

Public Transport Simulation Game

Software project specification

1 Introduction

This document is the specification for the Public Transport Simulation Game software project. The main goal of this project is to create a game about building and managing a public transport system in a city.

1.1 Main design pillars

The game will be a 3D, single-player only city simulation game. It will be written in the Unity game engine. It will be possible to play this game on Windows or Linux. When the player first starts the game, he will see a procedurally generated city. In the city, there will be simulated people, who will need to travel between places in the city. Initially, there will be no public transport, meaning that there will be a lot of congestion on the roads, as people will use cars as their main means of transport. The goal of the player will be to build a public transport network from scratch and thus help the city grow.

1.2 Similar existing games

In terms of the city simulation and overall feel, the game will be similar to an already existing game called Cities: Skylines. In Cities: Skylines, the main goal of the player is to build the city itself. There are some public transport elements in Cities: Skylines as well, but it is just one thing out of many for the player to focus on. The main difference between Cities: Skylines and this project is that in our project, the player only has to focus on the public transport, therefore allowing the public transport system to be much more complex.

Other similar games are Cities in Motion 1 and Cities in Motion 2. These are very close to what we are trying to accomplish, but there are some things that we want to do differently. Mainly, we want the city to feel much denser than in Cities in Motion. We want the streets to be much more congested when the player first starts the game. In Cities in Motion, the streets are almost empty. In turn, it feels useless to build a public transport network in a city where there are no problems with transportation. Another thing we want to do differently is that we want the rush hours to be much longer, and we want them to be a real problem for the player. This will provide a challenge and it will feel closer to real life.

2 Core game features

This is an overview of core game features that will be present in the game. Every feature on this list represents a core concept in the game, and there are functional dependencies between them.

2.1 City simulation

City simulation is the core of the game; the gameplay will mostly depend on the believability and realism of the city simulation. The simulation is divided into multiple parts, each serving as its own system in the game.

2.1.1 Car simulation

Car simulation is the main part of the city simulation. There are multiple parts of this system.

2.1.1.1 *Intersections*

There are two ways to divide the types of intersections we need to be able to handle: with or without traffic lights, and 3-way or 4-way.

Intersections without traffic lights will behave such that cars only slow down when coming, and they will automatically register at the intersection. The intersection handler will then instruct the cars to proceed through the intersection in a specific order so as not to crash into each other.

Intersections with traffic lights will behave similarly. The cars will register at the intersection handler, and he will let them through in an order specified by the traffic lights.

3-way and 4-way intersections will otherwise behave the same way.

2.1.1.2 *Cars pathfinding*

There needs to be a unified car pathfinding system in the city, that will be able to handle a large number of cars without slowing down the performance of the game. This pathfinding needs to be hierarchical in following layers:

- Navigation in the entire city: Returns a list of turns to make at intersections
- Navigation between intersections: Returns a list of waypoints for the car to go through. The car must not hit the car in front of itself.
- Navigation on an intersection: Also returns a list of waypoints, but needs to be synchronized with other cars. See chapter 2.1.1.1.

The car pathfinding will also depend on the current traffic situation in the city. If the main road is congested, it might be quicker to take a longer route, for example.

2.1.2 People simulation

Simulating people is another key part to the believability of our city. People will be represented by a model of a person on the street, but if the person is e.g. in a building, the model will not be rendered. Simulating people will comprise of multiple parts.

2.1.2.1 *Categories of people*

There will be three main categories of people:

- Students
- Workers
- Seniors

Every person will belong in one of these three categories. This category will influence the daily routine of the person. Students will mostly go between their home and school; workers will mostly go between their home and place of work and seniors will mostly stay at home.

There are some things that will be common for all categories, mainly going shopping and visiting points of interest.

2.1.2.2 *People pathfinding*

Pathfinding for people will be quite similar to pathfinding for cars, but there are a few key differences. Mainly, we will allow people to cross through each other. Therefore, there will be no collisions between people. People will be allowed to walk on the sidewalks and on crosswalks. We will try using Unity's NavMesh feature for this pathfinding, but if it slows down the performance too much, we will have to come up with our own solution.

2.1.2.3 *Deciding on means of transport*

The main activity of the people in the city will be going from place A to place B. This means that the people will need to decide how to get there. There will be three main means of transport for a person:

- Public transport
- A car
- Walking

There will be a complex decision system based on the time and money it costs to use each way of transport. Another factor for the public transport will be the person's satisfaction with previous public transport usage.

The main idea behind the system is that initially, as the roads are congested, using a car will be expensive both in time and in money. Therefore, many people will likely switch to the public transport which the player first built, because it will be much cheaper. But this will, in turn, cause the roads to free up again, leading to more people using cars again. Over all, this will result in an equilibrium between all the transport types.

2.1.2.4 *Daily life simulation*

In order to give people a believable reason to go from point A to point B, they need to have a simulated life. If done well, this will result in a realistic simulation of rush hours in the day, as many people will need to get to work at once.

Peoples' life will depend on multiple parameters. Mainly, it will depend on the category of the citizen. Students will go to school and back on weekdays, and on weekends, they will go shopping or they will go visit points of interest.

Working class citizens will usually go to work on weekdays, then go home. They might visit a store on their way back from work. On weekends, they will go shopping and visit points of interest. However, some workers will have to work on weekends, therefore their free days will be on different days than Saturday and Sunday.

Seniors will mostly stay at home, but they will also go shopping and visit points of interest. Because they will not go to work or school, they will comparatively do this more often than other categories.

For the player to have a realistic feeling of the city, he will have the option to click on people walking down the street or driving a car. A window will pop up showing the parameters of the person, as well as where he is currently headed.

2.1.3 Bus simulation

Simulating buses is another core part of the city simulation which, when done well, will help create a realistic feel of the public transport system.

Buses will be assigned to a line. They will go from one stop to another according to the line they are assigned to and in time intervals according to the time table of the line. They will need to be able to pick up and eject people on bus stops. When the time table changes, buses will magically appear on the line if more of them is needed, and disappear if less is needed.

Bus pathfinding will work in a similar way to cars.

2.1.4 Subway simulation

Subway simulation will be quite easy from a programmer's point of view. Different metro trains will be assigned to lines similar to buses, but the lines will be underground. Therefore, there will be no pathfinding, and calculating departure times to fulfil a time table will be much easier.

2.2 Events simulation

There will be an events simulation system in the game. In its core, it will only contain events generated by the city council, which will mostly be tasks for the player, such as building a public transport in a specific district of the city. This will come with rewards for the player. More events will be in the nice to have features.

2.3 Points of interest

All throughout the city, there will be different kinds of points of interest. These will be buildings that will not be generated as a part of a zone, but will stand alone. Points of interest will be divided into two categories:

- Schools,
- Recreational.

As students will need to go to schools, we need to generate them outside of zones generation.

Recreational points of interest will mostly be parks or different types of recreational places. They will serve as a place for people of all categories to visit once in a while, to generate random stress on the transport system.

2.4 City generation

City generation can be split up into multiple environment generation tasks. Generating a realistic city is a main goal of this core feature.

2.4.1 Terrain generation

The first step is to generate a terrain to generate the rest of the city on. The terrain will be generated using a classic Perlin noise with proper scaling and exponentiation.

The terrain should be mostly flat to leave enough space to generate the city on. Therefore, hills should be rare. In the core features, only hills generation is included. Other terrain features are in the nice to have features.

2.4.2 Roads generation

The next step is to generate a road system. This should result in a realistic grid of roads. There should be enough gaps in the grid, so as to appear organic.

With road generation comes the generation of support structures for other game systems. For example, we need to generate a set of waypoints on the roads for cars to navigate by. We also need to generate grids along the roads for building zoning.

2.4.3 Building zones generation

With grids in place next to the roads, we can assign them to specific zones. There will be three types of zones in the game:

- Residential,
- Commercial,
- Working.

The grids in entire districts of the city should be assigned to a single zone type. This will lead to specialized districts of the city. The idea is to generate large blocks of residential zones with some smaller blocks of working zones in between. Commercial zones should fill the rest. This comes from real life, where commercial zones tend to be quite small in comparison to residential areas or industrial/office areas.

2.4.4 Building generation

Now we have grids in three types of zones. The grids will be comprised of small squares. Each grid will be 4 squares deep and its length will be the same as the length of the road it is attached to.

This grid can be filled with buildings of different sizes and appearances. The goal of building generation is therefore to populate these grids with buildings depending on the type of zone of the grid and the free space remaining in the grid.

2.4.5 People generation

With the buildings generated, it is time to generate the people living in the city. Every person will have these attributes assigned to him:

- Name,
- Gender,
- Category (Student, Working or Senior),
- Place of residence,
- Place of study/work

These need to be generated in a proper way to fill the city. For example, each building will have a maximum capacity of how many people can live/work/study there. There should also be a set ratio of students, workers and seniors.

2.4.6 Continuous city generation

As the game goes on and the player is successful in building the public transport network, the city should grow and change. This will serve to create new challenges for the player, and the game cannot be ever truly finished. This continuous change of the city will have two main sources: new districts building, and replacement of buildings.

New districts will be generated to simulate area developments of the real life. Also, it will create a good reason for the player to immediately extend his transport network that way, as most of the residents of this new district will work in the old part of the city. If the city grew only slowly zone by zone, the player would not initially have enough new people living there to warrant extending the transport network.

Replacement of buildings will help the city feel alive. As the game goes on, old buildings will be demolished, and they will be replaced by newer buildings. These buildings will be of the same type, but they can have different capacities. This means that required transport capacity to already existing zones can also change over time.

2.5 Time simulation

Time simulation is another very important feature. It is what decides how accessible the game will be. In this game, the time simulation will be thirty times faster than real time, therefore, for every two minutes of real time, one hour will pass in the game. We believe this is fast enough not to be boring, but it is slow enough for realistic peak hour simulation to be possible.

The player should be also able to manipulate the speed of the simulation. He should be able to pause the game or to speed up the simulation. Pausing the game is useful for when the player needs time to build up the network suddenly. Fast forward can be useful when the player ran out of money and therefore has nothing to do but speed up time.

The game will keep track of days, weeks, months and years. These will increment the same way as they would in real life.

2.6 Public transport network construction and management

In this section, we will describe how the network building and management will work. This is the main way the player will interact with the game, and it should be as intuitive as possible.

2.6.1 Managing transport lines

The player will have the ability to manage transport lines, which are a part of the network. This includes creating new lines and modifying existing lines. The player will have the option to build and delete stops. Every line will have an associated route, which will be a circle between stops. With buses, this circle can have whatever shape the player wants on the road grid. The subway lines will be underground, and they will just go back and forth.

The player has multiple tools to manage lines. He can assign or remove buses/subway trains from lines. He can modify the timetable. The player can also modify the route of the line, adding or removing stops. Every line also has a name and a colour. The player can rename the line and change the colour.

2.6.2 Managing vehicles

The player will have the ability to purchase and sell vehicles as he sees fit. These vehicles can then be assigned to transport lines to serve them. There will be multiple types of buses, and the player will be able to choose which one he wants to use. They will differ in price and specifications.

It will also be possible to rename individual buses.

2.6.3 Building transport lines

2.6.3.1 Building bus lines

The player will have the ability to build new lines in multiple ways. The main way is to build the line with sequential clicking on stretches of the road, where new bus stops will be placed. If there is a bus stop already, it will be used, and a new one will not be created. The line building is finished when the player clicks the first stop again. The game will remember the order in which the stops were selected, and then create a line visiting all the stops in the correct order.

Another way to build a new line is to build the stops separately at first, and then connect them into a line. Therefore, the player will have the ability to build stops without any lines serving them.

The line building will be accompanied by a graphical representation. Everything in the city will become white, with only a coloured line connecting the stops to highlight the path the bus will take. The line will be of the same colour as the colour of the line.

2.6.3.2 Building subway lines

Building subway lines will be done by first building the stops, then the tunnels, and then the lines themselves. In order to build a stop, the player first selects a stretch of road where he wants the stop to be. A stairway to underground will appear. Next, the player will select a rotation of the underground platform that suits him by moving the mouse.

When the player has built a series of stops, he needs to connect them with tunnels. Each station has two endpoints, where the tunnel can be connected. In the core game features, it will only be possible to connect two endpoints together. Hence, it will not be possible to branch the tunnels in any way. This will greatly simplify line building. All that the player will have to do to create a new subway line is to click a tunnel, and the game will automatically deduce the path of the line based on the tunnel.

2.6.4 Managing lines

There will be multiple ways the player can modify transport lines. The basic ways are changing the name and the color of the line, the more advanced ways include changing the path of the line, modifying the timetable, and assigning transport vehicles to lines.

2.6.4.1 Path modification

The player will have the ability to modify the path of existing bus lines by drag-and-dropping with the mouse. It will be possible to drag an existing stop to change its position. It will also be possible to create a new stop by dragging from a path between two stops.

Subway line modification will be done automatically with the modification of the tunnel the line is in. Since it is only possible to lengthen the tunnel, modifying the line is trivial.

2.6.4.2 Setting timetables

It will be possible to set intervals for the vehicles. This denotes how often a vehicle should visit each stop. It will be possible to set different intervals for different times of day. There will be the following day times:

- 06 – 10: The morning rush hour
- 10 – 15: Midday
- 15 – 19: The afternoon rush hour
- 19 – 23: The evening
- 23 – 06: The night

The game will then automatically calculate the number of vehicles needed on the line for the interval to be fulfilled, and show it to the player. The player will then have to assign at least this number of vehicles to the line.

2.6.4.3 Assigning vehicles to lines

After the player has set a timetable, he will need to assign the appropriate number of vehicles to each line. It will be possible to do this in bulk, so as not to be too tedious for the player.

2.7 Money system

There will of course be a money system in the game. It is what motivates the player to expand his transport network, but it also serves as a limiting factor for the player, so that he does not expand the network too fast. Money also serves as a measure of success of the player. The more money the player has, the more successful he feels.

Periodic income and periodic expenses will be accounted at the end of each day. Other expenses, like purchasing a new bus, will deduct the money immediately.

2.7.1 Uses of money

The player needs money for the following purposes:

- Maintaining the transport network
- Paying salaries
- Expanding the transport network by adding new lines
- Buying new vehicles

2.7.2 Sources of money

The player can gain money from the following sources:

- Selling tickets
- Getting subsidies from the city
- Selling advertisement space in the subway and on the bus stops

2.8 Game saving

The player will have the ability to save his game at any point. The number of save slots is unlimited. The player can load any game he wants. The game will also feature an autosave system, which will save the game in periodic intervals.

2.9 Statistics

Various statistics about the city and the company will be available for the player to view. There will be two main ways in which statistics will be displayed to the player: a line graph, and an overlay over the city.

A line graph will be used for the following statistics:

- Overall city population over time
- City population of different citizen categories over time
- The amount of money available to the player over time

An overlay will be used for the following statistics in the following ways:

- Traffic: each road segment will be coloured depending on the current traffic
- Satisfaction: Each residential house will be coloured depending on the average satisfaction of its residents
- Average delays: Each stop will be coloured based on the average delay of vehicles serving the stop

More statistics might come up during development, therefore this list is not final.

3 Graphical User Interface

The main goal of the Graphical User Interface (GUI) is of course to present information to the player about the state of the game, as well as to allow the player to interact with the game. The GUI should be easily readable, intuitive, and it should minimize the number of button presses needed to accomplish required tasks. The GUI will again be comprised of multiple parts.

3.1 The main menu

The main menu will offer standard options for the player. It will allow the player to enter the main game, as well as modify the game settings. On Figure 1, you can see the progression of the main menu. The player will have the option to save and load as many games as he wants. The graphical representation of the main menu is not yet certain.

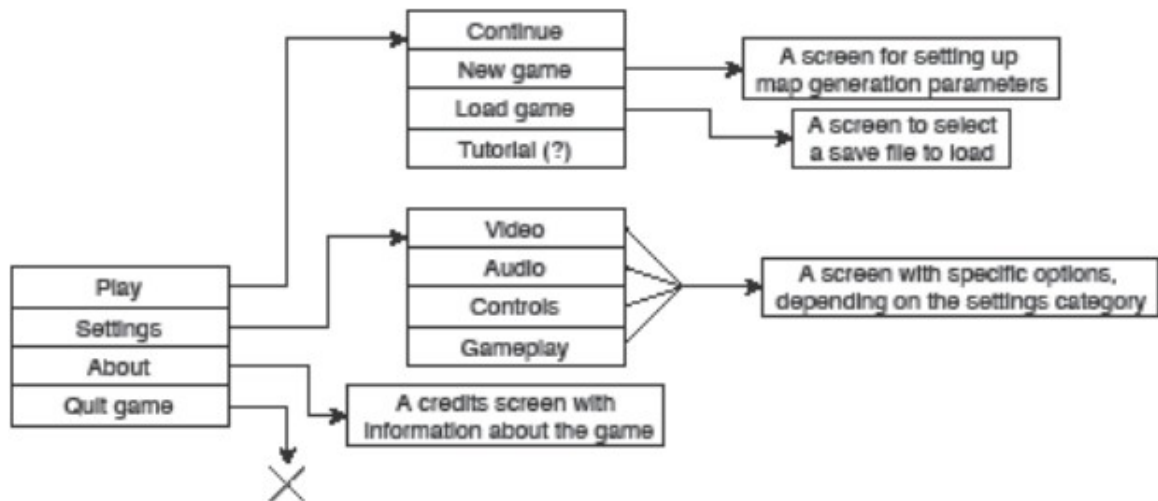


Figure 1: The main menu progression

3.2 The in-game GUI

After the player starts a new game, the in-game GUI is what the player sees the entire time. It displays information about the game state, and it allows the player to modify the game state.

3.2.1 The camera

The player will have the option to move the camera in the game. The camera will be moved in a standard way with the WASD keys. This will move the camera forward, left, backward or right respectively on a horizontal plane. In order to move the camera up and down, the player will have to use the E and Q keys respectively. It will be possible to use the mouse wheel to move the camera forward or backward in the direction it is facing. When the player holds down the Shift key, the speed of camera movements will increase.

There will be a maximum upper limit on how far the camera can go, which will allow the player to overlook the entire city. It will be possible to zoom in all the way down to the city level.

3.2.2 The heads-up display

The heads-up display (HUD) is what shows the player most of the information. It also contains many ways of interacting with the game. On Figure 2, you can see the first prototype of the HUD. The top panel contains the following elements, from left to right:

- A display of current time
- Four buttons to modify time simulation speed (pause, 1x speed, 2x speed, 4x speed)
- A display of current money supply and flow
- A slider showing the current overall traffic in the city
- A slider showing the current overall passenger satisfaction

In the top left corner, there are six placeholder buttons, which, when clicked, will display various statistics for the player, either as overlays over the city, or as individual windows with graphs. It will be possible to hide these buttons.

In the bottom left corner, there is a large placeholder panel for notifications. This is where important notifications will be shown. One notification will have the form of a shallow rectangle. These rectangles will stack on top of each other, where the newest notifications come in from the bottom, and the oldest notifications slowly disappear on the top. Upon clicking the notification, the camera will shift towards the area where the notification happened, so that the player can better see what is going on.

On the bottom, there are two panels on top of each other. The smaller (bottom) panel contains buttons with categories like “bus”, “subway” or “stops”. The bigger (upper) panel will contain specific buttons for the current category. For example, for the “bus” category, the buttons could be “Build new line”, “Manage buses”, “Manage lines”. However, the content of these buttons will change during the development of the game. The player will have the ability to hide the bigger panel, if he does not currently need it, by clicking again on the button of the category that is currently open.

In the bottom right corner, there are two placeholders for a compass and a bulldozer. The compass will show the player the current orientation of the camera. The bulldozer button will allow the player to destroy already built stops or subway lines.

We decided not to include a minimap. The main reason for this is that if the player needs to see the entire city, he can just zoom out with the camera and get a better look himself over the whole city. Then he can zoom in where he needs to. For big cities, this will be much better than trying to click on the right spot on the minimap. This is also the way it is done in *Cities: Skylines*.

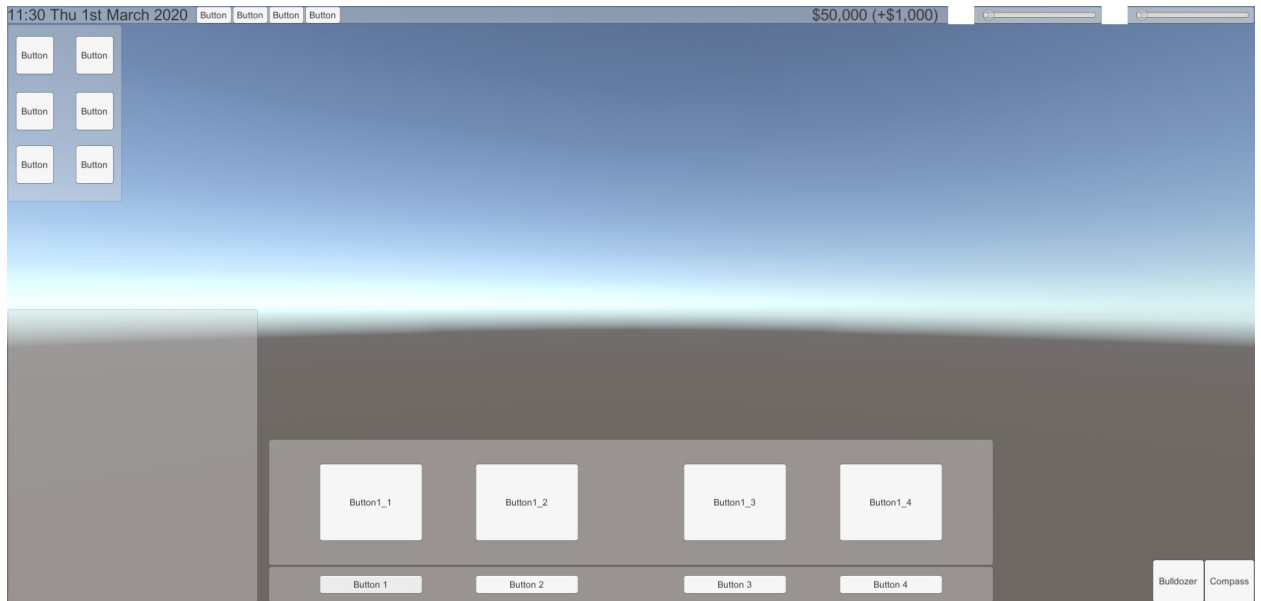


Figure 2: A prototype of the in-game HUD

3.2.3 The lines management screen

The player will need the option to modify transport lines when he needs to. This will be done in a separate window, which will open when the player clicks on a specific button in the bottom panel of the HUD. There will be separate windows for managing bus and subway lines.

In the upper part of the window, there will be buttons to add or remove lines. The rest of the window will serve to display a list of lines. Each item in the list will display the following:

- The name of the line,
- The colour of the line,
- The number of currently assigned buses/subway cars,
- The calculated minimum required number of assigned buses, in order to fulfil the timetable,
- A button to modify the path of the line,
- A button to change the timetable of the line.

The player will have the option of modifying the name of the line or its colour by clicking these elements of the line.

When the player clicks the button to modify the path of the line, the window will disappear, and the player will have the ability to drag the path of the line in order to change it.

When the player clicks the button to change the timetable, a simple window will appear, where the player will have the option to select intervals for different time periods of the day.

3.2.4 The vehicles management screen

The player will also need the ability to manage his vehicles. This will also be done in a separate window, and there will again be different windows for buses and subway cars. The top of the window will contain the following buttons:

- A button to purchase new vehicles,
- A button to sell existing vehicles,
- A button to assign vehicles to transport lines.

The rest of the window will again display a list of vehicles. For each vehicle, only its name and the line its assigned to will be displayed. The name of the vehicle will be changeable by clicking on it.

It will be possible to select multiple vehicles at once. This will allow the player to sell vehicles in bulk, as well as to assign multiple vehicles to a line at once.

When the player clicks the button to purchase existing vehicles, a new window will appear, where the player will be able to choose what type of vehicle he wants, as well as how many of them he wants to buy.

3.2.5 The information panel

When the player clicks on an object in the game, like a person or a bus, the compass and the bulldozer icons in the bottom right corner of the HUD will disappear, and an information panel will appear. This will contain select information about the clicked element. For example, when the player clicks on a person, a panel will appear showing the person's name, its place of residence, its place of work and where the person is currently going.

4 Nice-to-have game features

In this chapter, we will present a list of features that we would like to have in the game, but we are afraid they might be too difficult and/or time consuming to add. Therefore, these features might not be present in the finished game, but if we find the time at the end, we will add some of them.

4.1 Trams

As a third means of transport option, trams would be a great addition to buses and subway. However, it produces its own set of challenges, and we feel that they are not different enough from buses to be included in the core game features.

4.2 Sounds and music

Sounds and music would be really great to have in our game. However, none of us is skilled in music or sound making, therefore we added them into the nice-to-have section. If there is enough time left at the end of the development, sounds and music will be top priority.

The sounds we would like to have are mostly ambient sounds such as the sounds of cars, buses and metro, as well as humming of air conditioning units on buildings. The level at which these sounds would be heard would depend on the position of the camera. For example, if the camera is close to a car, the car's engine could be heard very loudly. There could also be sounds for pressing down buttons.

As for the music, it would probably be very simple. Its main purpose would be to serve as a background to the game. It should not distract in any way from the gameplay.

4.3 Different people behaviours

It would be great to have different behaviour schemes for different people. For example, some people might not take the optimal route to the desired location. Some people might be more likely to damage a bus stop, or they might go shopping only once a month. In the end, this mechanic might not be very visible to the player however, and it would hardly enrich the feeling of the city, as the characteristic of one person might get lost in the thousands of other people.

4.4 Additional events

It would be great to have more events happening in the city. These would give the player an enriched feeling of the city, and they might give him additional problems to solve. For example, a road might be closed for repairs. All bus lines that used that stretch of road will be automatically repathed to connect the desired bus stops, but this might not be optimal for the player. Therefore, the player might need to cancel some bus stops, or reroute the line manually. There might also be car crashes. These would work about the same way as road closures, but they would be much shorter in duration.

Another example of events is vandalism. People might break the player's bus stops and/or buses, especially when they are angry with his service. This would increase maintenance

costs for the player, and it would serve as a good reason to consider making the services better in the area.

In the subway, people might also commit suicides. This would cause temporary closures of the subway line, and the player might have to create a temporary replacement bus service, for example. However, this might prove to require too much micromanagement from the player in the long run.

4.5 Additional city features

It would be great to have additional terrain and city features to improve the overall look of the city. Considering the terrain features, it would be great to have rivers and lakes. However, it is not easy to generate these procedurally. Also, it would require bridges to be generated over them.

As for the city features, we would like to move away from the grid system. With the grid system from the core game features, we can just copy a number of road models to create the road grid. Moving away from that would require us to generate meshes in Unity from code. This could be very difficult; however, it would allow us a much greater flexibility in road generation. For example, it would allow us to generate curvy or even wavy stretches of road.

4.6 Graphical day and night representation

As much as we would like to represent night graphically, it would require us to do a lot of menial work such as adding light maps to windows, adding street lamps, etc. However, if this is done well, the result can look good. Cities: Skylines is a great example of this.

4.7 Additional bus line path building options

It would be great to add some more options for the player when he is creating a path for a bus line. For example, we would like to add a mode of line pathing for lines that only go back and forth on the same path. This mode would allow the player to build stops in both ways at the same time.

Also, we would like to add a waypoint system. Waypoints would simply be places that the bus must pass through, but they are not a bus stop. Without this, there would be no way for the player to force a bus to go the way he wants without adding a bus stop.

4.8 Tutorial

Adding an interactive tutorial for new players to learn how to play the game would be beneficial, but also very time consuming.

4.9 Research system

A research system that would allow the player to research new vehicles and improvements of current vehicles would be nice, but it would take a lot of effort and it is not central to the idea of a public transport company.

4.10 Weather and seasons simulation

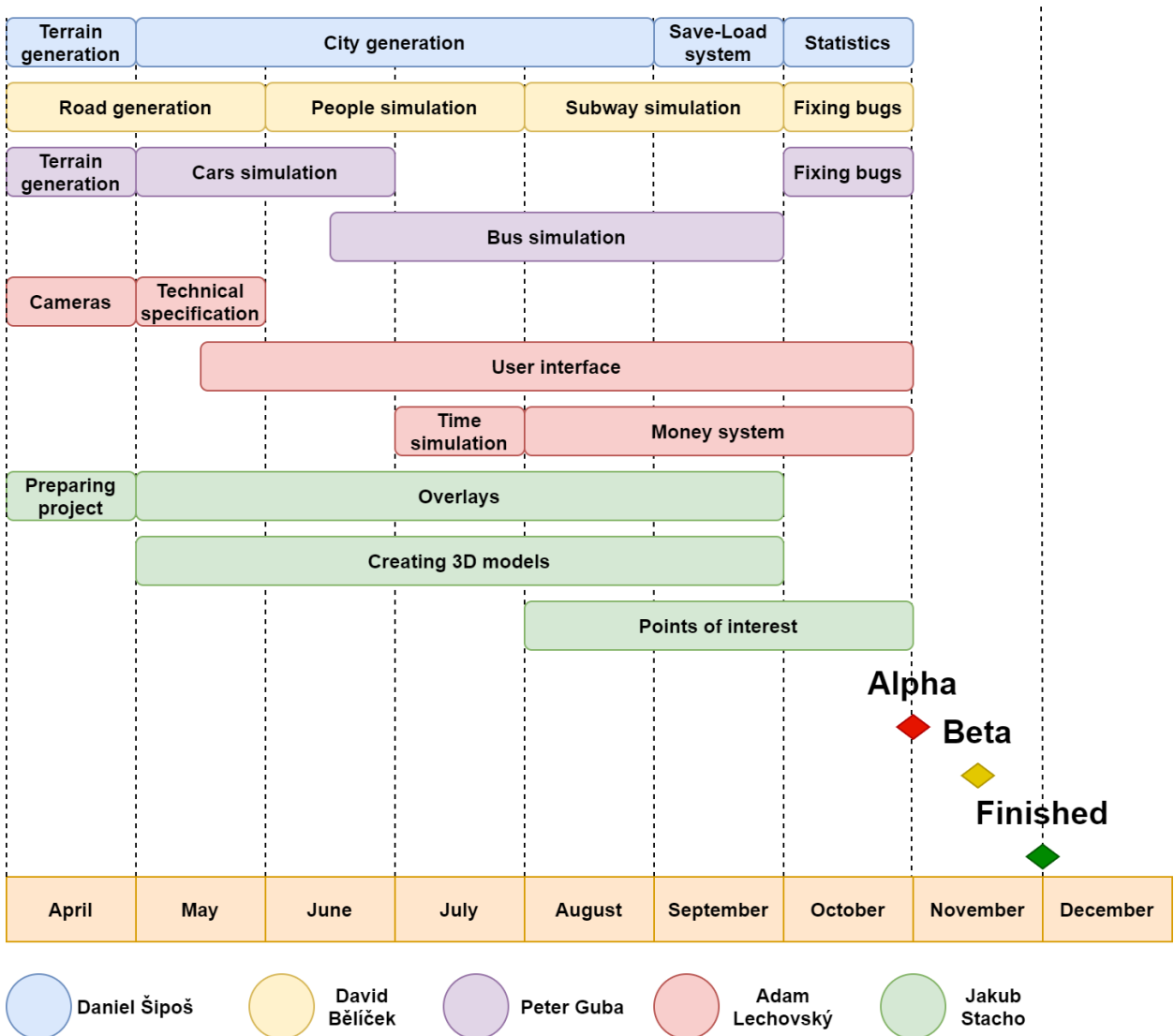
It would add a lot to the feel of the city if there was a simulation of weather. There could be rain, thunderstorms, snowstorms, etc. This would also depend on the time of the year; for example, it would snow in winter and it would be sunny in summer.

5 Project management

This team consists of five members:

- Daniel Šipoš (programmer)
- David Bělíček (programmer)
- Peter Guba (programmer)
- Adam Lechovský (game designer)
- Jakub Stacho (team leader)

All work is divided between team members based on their skills and preferences. The chart below represents a working plan with couple of milestones.



Works on the project are divided into two weeks long sprints, using an agile development method. At the end of October we should reach the Alpha state of our game and start testing our game. In the half of November, we should get our release to the Beta state, meaning that

there will be no new features added. Withing two additional weeks we should be able to make final release. If some phase of the development goes wrong, we still have a couple more weeks as a reserve.

5.1 Used technologies

The target platform, the game is developed for, are Linux and Windows. We will be using various software and services for designing, developing and synchronization of our project.

5.1.1 Unity3D

We chose Unity3D over other game engines because of the skills our project members have with this engine. Not all project members have worked with Unity, but Unity provides a huge amount of great learning materials and it's beginner-friendly. Also, Unity has a broad community which means that most of the troubles, newcomers run into, are already solved.

5.2 Blender 2.8

Blender is favorite software for modeling and making animations among indie developers. It's because Blender is free, very powerful and exportation of models and animations into Unity is fast. From release version 2.8, Blender becomes more beginner-friendly and very easy to use.

5.3 GitLab and Google Drive

We have a repository created on GitLab with GitLFS setup, in order to synchronize progress on the project. All large files such as models and pictures are tracked by GitLFS, to not making our repository big. Documentation, model details, game design documents are stored in the Google Drive folder, which is shared between all team members.

5.4 Slack

For communication, we are using Slack, which is commonly used in the industry. We can have a number of different channels inside of Slack and so divided communication into more separate chats. Slack also offers mobile app and integration with number of software like Trello, Google Drive, or Jira.

5.5 Trello

For task management we are using Trello. Trello is software where we can create tasks, assign them to different people, or divide into different groups. It's great to keep tracking the state of a task. Trello is free and you can create teams up to 5 members.