would be programmed in C++.
2. The GUI part would involve OpenGL.
3. UDP sockets will be used for network data transfer.
4. POSIX threads will be used to run the network and back-end in parallel.
5. Inter thread synchronization would be done using mutex lock.

3 Sub Components

3.1 Back End

TODO: FARAN

3.1.1 Alien

Listing 1: Class Parameters for Alien

```
class Alien
{
private:
    float XPos;              // X coordinate
    float YPos;              // Y coordinate
    float Angle;             // Orientation angle
    Color ColorOfAlien;      // Color
    int Level;               // AI difficulty level
    int PresentLives;        // Lives left
    int NumberBullets;       // Bullets fired per shot
    int NumberMissiles;      // Number of missiles left
    int AlienType;           // Type
};
```

3.1.2 Ship

Listing 2: Class Parameters for Ship

```
class Ship
{
private:
    float XPos;              // X coordinate
    float YPos;              // Y coordinate
    float Angle;             // Angle
    std::string Name;        // Name of player
    Color ColorOfShip;       // Color of ship
    int Lives;               // Lives left
    int Score;               // Score of player
    int Multiplier;          // Multiplying factor
    int Kills;               // No. of kills
    int Id;                  // Player id
    int NumberBullets;       // Bullets fired per shot
    int NumberMissiles;      // Number of missiles left
    int AILevel;             // Level of AI
};
```

3.1.3 Color

Listing 3: Class Parameters for Color

```
class Color
{
private:
    float R;                    // Value of R component
    float G;                    // Value of G component
    float B;                    // Value of B component
};
```

3.1.4 Bullet

Listing 4: Class Parameters for Bullet

```
class Bullet
{
private:
    float XPos;                 // X Coordinate
    float YPos;                 // Y Coordinate
    float VelX;                 // Velocity X
    float VelY;                 // Velocity Y
    Color ColorOfBullet;        // Color
    int ShipId;                 // Id of ship fired from
    bool TypeAI;                // If AI bullet
    bool TypePlayer;            // Player type
};
```

3.1.5 Board

Listing 5: Class Parameters for Board

```
class Board
{
private:
    std::vector<Ship> VectorShips;      // All ships
    std::vector<Bullet> VectorBullets;  // All bullets
    std::vector<Alien> VectorAliens;    // All aliens
    double DimensionPosX;               // Dimensions + x
    double DimensionPosY;               // Dimensions + y
    double DimensionNegX;               // Dimensions - x
    double DimensionNegY;               // Dimensions - y
};
```

3.2 Artificial Intelligence

TODO: KABIR

3.3 Graphics

TODO: KG

3.4 Network Part

TODO: SOCCER

4 Interaction amongst Sub Components

4.1 Back-end and UI

The GUI will use the data structures directly. The "Board" class will be available to generate the display on the screen. The user inputs from keyboard and mouse will be used to generate changes in the class object.

4.2 Back-end and Network

TODO KG

5 Testing Of Components

5.1 General Unit Tests

Listing 6: Class Parameters for Test

```
1 class Test
2 {
3 private:
4     bool verbose;                //If test is to be conducted
5     std::string description;     //String description of the test
6     bool isPass;                 //Boolean if the test has passed
7     void PrintPassFail(bool);    //Prints the status of the test
8 };
```

We will use the aforementioned class "Test" to perform unit tests on the different files created. This will ensure that all the functions work correctly against some test cases.

5.2 Graphics

To test the graphics component, we will create aliens and ships at chosen positions. The positions should change appropriately based on user inputs. Bullets should also be fired from the correct positions. When a bullet hits an alien or a ship, proper GUI effects will be added.

5.3 Artificial Intelligence

TODO KABIR

5.4 Network Component

TODO SOCCER

5.5 Overall Testing

For overall testing of the game, we will play the game with as many players as possible and verify the smooth functioning of AI and network component. The network for some clients will be shut down suddenly to check if the transitions for the AI and others are smooth.

6 Extra Features

6.1 Competitive Multi-player Mode

In this multiplayer mode, the players will be fighting each other. They will try to shoot each other down. Aliens will not be present in this mode. AI players will be present in this.

6.2 3D Game-play

The game will be taken to 3D in which the aliens and ships can be viewed in a 3D perspective. The camera position will be adjusted by the user based on input from mouse.

6.3 Sound Effects

Sound effects will be added for the entire game-play. When bullets are fired or some shoot down takes place, appropriate sounds will be played.

6.4 Replacement of Player by AI

In case of network failures, the player whose network has gone down will get replaced by an AI player of the same level as the player was. This will ensure completely seamless transition in case of network outages. Once the network comes back on, the player will replace the AI control.