Ohjelmointi 3: Nysse

Mikko Kemppi, Olli Pettersson

Short description

This game is a top-down-view arcade game, where the goal is to bomb as many Saab 900 -cars, Kela-stops and enemy planes as possible with your helicopter, while avoiding Red Baron's evil planes and collecting powerups.

Technical disclaimer:

This game runs choppy on the remote desktop and contains sounds that cannot be heard when playing in the remote desktop. It is suggested to play this game locally.

Controls:

WASD: Control helicopter

Space: Drop bomb

Esc: Close game

Game process:

The program starts with a setup dialog, where the player inputs their player name and selects difficulty. Game cannot be started if player name contains some of the following characters: ",","/","*", or the difficulty is not selected. Incorrect player name displays a notification to the user. Difficulty determines the number of enemy planes, the size of the bomb explosion and the amount of powerups. When everything is set up correctly, the player can press start.

After pressing start the game starts. The player can now move the helicopter with WASD keys. Left top corner displays stats about the current game; remaining lives, amount of dropped bombs, current score and player name. Player starts with three lives on all game modes except instagib. The helicopter cannot turn 180 degrees with one key press (e.g. directly from west to east). Pressing space drops a bomb in the current location. There are several different actors in the game screen, they are listed below:

- Red plane: Damages the player if it collides with the player's helicopter, gives 15 points when destroyed
- Kela flag: Acts as a Saab stop, Saabs spawn and stop here, gives 1 point when destroyed
- Saab 900 car: Moves according to bus lines and stops at Kela-stops. Cars have three different variations, determined by the number of passengers inside. Green cars have 0-4 passengers, yellow cars have 5-9 passengers and red cars have over 9 passengers. Cars give as many points as there are passengers when destroyed. Moving is determined in logic.
- Powerups: There are two kinds of powerups that activate when bombed: plane reset and helicopter upgrade. Plane reset resets the number of planes spawned every

second to the 0 the start of the game. Active planes won't be destroyed. Helicopter upgrade gives the player a better helicopter that moves faster, is smaller and can do 180-degree turn.

- Bomb: Bombs dropped by player, explode after 3s at the spot where it was dropped and destroy all other actors inside actor radius. If explosion hits player's helicopter, one life is reduced.

The player is supposed to bomb everything it can while staying alive. Player gets damage if the helicopter hits a plane, they get caught in their own bomb explosion, or they hit the map edge. The number of enemy planes increases every five seconds. Easy, medium and hard games have three lives, instagib has one. After the player loses all their lives, the game stops and a new game over dialog is displayed. Game over dialog displays last games score, amount of dropped bombs and highscore for the selected difficulty. Highscores are displayed in descending order based on player score, and only the highest score of each player is displayed. After viewing the end game screen, the player can quit the program. There is no restart button.

Additional functionalities

This game contains several additional functionalities, which are listed below.

Steady game update

The game state is updated constantly, not only when something is pressed. This is made by connecting a timer with 10ms timeout to update function, which updates the locations of all necessary actors.

Scrollable map

The game is centered on the player character, and the maps scrolls according to player movement. This is made by centering game window to the player on every update tick.

Steady movement of player character

The player character moves automatically and smoothly constantly to the selected direction. This is made by updating player location and visual on every update.

Passenger counts

Passenger counts are displayed visually by changing the color of cars. Green cars have 0-4 passengers, yellow cars have 5-9 passengers and red cars have over 9 passengers.

Live stats display

Game displays current sessions stats. Player's score, amount of dropped bombs and current lives are displayed at the top left corner of game screen.

Highscores

Every session's score and player name are saved to a .csv file in the same folder as the game .exe. This file is parsed and the top score for each player for current difficulty is displayed at the end game screen, sorted in descending order. Each difficulty is saved to a separate file

and all sessions save a new entry to a file, but only the highest is displayed. First game of each difficulty generates a new file and subsequent games append to these files.

Player upgrades

Player can pick up upgrades and powerups by bombing them. There are two kinds of powerups; one resets the increasing number of planes to the initial start value, thus making the game easier momentarily, the other one upgrades the helicopter to a better one.

Sounds

This game contains several different sound effects for different actions, and a background music. These cannot be heard on the remote desktop.

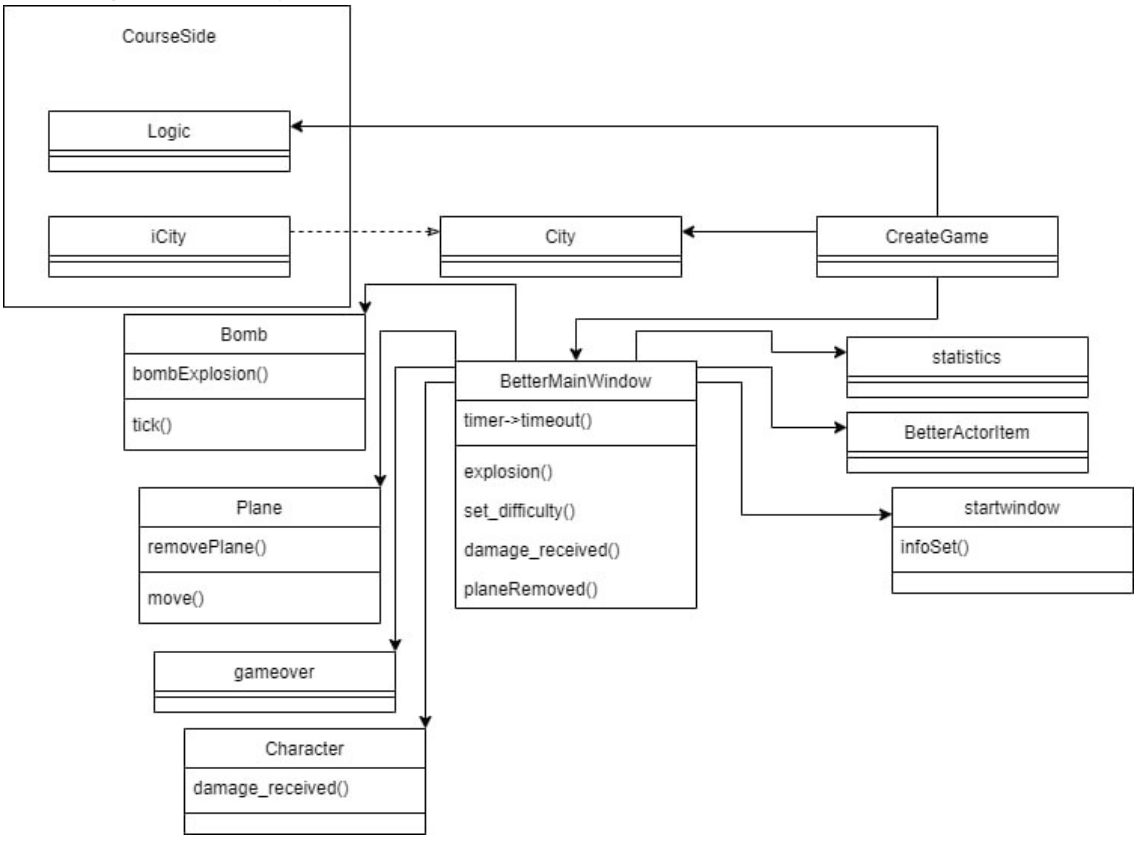
Difficulty levels

This game can be played on four different difficulty levels: easy, medium, hard and instagib. Easy, medium and hard have three lives, and a differing amount of planes spawned, different amount of plane reset powerups and a different bomb explosion size. Each difficulty level saves highscores to its own file.

Turning of game (peli kääntyy :DD)

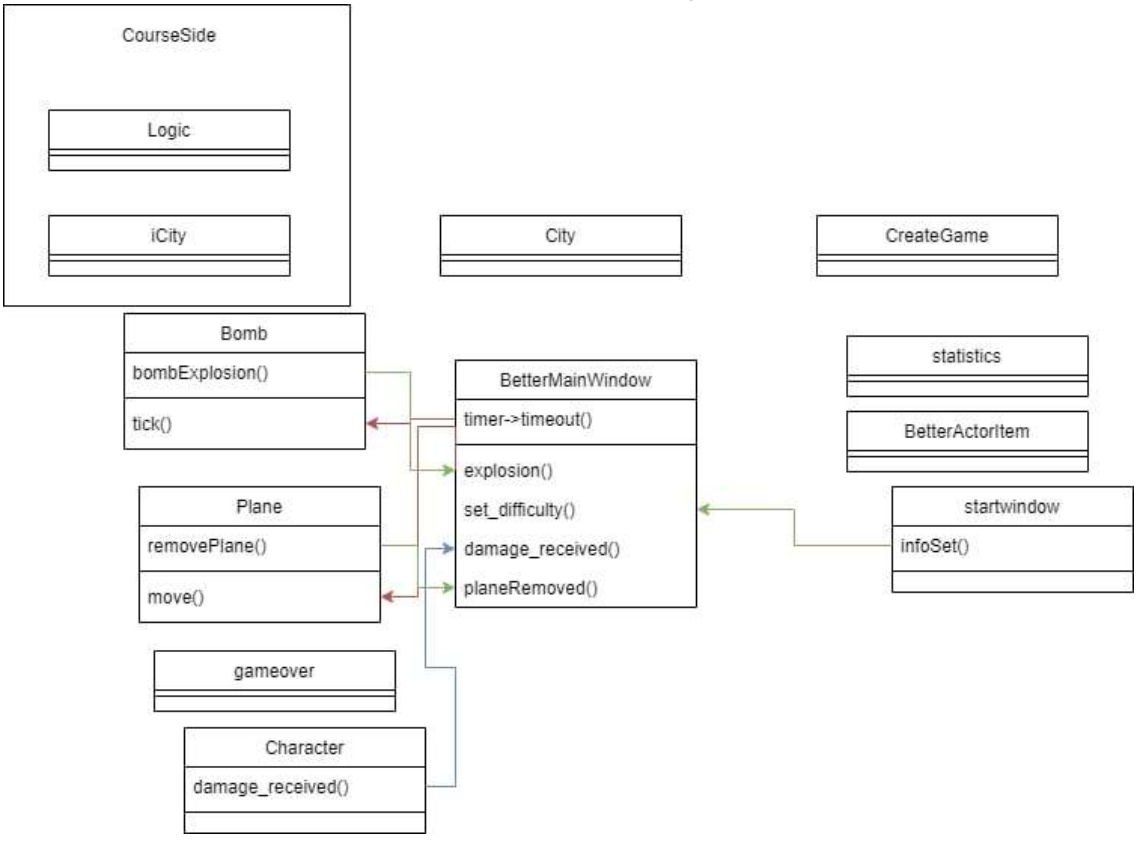
Game can be rotated by pressing the R-button.

Class diagrams and responsibilities



CreateGame class contains pointers to city, logic and bettermainwindow. City is inherited from Interface::ICIty, but has also pointer to bettermainwindow. Mainwindow keeps track of

betteractoritems which are stops and busses given by logic. Betteractoritems are saved to map with corresponding actors, so finding and removing right Betteractoritem by actor is possible. Betteractoritems are keep track of position, actoritem status and points. Betteractoritem class also sets color for actoritem if actoritem is Nysse. Passengers will not be added to scene to avoid lag, but they are accounted for in other ways. Bettermainwindow has pointer to statistics class, which keeps track of stats of the game. Character pointer points to character class, character class is moved every time update is called. Character can be transformed from normal helicopter to apache helicopter if powerup has been hit by bomb. Character also emit signals if character crashes to edge of map, red plane or own bomb. Planes are added to map from mainwindow, plane move functions are connected to timer ticks so planes will fly horizontally or vertically. Plane also knows if it's destroyed. Bomb class is called by character if space is pressed. Bomb gets position and direction from character. Startwindow is used inside Bettermainwindow constructor and will set info for statistics. Gameover class is called when Bettermainwindows gameover function is called.



This picture contains all connected signals and slots in project. All connections are connected from or to Bettermainwindow.

Testing

This project contains unit tests for the statistics class. No other tests were deemed necessary, and a lot of testing was done during runtime.

Work distribution

Work distribution was defined at the start of the project, and we followed it closely. Olli was responsible for the game itself, drawing actors and creating the game logic, Mikko was responsible for everything else around this, e.g. statistics, start and end window and unit

tests. Workload ended up being decently even, and work was mostly done as pair programming, or at least in the same room. This made communication easier and problem solving faster.

Known problems

If bus moves at the edge of moving game screen at the same speed as the screen, the bus leaves a weird choppy trail. Does not affect gameplay. Start window sometimes displays invalid player name error, even though the name is viable. Game can be started despite this, as the start button works correctly.