NowItWent - Tampere Adventure Game Documentation v. 1.1

Robert Cantaragiu – robert.cantaragiu@tuni.fi

Olli Lääkkö – olli.laakko@tuni.fi

1. Preface

The goal of the project is to implement a game – named NowItWent – where the busses and passengers moving around downtown Tampere act as a part of the game. The game is a joint effort between the student group and the course staff. The staff has implemented the busses, basic passengers and the logic that moves them around town. The student team implemented the functionality for the city itself, the playable figure and the main window.

We have implemented the minimum and intermediate requirements, as well as some extra features. Section 2 describes the software architecture and class diagram. Section 3 describes the implementation of the extra features. Section 4 provides information about the planned and actual division of work. Section 5 describes the game play and user manual.

2. Software architecture

The project contains two namespaces:

CourseSide – Implemented by the course staff.

StudentSide - Implemented by the student group.

Additionally, the course staff implemented some interfaces and the student group implemented a dialog window and unit-tests that are not included in the namespaces.

Next we describe the classes that were used in the project.

CourseSide namespace classes:

- Nysse class – implements the functionality of the bus objects;
- Passenger class – implements the functionality of the passenger objects;
- Stop class - implements the functionality of the bus stop objects;
- Logic class – handles the data reading from Resources. Handels the non-playable actors movement.
- SimpleActorItem class – draws a movable object on the map.
- Location - Defines a class that contains methods for handling location. (coordinates)

CourseSide interfaces:

- IActor - Defines a single actor (= an object acting in the game)
- ICity - Defines an interface that represents the city's operations.
- IStatistics - Defines an interface for scoring statistics.

StudentSide namespace classes:

- Player class – implements a player object that is drawn on the map.
- Actor class – implements graphics for buses, passengers and stops.
- Bomb class - implements bomb objects that are drawn on the map.
- City class – Inherited from ICity interface. Implements:
    - setBackground -Setting the map background.

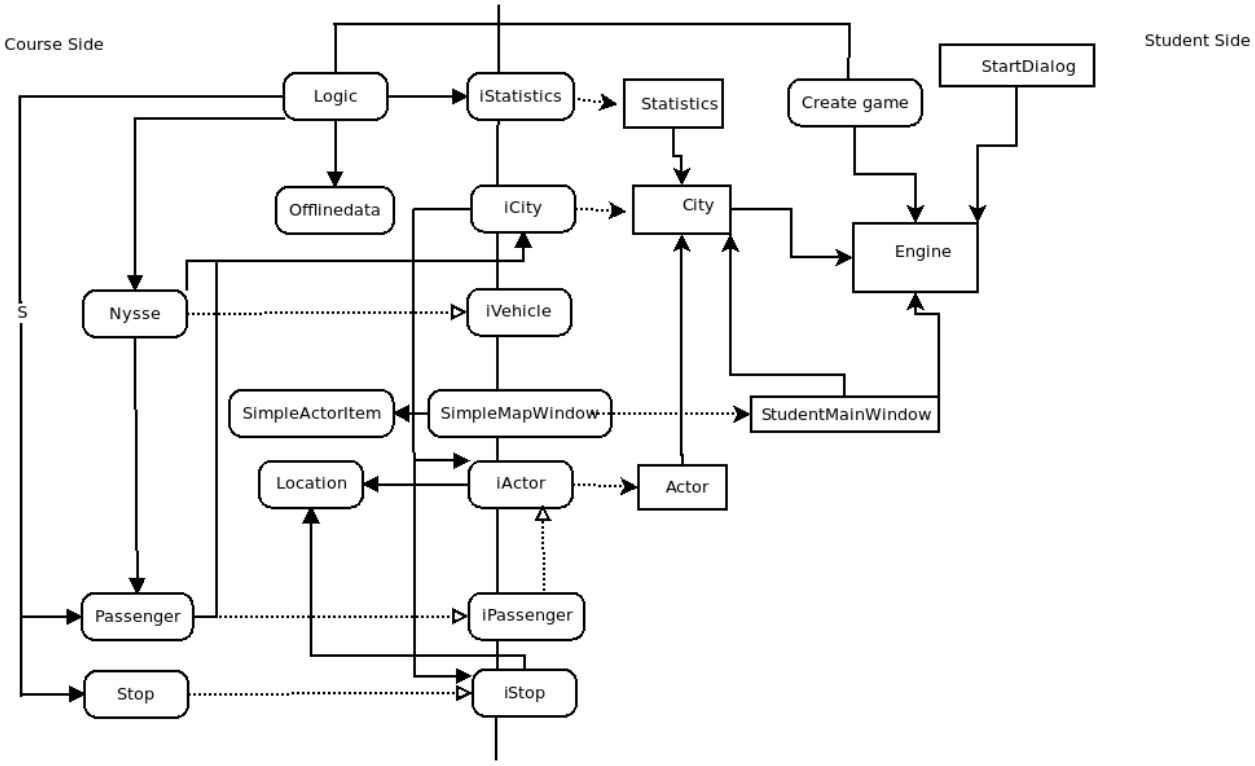
    - addActor - Adding actors (stops, busses, passengers) to the map.
    - removeActor - Removing actors.
    - getNearbyActors - Finding actors in the proximity of a specified location. Returns a vector containing them.
    - isGameOver - Ending the game. Stops all movement of non-playable actors.
    - actorMoved – Moves the actors, updates their coordinates. The function is constantly called from the CourseSide logic so it also listens for a playerEat_ truth value that tells whether it should call getNearbyActors and remove the specified actors in the returned vector.
- startDialog class – Opens up a dialog window when the game is launched. Contains:
    - A story about the game
    - Gameplay timer choice. Can be 2, 5 or 10 minutes.
    - Background choice (big map or small map)
- Engine class – creates the dialog window, reads it's inputs and then creates the game mainwindow based on these inputs.
- StudentMainWindow class – Implements a student-made mainwindow. Contains the functions:
    - setSize - Setting the size of the map.
    - setPicture – drawing the map (big or basic) on centralwidget.
    - addPlayer - spawns a player object to the map. The location is chosen at random.
    - addBomb - spawns a bomb object to the map. The location is chosen at random.
    - addBombs - calls addBomb based on the map size.
    - setBombVisbility - makes bombs visible/invisible.
    - keyPressEvent – Listens for keypresses: W, A, S, D moves the player object while Spacebar removes all edible actors (busses and passengers) in the players proximity and adds score.
    - addActor – draws actor objects on the map. The location comes from the CourseSide logic
    - addNysse – draws Nysses by calling addActor. Saves the object to a map made from a shared pointer of the IActor interface and a pointer to the respective CourseSide::SimpleActorItem.
    - addPassenger - draws Passengers by calling addActor. Saves the object to a map made from a shared pointer of the IActor interface and a pointer to the respective CourseSide::SimpleActorItem.
    - addStop - draws Stops by calling addActor. Saves the object to a map made from a shared pointer of the IStop interface and a pointer to the respective CourseSide::SimpleActorItem.
    - removeActor – Removes an actor item (busses or passengers) from the map. Adds score when an actor is removed
    - findActorInMap – searches for an inputted actor object in the nysses_ and passengers_ maps. Returns a truth value.
    - updateNysseCoordinate – updates the bus coordinates.
    - updatePassengerCoordinate - updates the passenger coordinates.
    - updatePlayerCoords – updates the player coordinates.
    - return_logic – returns a pointer to the logic object created from the CourseSide::Logic
    - returnplayer - returns a pointer to the player object created from the StudentSide::Player
    - returnPlayerEat – returns a playerEat_ bool value that the city class constantly listens for.
    - switchPlayerEat – Changes the playerEat_ bool value after city has removed all actors in the player's proximity.
    - isBigMapSelected - returns a bool value depending on whether the big map is shown in the main window or not.
    - startTimer – Starts the timer in the game.

- updateScore – updates the score and displays it on the score LCD widget in the mainwindow.
- updateTime – updates the time (minutes and seconds) every second on the LCD widgets in the mainwindow.
- gameOver – Ends the game if the time is up or if the player goes out of the map. Stops all player movement and disables him. Sets gameOver_ value to true.
- returnGameOver – returns a gameOver_ bool value that the city class constantly listens for.

- Statistics class – collect data about game actions. Includes
  - GivePoints- return player poins
  - PassengerDied – called every time when passenger die. Save earned points.
  - MorePassenger – keep a record of passengers in game
  - NysseRemoved – called every time when player eats the bus. Save earned points
  - NewNysse – keep a record of buses in game
  - NysseLeft – same than last one, but opposite direction
  - ClearPoint – remove earned points.
  - GetPassengers – return number of passengers
  - GetNysses – return number of buses

UnitTests tests statistics class functions. Each of statistics class functions are tested there.

Class Diagram:



3. Additional features

- Even screen updates: The player and other actors move smoothly. Instead of immediately jumping from one place to another, the player is moved smoothly via keyboard and the other actors are moved with a suitable speed.
- Minimal screen update: When an actor's state is changed, only its surroundings are updated in the game area instead of drawing the entire map.
- Own feature: Add score when player eats other actors. Then displays it immediately on the LCD widget.
- Own feature: Game timer. Can be 2, 5 or 10 minutes. It is displayed on the LCD widget in realtime.
- Own feature: Can choose map. The map size can be big or basic.
- Following the game state: the time and game score is shown in real time on the mainwindow.

Extras:

- Game story in dialog window.
- Colorful dialog and mainwindow

4. Work Division

Initially, we wanted to do a different game however as time passed by, we realized that it might not be feasible, so we switched to the current idea of the game. Initially the work division was as follows:

Robert – draw actors, draw player, player movement, dialog window, student main window (together)

Olli – draw stops, actor movement, unit tests, student main window (together)

The work division turned out to be:

Robert –Student mainwindow, draw player, player movement, player eating functionality (key press and actor remove from map), dialog window, documentation. For the second version: unique graphics for each actor on the map, bombs and their functionality.

Olli – draw actors, make actors move, actor remove, unit testing, statistics class, player eating functionality (get nearby actors), initial files creation.

5. Game Manual

The program starts with a settings dialog where you can choose the map size (big or basic) and play time (2, 5 or 10 minutes). The dialog also contains a game story and a tutorial. After entering the settings, the program moves to the main window of the game where the map of downtown Tampere is shown as the playing field. On the map there are busses, bus stops, bombs and a playable figure.

The busses move around the city in real time according to real bus lines and timetables. The passengers get on and off the busses at the busstops. The player controls the playable figure that can move around the map and eat (remove) other actors in the game. The map also contains bombs that are displayed every 10 seconds. If the player steps om a bomb he dies. The game ends if the time ends, the player steps on a bomb or if he goes out of the map.

Playable figure:

Keyboard actions:

W – moves the figure up

A – moves the figure to the left

S – moves the figure down

D – moves the figure to the right

Spacebar – eats busses and passengers when the figure is on them.

Busses:

 Move around the city, pick up passengers and moves them to other stops. Gives 50 points when eaten

Passengers:

 Move with bus or stay in stop (easy prey). Gives 3 points when eaten

Stops:

 Passengers that are not in buses stay in stops.

Bombs:

 Displayed every 10 seconds. If the player steps on a bomb the game ends.