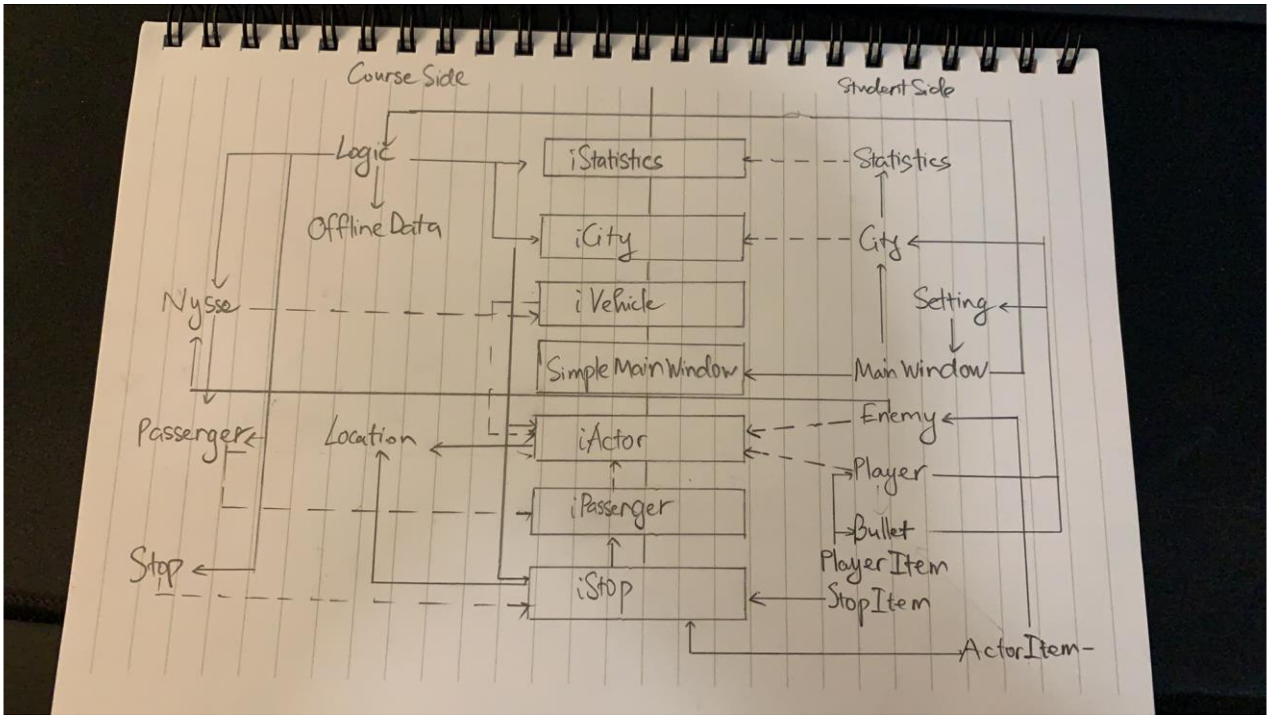
NowItWent Documentation

1. Structure of the StudentSide implementation:



2. Responsibility of key classes:

   2.1. City class:

   - Set the background image of the game using the provided images and QImage class.
   - Set the game state, game over state.
   - Add stop and other actors (Nysse, Passenger, Own Actor) to the game.
   - Remove actors (Nysse, Passenger) from the game.

   2.2. ActorItem class:

   - Add and remove actor graphics items to the game scene (Nysse and own actor graphics item).
   - Set and update the position of the graphic to the scene.

2.3. PlayerItem class:

- Manage the movement and direction of the player actor.
- Manage the action (shooting).
- KeyEvent handling.
- Set and update the position of the graphic to the scene.

2.4. Enemy and Player class:

- Manage the existing state of the actor.
- Get location and set location (move)

2.5. Bullet class:

- Move according to Player Item current direction.
- Check collision with other QGraphicsItem. If collided, remove the item and itself from the scene.

2.6. Statistics class:

- Keep track with statistics such: score, number of Nysses, number of Passengers in real-time
- Return the statistics to MainWindow

2.7. Setting class:

- Get input from users and output it to MainWindow

3. Project functionality

- At first, the user chooses from the drop-down menu in setting UI. The param will be passed to MainWindow class.
- MainWindow class will initialize the game view, score panel, time panel in the UI. Then it will call the Logic class and City class to set the background image, init the game.
- The main functionality of the City class is to control the scene animation and manage both ActorItem class and Actor class.
- For the ActorItem management, City class is called in Logic class and takes a shared pointer to an Actor as parameter, then City will pass the actor shared pointer

down to ActorItem class to let ActorItems keep track with its Actor state (location, isRemoved).

- The PlayerItem class takes shared pointer to Player Actor as parameter and hero type from MainWindow, which received from setting UI. PlayerItem class have a KeyEvent handler to control the player figure, each time a key is pressed, it will send an event signal, if it was an arrow key or WASD key, the PlayerActor will update its location using move function, and then PlayerItem will setPos according to the location. Otherwise, it will stay still in the game scene.
- Bullet is an QGraphics Item, which will spawn if the key pressed is Spacebar, it will move base on the current direction of the PlayerItem. If it collided with other GraphicsItem, it will call the collidingItems() method from QGraphicsItem and check if it a Nysse or an Enemy, if true, set its actor isRemoved is true. Then the City will take care of the rest.
- Statistics keeps track of the number of buses, the number of passengers, score in real-time and return them to MainWindow for display purpose.

4. Extra features

- Passenger amounts: The passenger amount is displayed on Nysse's icons, but not on stops since some areas may have a high density of stops, such as Keskustori. Therefore, display the number of passengers at the stop can cause some confusion. However, you can see the number of passengers at the stop by move the mouse cursor to the stop.
- Following the game state: The statistics collected during the game are shown in real- time update (score, Nysse left, New Nysse coming).
- Passed time update.
- Scene-controlled animation: We override the QGraphicsScene::advance () in the city.hh to update the game smoother, running at 30fps. This slot advances the scene by one step, by calling ActorItem::advance (int phase) for all items on the scene. This is done in 2 phases: In the first phase, all items are notified that the scene is about to change, and in the second phase all items are notified that they can move. In the first phase, ActorItem::advance (int phase) is called with phase 0, and 1 is passed in the second phase.
- Option to choose many characters and different bullets.

5. Division of work

- Quan Do: other specialized actors, bullet, documentation
- An Ton: City, Statistic, Unit tests
- Both: better gamewindow, create game, setting class

The actual works is the same as the agreed division of work.

6. Gameplay

- At the beginning of the game, there will be a Welcome window with the name of the game Avenger game to let you choose between setting and quit game.
- If you choose Setting, a setting UI will appear to let user choose level, mapsize and characters (Iron Man, Captain America or Thor), each character comes with different bullet graphics. In the easy mode, there will be an enemy (Thanos) appear every 500 ticks (about 15 seconds in real-time) while in normal mode, there will be 2 Enemies for you to destroy. Destroy an Enemy gives no reward, but if you don't, after jumping to a random location 3 times, it will disappear and destroy about half of the current Nysse on the map, reduce your chance to get a high score. Easy mode lasts for 1 minute, while normal mode lasts for 3 minutes.
- After the setting UI, you will be brought into the MainWindow, where you can start to play the game. You can use either arrow keys or WASD keys to move the player and Spacebar to shoot. You only get the score if your bullet hits the bus. Each bullet destroys one bus only. You cannot move out of the game window.
- Whenever you destroyed a bus, the score increases by 1, the more passenger on the bus, the higher score you received. Each passenger also increases the score by 1.

7. Known bugs or missing features

- Missing features: Although there is an option to choose between, small map or a scrollable map, we have not implemented the scrollable map successfully.
- Known bugs: Even though we set flags and setFocus for the player graphical figure, but at the begin of the game, the player cannot move since it's not focused, so users should click on the player graphical figure, then they can play normally.