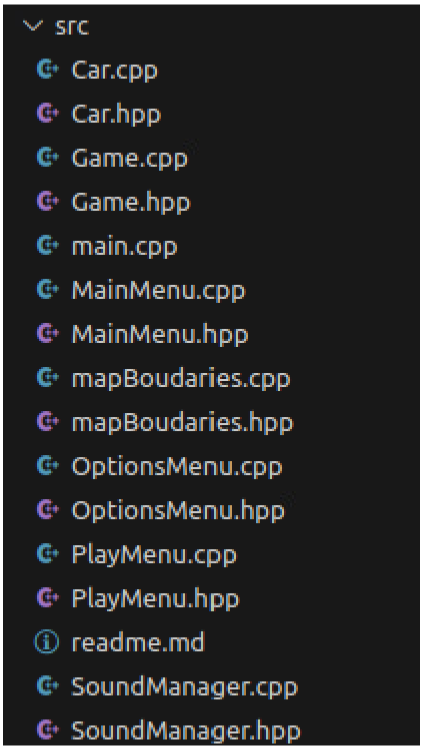
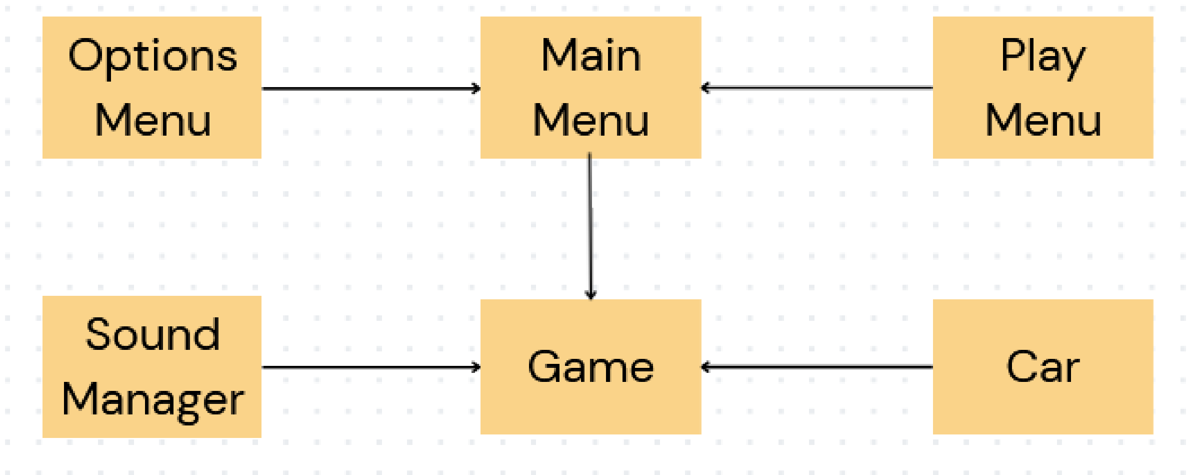
1. **Overview**: the software is able to successfully operate like the original micromachine game including its movement dynamics, bots, win condition, SFX, boosts, different vehicles, different terrain handling and boundaries. The software doesn't provide splitscreen, a level editor and weapons.
2. **Software structure**:

   This is what our structure is composed of (along with the SFML library):



```
src
  Car.cpp
  Car.hpp
  Game.cpp
  Game.hpp
  main.cpp
  MainMenu.cpp
  MainMenu.hpp
  mapBoudaries.cpp
  mapBoudaries.hpp
  OptionsMenu.cpp
  OptionsMenu.hpp
  PlayMenu.cpp
  PlayMenu.hpp
  readme.md
  SoundManager.cpp
  SoundManager.hpp
```

As for the relationships between the architecture, you can see it in this diagram:



3. **Instructions for building and using the software:**
    - **Make sure you have SFML installed (but should be provided in the file already)**
    - **Make sure you have VScode installed with all the right extension to run C++**
    - **Choose a folder in a Linux environment where to put the software**
    - **mkdir Any_name_you_want**
    - **cd Any_name_you_want**
    - **Make sure the repository is empty**
    - **Make sure you have an SSH key and that you have the proper authorizations to use it**
    - **git clone the repository with "git clone git@version.aalto.fi:cabrask1/micro-machines-2.git ."**
    - **cd libs**
    - **cd box2d**
    - **mkdir build**
    - **cd build**
    - **cmake ..**
    - **make**
    - **cd ~/Any_name_you_want**
    - **mkdir build**
    - **cd build**
    - **cmake ..**
    - **make**
    - **cd ..**
    - **code .**
    - **Start debugging**

Basic user guide:

Upon starting the game, the main menu will appear. To select something you can use the arrow keys on your keyboard and press enter to go forward. In the actual game, there are only 4 control keys and 1 boost key:

- **Top Arrow:** Accelerate
- **Bottom Arrow:** Decelerate
- **Left Arrow:** Turn left
- **Right Arrow:** Turn right
- **Space:** Boost key

In order to win you need to make 3 laps around the racing course.

4. **Testing**:

    For testing, we used the google test extension along with its documentation and methods. We focused on anything that is relevant to the software, even the details, to ensure that everything works correctly.

5. **Work log**: Work was evenly distributed between the members of the group as we can see in this work log divided per sprint

Sprint 1 (10 hours):

- **Designing the gameplay:**

  Status: Completed

  Responsibility: Team

- **Setting up the workspace environment:**

  Status: Blocked

  Responsibility: Team

- **Figure out how to use libraries:**

  Status: Blocked

  Responsibility: Team

- **Figure out the structure of the code:**

  Status: Blocked

  Responsibility: Team

Sprint 2 (10 hours):

- **Setting up the workspace environment:**

  Status: Blocked

  Responsibility: Team

- **Figure out how to use libraries:**

  Status: Blocked

  Responsibility: Team

- **Figure out the structure of the code:**

  Status: Blocked

  Responsibility: Team

Sprint 3 (20 hours):

- **Setting up the workspace environment:**

  Status: Success

  Responsibility: Team

- **Figure out how to use libraries:**

  Status: Success

  Responsibility: Team

- **Figure out the structure of the code:**

  Status: Success

  Responsibility: Team

- **Create main menu:**

  Status: Ongoing

  Responsibility: Petteri

- **Implement bots:**

  Status: Success

  Responsibility: Jemma and Kevin

- **Implement movement:**

  Status: Success

  Responsibility: Team

- **Use and edit maps from micromachines:**

  Status: Success

  Responsibility: Jemma

- **Pick car sprites to use:**

  Status: Success

  Responsibility: Kevin

- **Choose SFX and implementation:**

  Status: Ongoing

  Responsibility: Kevin

Sprint 4 (20 hours):

- **Figure out how to use libraries:**

  Status: Success

  Responsibility: Team

- **Create main menu:**

  Status: Success

  Responsibility: Petteri and Michael

- **Choose SFX and implementation:**

  Status: Blocked

  Responsibility: Kevin

- **Implement boundaries:**
  Status: Success

  Responsibility: Petteri

- **Implement boost:**

  Status: Success

  Responsibility: Michael

- **Implement oil spills:**

  Status: Success

  Responsibility: Petteri

- **Implement different terrain handling:**

  Status: Blocked

  Responsibility: Jemma