



I, Game & Robot Hackathon

AI-Land Mobile Game

Game Features



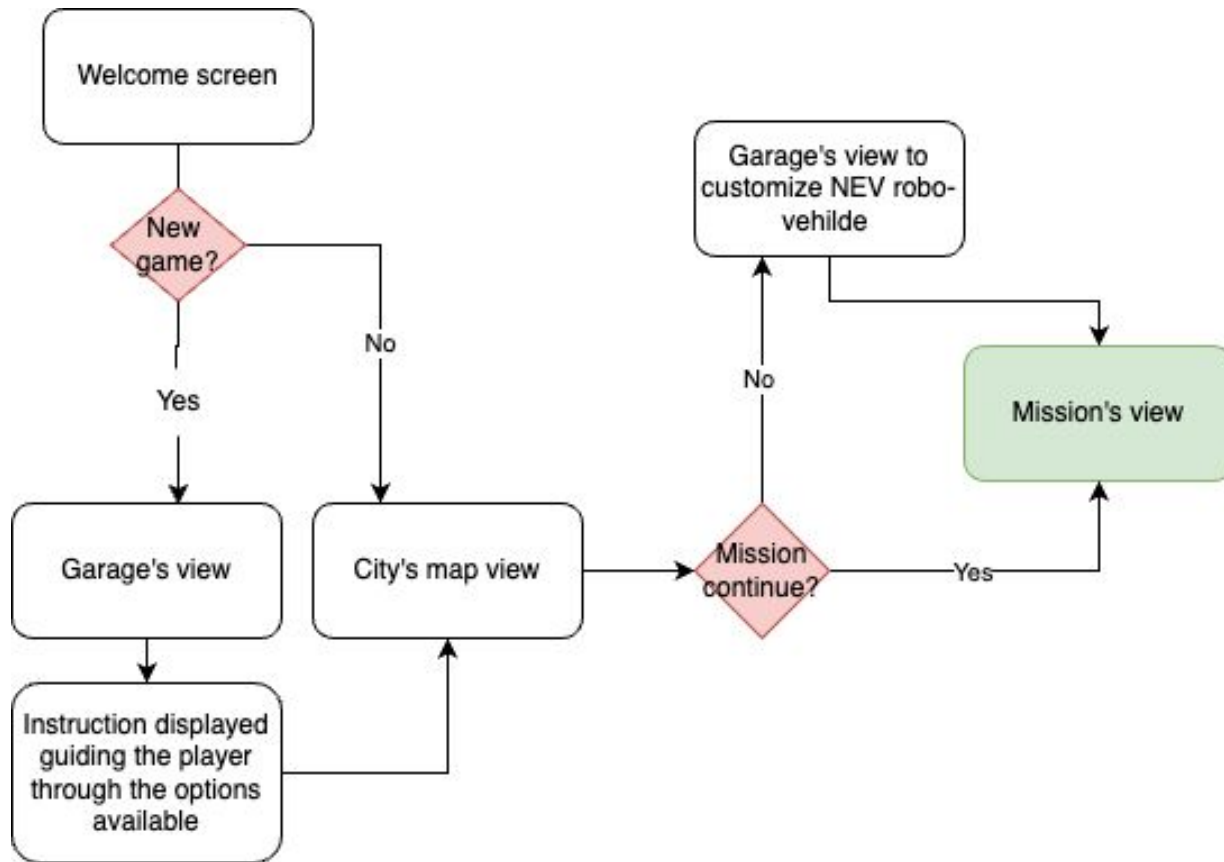
@agwisniewska
@rafalkowalski

Game flow

Typical game flow will be linear. Players will be given a portion of the narrative, **which would be a reason to do missions**, and once the mission is over the same basic sequence will happen with some new challenges to keep the game interesting. Scattered between the usual game flow conventions will be special set pieces, which will behave as a buffer between challenges in the main plot and will mostly house a cutscene, or fleshing out plot

The main flow of the game is that a player does missions, and while doing this it teaches the robo-vehicle new skills, as he is responsible for supervising the tests of newly-designed robo-vehicles, which target to be fully independent.

The game starts with presenting a main goal to the player, with an exception when it comes to the activities of the enemies. That part will be hidden from the player until it happened for a first time, and even in that case, not whole explanation should be revealed to the user at once, but rather step by step while progressing with the game. That way it would be a surprise for the player to see that something is going on, and also start looking for a mitigations of the bad impact they have on the whole progress in the game.

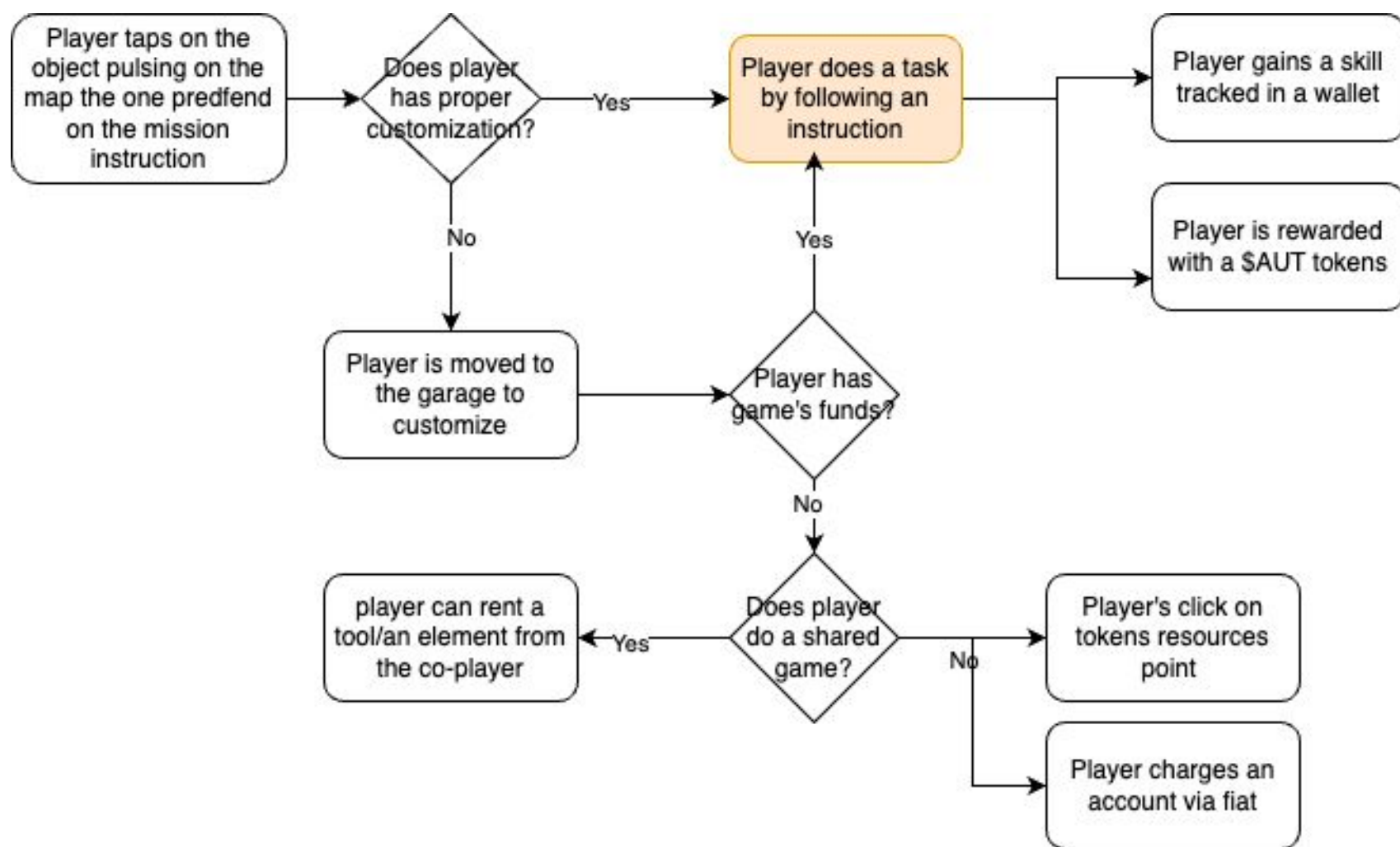


Game flow - Mission

The mission is announced to the player with the prerequisites which may be helpful as a text message/email which is sent to the player, while the details are unknown, as are supposed to be discovered by the player while doing and react accordingly (eg. a message says that for the mission a shovel and thick tires are needed) so the player could prepare himself in advance, but as it's human to be mistaken from time to time, it may happen that a player receives an incomplete message or/and something changed at the latest moment, or something happens in the city and other parts would be needed as well. In such case a player would still be able to react, similarly when it comes to the case where the players can't afford for the things they need and need to collect some tokens first.

Game Flow - Player

Player starts a game with a some amount of tokens with an instruction to invest into few areas which would be the source of the token (a game coin) through the whole gameplay. It would be solar panels, energy stations and different sensors. Tokens would be also earned by gaining new skills, completing a whole mission and then through staking (renting a vehicle or a tools) or selling them. However, it's up to the player to manage its resources smartly to progress in a game and be able to complete new missions.



Game Flow - Board

The UI of the game is supposed to be a one board at a time, with the possibility to switch between boards, which should show the parts/districts of the city. The initial state would be something like the map view's from the top. The mission's place can be shown via the pulsing icon/sign where the player is moved while clicking on it. Once redirected, a player does what they were supposed to do by announcing inside a text message. Each step a player takes is counted as a new skill, assuming that all requirements are fulfilled. Eg. going straight, turning left, turning right, reversing, accelerating, overtaking, packing, unpacking, opening/closing the door/roof/window, updating internal mode, customizing, pulling up, reacting to pedestrians, animals, cyclist, driving on roundabout, reacting to traffic lights, road signs.

Requirements means that eg. sensors are placed properly, there are connection's stations, charging stations to enable the smooth progressing of the vehicle. In such case, each mission is simply an ability to acquire new skills and progress with the level of autonomy. And each step completed is awarded via \$AUT tokens being added to the player's wallet. The tokens are presented as gold coins.

The player can check his wallet anytime by pushing the wallet icon available on the screen.

A player and a game communication takes place via messages, which reaches game mode as a notification pop-ups, then the player opens them (and they take the whole screen width)

Game Flow - Completing Mission

Once completed all inside steps, a player completes a mission. It's announced externally and the gameplay is cut by the cutscene in AI-developer station(=a garage). Then the player enables the self-mode of the robo-vehicle and watches how is it going. For the gameplay it means that a player tests what he taught the robo-vehicle and checks if it achieves the autonomy to do this specific sequence of steps(=repeat this mission). It's being presented as a video-recording, a player and a AI-developer watches on a screen how the robo-vehicle proceeds. In most cases it goes smoothly and is finished by a green checkmark on the screen and an award of progressing in a level of autonomy) and the ability to move forward for the player. In some cases, though, that's the place where the opponents of the player takes their places and stops the player.

It may happen that while watching a video-recording, they notice that the sequence of the steps is different, eg. there is a fallen tree on the road, or a new sign has been added, or there is a huge hole in the road, or there is a detour, or a sensor is missing, and because of that (if the robo-vehicle doesn't have a skill to do this from the previous missions) it stops and the player's fails to achieve a new level of autonomy has to teach the vehicle the skills missed. All of this may happen because of the activities of player's enemies. It's not very clear for the player from the scratch what's the origin of those problems, as they have a WTF impression, and think at the beginning that these are only the random events, but they discovered it after a while while discussing with an AI-developer. It's him who suggested it, as he also plays a narrator-role in the game. All skills acquired or missions completed are saved on the player's account, which can be transfer to other gameplays

Game Flow - Enemies, Issues, Viruses

The way of how the enemies play, means that while progressing with the skills a player become more and more immunised to them. So skills gained from previous missions eg. reacting to a fallen tree can be used while doing a self-mode, even if this element was not the initial part of the mission. (Eg. a player completes a mission without a fallen tree, then the video shows how the robo-vehicle avoids the obstacle whether) move to a new mission with the same goal.

Game Flow - Issues

Additional issues while playing are the cases where a robo-vehicle's system gets a virus or is hacked by the enemies. Getting a virus means that the help of the AI-developer is needed which costs some amount of money and stops a player for a while, while being hacked basically means that the player stops the game and needs to recover/ or in other version they could also be a game over. Hacker attacks don't happen often and there could be few countermeasures which may be taken in order to prevent it. First, is to install/reinstall a better software which, unfortunately, is very expensive so a player cannot afford to do it at the beginning, while the other mitigation is to have a team, other players included, because that means that if the player is stopped, the whole gameplay can still progress. While playing together the main goal is to achieve the level of autonomy as a team, as it means the maturity of the software system which can be easily deployed to any robo-vehicle, not only to one.

Game Flow - Friends

Engaging other players to your game also means that you are able to move faster and share your skills with others. The number of players could be limited to eg. 6 but so not to mess the board it would mean that each user would have the playing self-mode in different places all across the city. So once completed successfully the places would be restricted to other players, so not to bore the players, but it would also mean that the skills acquired are distributed to others. In this meaning the shared game could end-up quicker.

Game Flow - Limitations

Two limitations which may make the flow of the game more addictive are the time limitations and the damage/the wear of the tools/elements. To get them back a player needs to create them in the marketplace and pay for it. It includes the whole equipment of the robo-vehicle.

Examples

Tasks

- pick up a child to a school
- deliver a coffee
- deliver groceries
- deliver ice-creams
- pick up a family on a trip
- pick up a passenger to the train/airplane
- letter/package delivery
- top secret letter/package delivery
- deliver medicaments
- take an old lady to a pottery class
- deliver organs to the hospital
- deliver a blood

Skills

- red lights
- broken pavement
- heavy rain or other weather condition
- road accident
- road works
- closed road (different cases)
- dead road
- no parking space
- bus-pas/taxi-pas
- demonstration on the road
- terrorist attack
- bomb explosion
- common panic
- speed not enough
- broken wheel
- broken engine

AI challenges

- huge traffic jams
- night
- speed of other drivers
- huge fog, bad weather
- detour
- broken infrastructure
- electricity
- mismatched road signs
- animal on the road
- child on the road
- roller skater/cyclist/scooter on the road
- hackers' attack - too fast/too slow, wrong direction
- error in code (unforeseen circumstances)
- road obstacles (pits, crabs)
- car-parking in a shopping center

A map is available to player (a city is there divided into districts with different missions)

Logical riddles range - easy, moderate, difficult

man - vehicle communication via voice / buttons and info-entertainment

Game's core functions

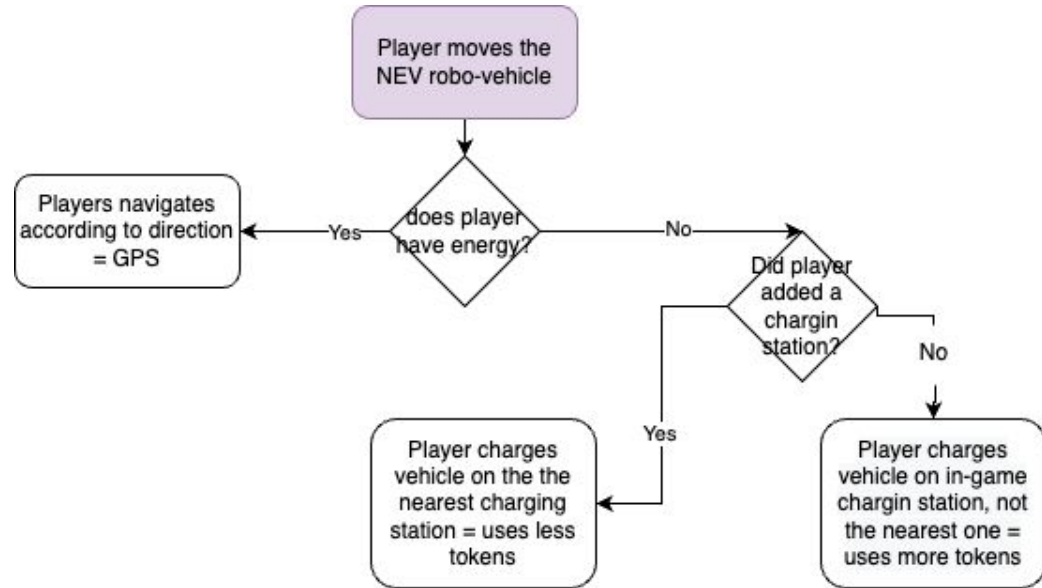
[Core] - Driving by car and completing a mission (=learning a specific task)

The game consists of missions (eg. section of the road without traffic lights and without pedestrian crossing so to deliver a small cargo to the post office) and skills like pulling off at the traffic lights, driving by the roundabout, stopping at the side of the road

The missions have different levels of complexity depending on the road conditions and task's complexity. As the the main goal is to make the NEV-vehicle fully autonomous which will be able to complete daily tasks on its own, the player needs to teach it all the roads and all configurations in the city and how to customise itself to do the specific tasks and (so how to make immuned to the obstacles which it can experience)

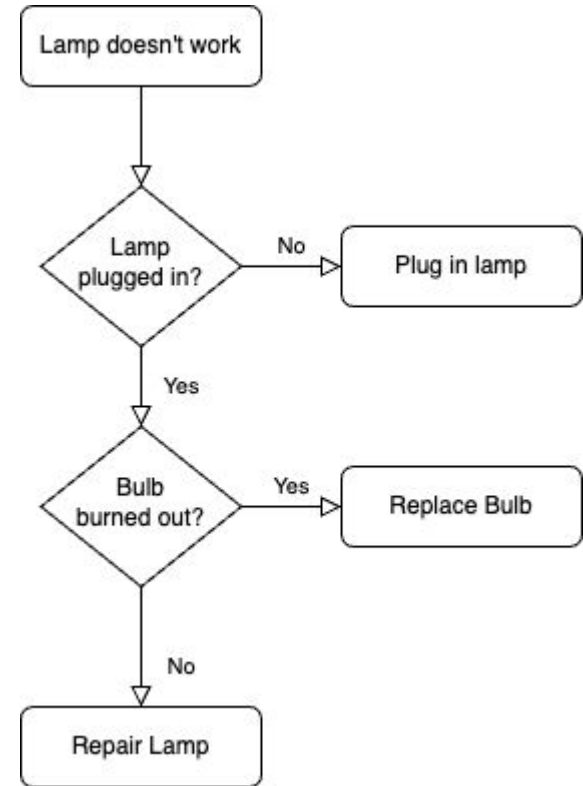
So Eg. at the beginning the NEV-vehicle requires player's control, so the player is within the vehicle, and once a section of the road is done successfully it means that the NEV robo-vehicle gains a bunch of new skills and is able to repeat the road A to B all by himself, and progresses on the road to be fully autonomous.

Each skills mentioned at the beginning is recorded at the player's wallet, and those are the skills they can also share with other players (awarded with tokens)



The missions completion moves the player forward on its way to achieve the full autonomy.

Eg. section of the road without traffic lights and without pedestrian crossing and the mission is to take a small cargo to the post office. So the player does it with their car for the first time, and in case of success there is a video-recording displayed to the player which presents self-driving car doing this task fully independent, but it still won't be able to complete the other ones until it has been taught to do so.



[Core] - Vehicle's customization

Each mission is announced/showed with the prerequisite to complete it. It basically means that the player needs to have the sufficient funds to buy the special tools (=learn the vehicle to customize itself) There is no strict order of the missions, but the only dependence here is that the more complicated it is, the more funds and settings it requires.

Customise your own or others players vehicles, play with colors, artists designs, exterior or interior look, or software upgrades.

Exterior customization	Interior customization	Embeded systems	Drivers	Additional Features
<ul style="list-style-type: none"> - Grille - the better grille the better cooling for the batteries and cpu's - A pillar - Roof - assemble the railings to extend the functionality for example for additional trunk space - Roof line - Possible to install additional lights - C pillar - possible to assemble additional transporting items - WheelHub cover - Fender - covers all suspension components - Front wind shield - to generate more energy from solar panels which are installed in the wind, protection features against rays, opacity to change interior visibility - solar panels - connected to the game's enhanced mechanics as it enables to move without the need to stop for charging - Hood - better access to the engine, allows to add additional cooling - Bumper - assembly of additional features, like a bulldozer for removing the snow from the road or helping in the road work, which helps to move other cars when the car is stuck into the ground - Door panel - different types of doors(wide, sliding), needed for a specific mission - Side window 	<ul style="list-style-type: none"> - interactive mood lights - - dashboard - HUD - displays differents information about the current situation on the road, AI control percentage, current speed, engine params - game consoles - internal equipment - umbrella, table, heating - temperature - can be adjusted to specific mission - internal design - comfortable chairs - inner space - can be modified to transport people, large-sized furniture, liquids, packages, som times specific items needs special environment 	<ul style="list-style-type: none"> - heating system - crucial to maintain the level of temperature inside the vehicle - gps system - needed to shows the path to the point - Road trackers - collecting data for ai learning - ai data collector - collecting events which happens around the car - object detector - detecting items which are on the road, vehicle can break if it is needed or accelerate or turn - collision system - emergency breaking - driving system - - engine sensors - sensors for measure endurance items 	<ul style="list-style-type: none"> - New software - Software upgrades - increase item endurance 	<ul style="list-style-type: none"> - Different batteries capacity - Fridge - winch - hook - bulldozer - scrubber - energy generator - apply the energy to different objects - solar panels - Faraday cage - outer user doesn't have an connection to the items which are inside the vehicle - safe encrypted WiFi - antivirus - better protection level

Player roles

Items Inventor

Be a creator, invent items fulfilling NEV vehicle standard, produce unique items or for wider audience. Feel like a producer who providing good quality products to other game players. The player can use external tool for modifying default items, change shapes of the item, modify core functionalities of the item.

Items Designer

Be a designer, play with colors, graphics, apply your ideas to be significant, recognizable personal label, get more tokens, by creating designs to the society.

Achieving a level of autonomy for mission's completion or for acquiring a bunch of new skills.

For each mission's completion, and a bunch of skills gained inside, a player receives a new level of autonomy. There should be predefined numbers of levels to the full autonomy eg. 7, so to be clear for the player how many of them are needed to be achieved. Also it's worth mentioning that to on the overall game's statistics both level of autonomy and the tokens earned are being taken into consideration so not to discourage players from sharing their skills, but to show that they also need tokens, so need to cooperate with others (as they earn tokens through such actions)

Sharing

- **your skills with other players**

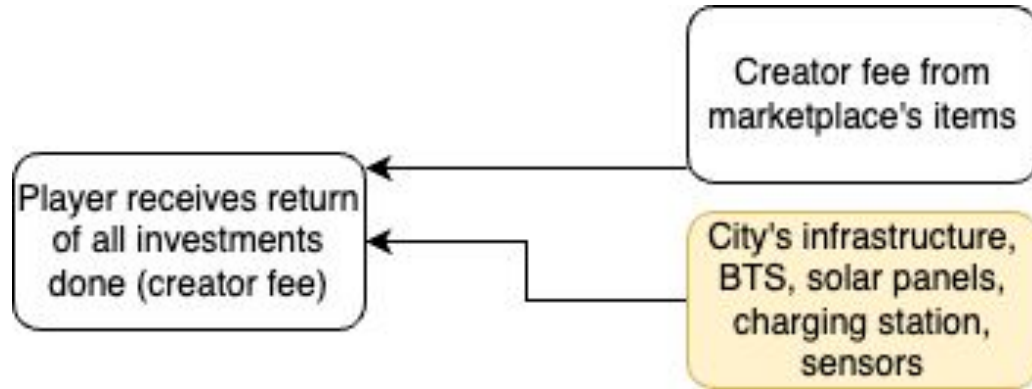
While driving through the city on the board a player gains skills which are required to progress on the level of autonomy. A skill is every single thing which a NEV does, like starting, stopping, packing things on the board, driving one side road, driving on a road with heavy traffic

- **your ownership of the robo-vehicle, its elements and the design**

Selling the things you created and/or don't need anymore, or renting them for a predefined period of time to help them while playing a game. The idea is that renting or buying the things from other players will be cheaper than buying the things offered by default on the marketplace. Also renting enables progressing when you run out of the money.

Idle

The player gets a fee in case if he/she built stations or solar panels or other infrastructure items while being away from keyboard



Building infrastructure

The player to earn more tokens and gain the experience in a unique skills, can build the infrastructure on the game map by installing objects, which are responsible by providing energy, gathering data. As a user, you can assemble the sensors to the traffic lights, crossing, and vehicles' exterior or interior. The more sensors you have the more reliable you AI model is. Game player can also add more solar panels to produce more energy for the land, the solar panels can be bought in the special energy farms. Besides of that you can direct the energy to the energy stations which can be placed close to the important places in the map. The objects could be upgraded to be more efficient in case of new driver release. AI land is becoming self-sufficient, and tokens revenue gaining by building the infrastructure.

Building infrastructure items

- sensors
- bts stations
- solar panels
- charging stations

Alternate Mechanics

Solving riddles as a phone call to a friend (an AI-developer)

A player can teach the robo-vehicle a new skill by only implementing a proper software which is done by solving different-levels puzzles which may be something like sudoku, maze, or similar.

Enhanced Mechanics

- **Driving with a higher speed**

Initially robo-vehicle is not very fast

- **Driving for a longer time**

At the beginning the time to do a task without the need to load the energy is not very long.

- **Independence on the current charging stations**

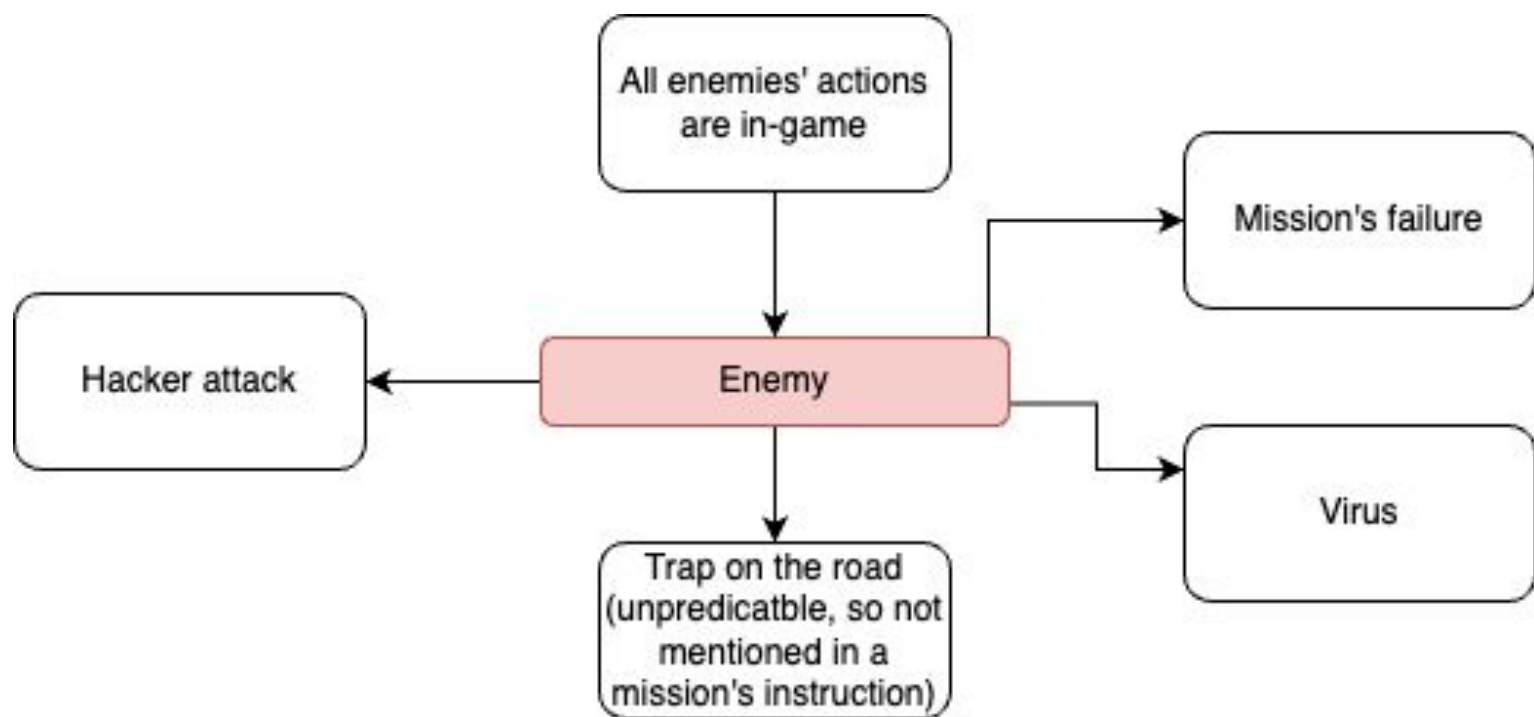
- **Independence of the external Internet resources**

You have own source of the internet, it is your own safe, protected infrastructure, you can provide it to the other users

- **Higher protection on the hacker attacks and viruses**

the latests software upgrades, antiviruses,

Opposition Mechanics



Random obstacles on the road

When a player teaches a robo-vehicle a new skill (the section of the road with a specific mission) its (a robo-vehicle) autonomy progresses. So, a **mission is completed while two conditions are met together**

- 1) the robo-vehicle did a mission with a player,
- 2) it repeated it on its own, while the repetition is presented as a cutscene, a quick video-recording to the player, which shows the whole mission, but with an extra thing which wasn't there while teaching (eg. a fallen tree on the road, so if the robo-vehicle doesn't know how to overcome it it stops and the player needs to return to game to gain this skill, unless they could get and apply this skill from another player (awarded with a few tokens).

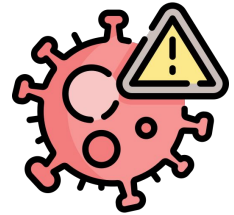
Random obstacles on the road

It means that the progress is stopped (a player cannot achieve a new level of autonomy, and it can also mean that they lose the time)

But it also means that the more supporters you have the quicker and easier you can compete the missions (because by loaning/sharing you can quicker gain the missing skill)

A Virus

A virus it means that the software of the robo-vehicle has been hacked and to retrieve it they need to pay the AI-developer to fix it, using their tokens, so it also slows them down, because tokens are needed to buy different things on the marketplace.



A hacker attack

It happens depending on the level's of user's protection layer (the higher level of protection there is a smaller likelihood of this to happen), but it stops a player for a while (eg. 30min) The only way to proceed in case of this is to get a help from another player (so it's only a solution in a shared game) which can play when an attacked player is recovering.

Over taking some embedded systems

The hacker can take control over the heating system and directs to break transported products, like unfreeze fruits, heating the vaccines, melting the plastic products. Over take GPS system to direct the player to wrong location.

Social model

A game itself can be taken alone or with a game partners.

- As the main concept of the game is to play against the cars' manufacturers stakeholders and move forward with an AI and digitalisation, the player has a greater chance to win once they had a partner, as the game evolves around the idea that the group is more effective also against the opponents. than a single player. Together they are then faster to proceed, they can share resources and tools, can replace each other when the other player is under attack or out of energy.
- Also collaboration counts as a player can keep gained skills from a shared game and use them while doing/continuing a solo-game and because of that progress faster (unlock new missions, don't repeat gaining the basic skills, or have an replacement in a gameplay while recovery)

- Common game can start anytime, a player can invite others to join also in the middle of the game. The different skills, levels of autonomy, tokens possessed or elements collected don't impact the other players possibility to engage.
- Also a leaderboard can be an option to show players so they can climb it and compete with other players with their level of autonomy, the tokens collected, and items collected/produced.
- Also the ability to stream a game on socials, or invite your friends from socials to join you.
- Also related to core game mechanics called sharing it's also an additional way to help other players to progress. The second-hand items are cheaper, or you can rent the items you own.
- Also buying the items created by other players is cheaper, so it's away, especially for beginners with no progress in a gameplay.

Interaction instruction

&

core UI design

Tutorial

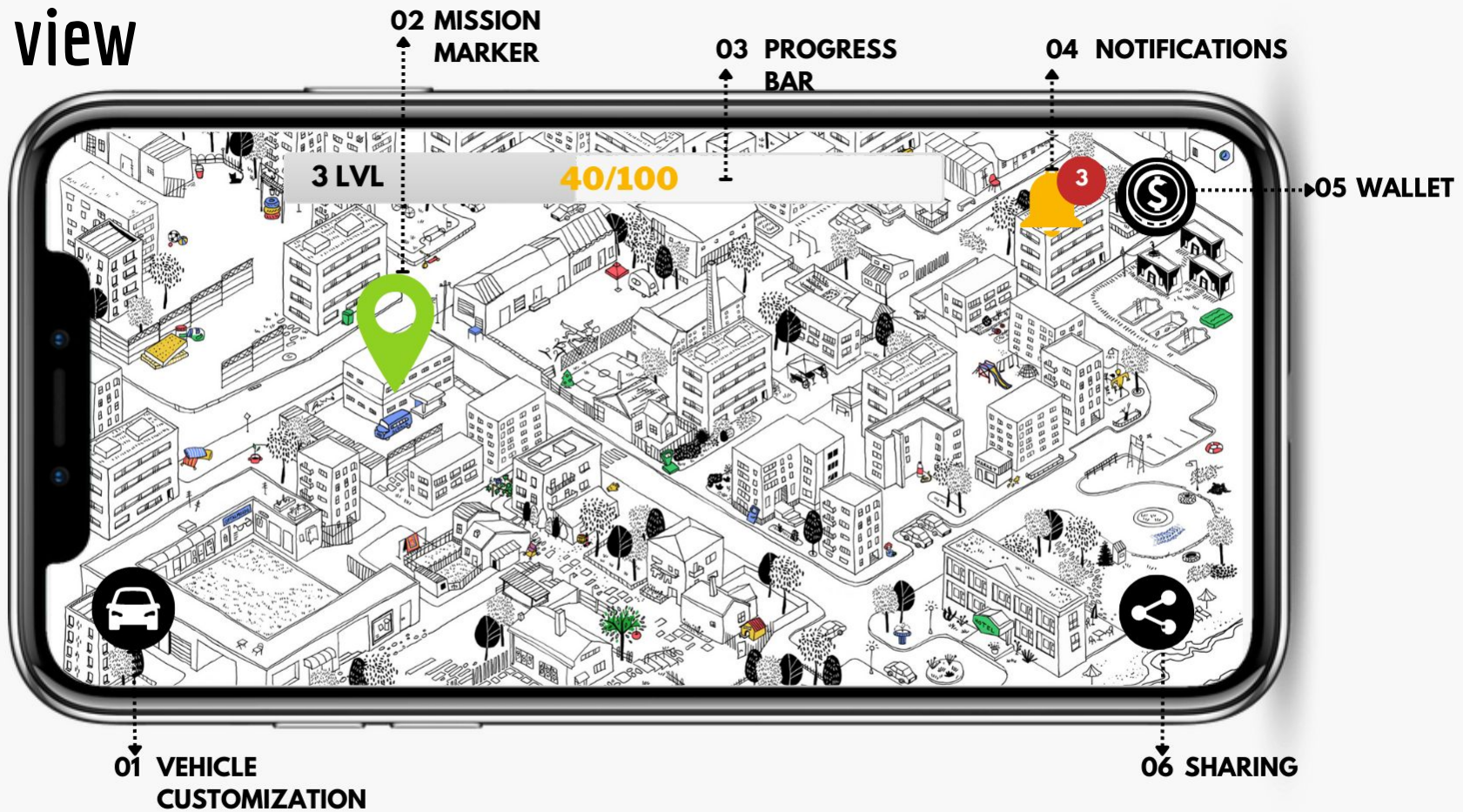
The game's tutorial should be the contextual, step-by-step lesson.

That way, the players can be instructed with pointers and text lines, which are not too long, they quickly explained what's going on. This talkative helper, AI-developer is a person, who guides the player through.

As players uncover new features, the tutorial continues. Every time a new booster, obstacle, or something else new appears on the board, the game explains them. This way, players don't need to figure out their purposes on their own, it's happening via chatting with a in-game friend, AI developer. Every time they complete a tutorial action, players earn extra resources and currency. This makes their tutorial actions feel worthwhile.

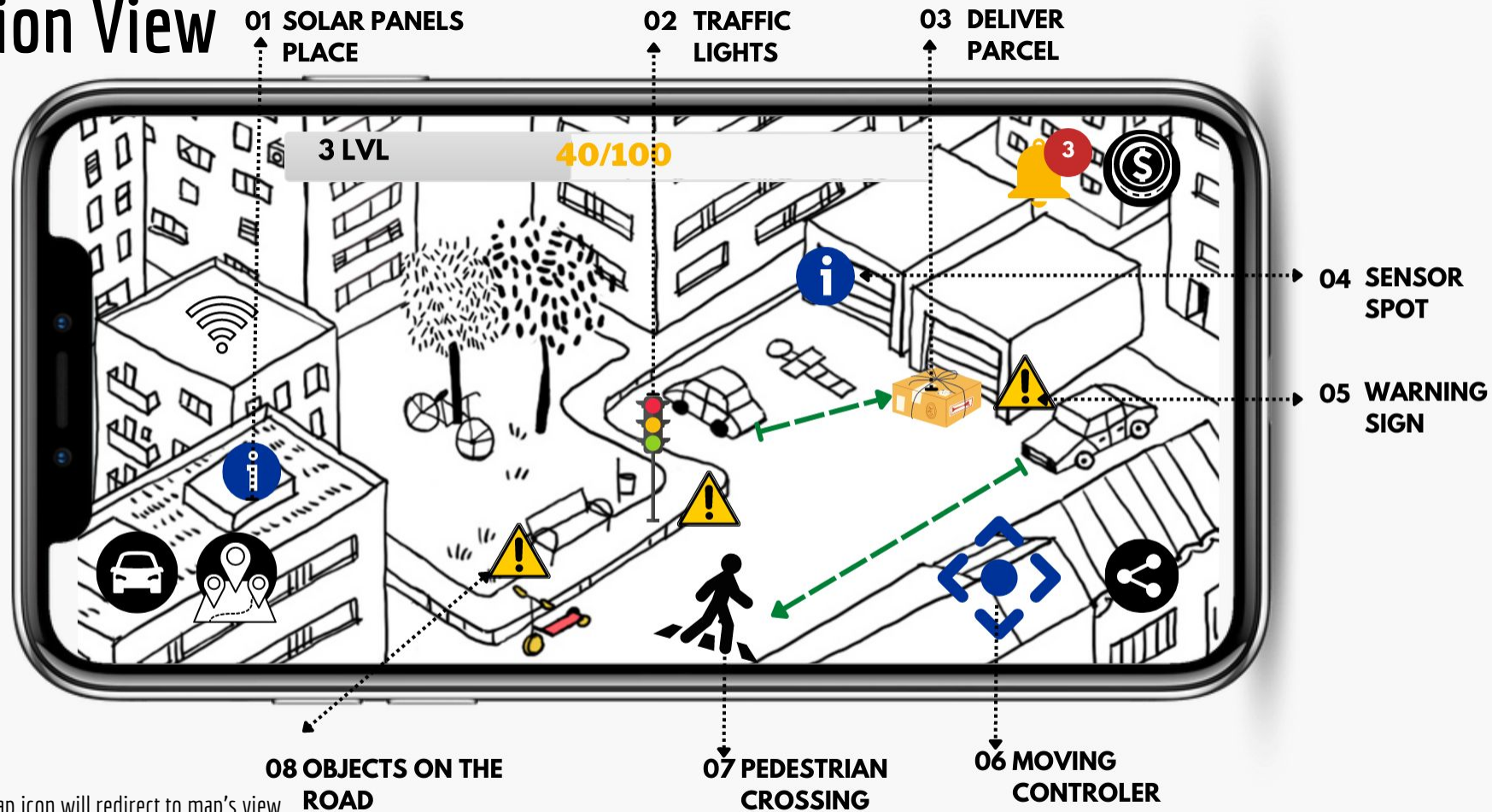
Controls

Map's view



Pressing a pin on the map's view redirects to a mission's view and enables user's to reach a detailed view

Mission View



- pressing a map icon will redirect to map's view
- pressing car button open a marketplace to choose NEV robo-vehicle items.

Resources:
Diana Stanciu

Mission View

The top right corner shows you how well the playing is doing in terms of the games' currency: &AUT token. At the top middle of the screen, you can see how many **wheels** you collected. It works as a **progress bar** to let you know how much more it takes to level up.

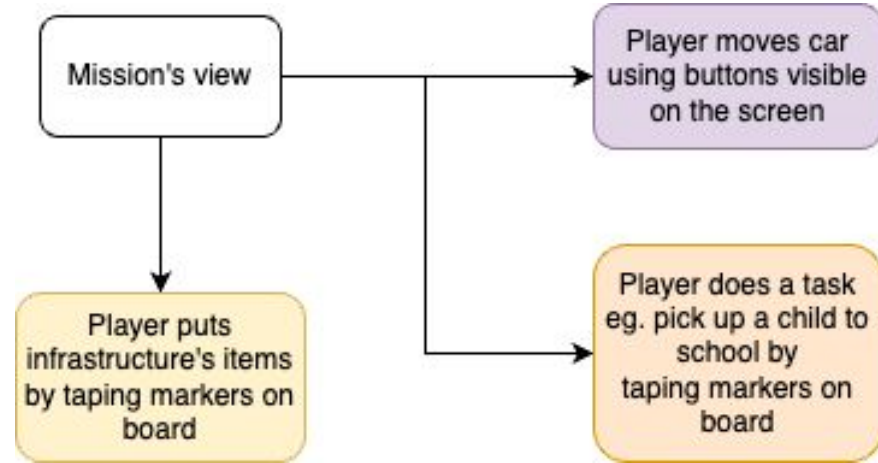
The bottom right is where you can get **social, invite other players to the game, share on social media/stream on different channels.**

In the bottom left corner, there are is a garage icon. This is where the player gets everything to progress, uses a garage to get the guidance and help, also chat with a in-game friend, and also as a configuration space with an access to marketplace.

Mission View

Controls:

- pressing left/right buttons to turn left/right,
- pressing a gas button to start the robo-vehicle
- pressing a break button to stop the robo-vehicle
- pressing a phone button to connect with a AI-developer to get directions/clues to the following
- pressing a wallet button to show account details (token, skills recording and level of autonomy)



Garage View

- Managing the vehicle components
- Customizing the vehicle by installing new components
- Changing components
- Selecting colors
- Pressing a map icon would redirect to map's view

Marketplace View

- Selling/Buying vehicle items
- Renting items from other game players

Core UI design

[UI] City's plan

The whole city will be viewed from directly above and presented like a map, then each mission would be presented there as pin of the map, and the plan would be updated automatically while player progresses with the main objective.

Then an each mission, handled in a different place in the city would be displayed as a more detailed view including all city's facilities (buildings, roads, people, bridges, traffic lights and road signs, lights, pedestrians, other vehicles, bins etc)

[UI] City's plan

The screen will scroll left/right/top/bottom as the player's vehicle, so that the view is kept approximately centered. The screen will not tilt or rotate but it would be possible to zoom in/zoom out so to do the moves which require more precision. The screen will also zoom in/zoom out automatically depending on the action, eg. it can zoom in, if the players approaches the obstacle on the road, or zoom out when they only needs to go straight.

In case of shared playing each player will have its own view (a separate place's view as described above)

[UI] Pop-ups

small popup windows showed during the game play, which would be the hints to the player's or an explanation of what happens, what to do next, etc.

[UI] - Token

coins, a skill acquired by a player or a player's return of investments would be awarded by the scattered tokens