**Lesson 4**

* [Neural Networks, 1 lesson](https://kiber.one/mycabinet/get_file/496/)
* [Neural Networks, 2 lesson](https://kiber.one/mycabinet/get_file/497/)
* [Neural Networks, 3 lesson](https://kiber.one/mycabinet/get_file/498/)
* [Neural Networks, 4 lesson](https://kiber.one/mycabinet/get_file/499/)
* [Neural Networks, 5 lesson](https://kiber.one/mycabinet/get_file/500/)

|  |  |
| --- | --- |
| Course | Neural Networks. Lesson 4 |
| Lesson topic | ●      familiarization with flowcharts;  ●      studying forms and types of models;  ●      mastering algorithms of models. |
| Lesson objective | ●      familiarization with the program and basic concepts;  ●      training to use the program blocks |
| Motivation | Creating your own scripted content is exciting and challenging. |
| Tools | Automated training places (computer class), projector, Kandinsky neural network |
| Goals | 1.     create conditions for practicing the skills and abilities acquired during the lesson;  2.     promote the formation of independent cognitive activity;  3.     create conditions for the development of such analytical abilities of students as the ability to analyze, compare, generalize cognitive objects, draw conclusions |
| Expected result | ●      gaining skills in working with blocks;  ●      improving students' analytical skills |
| Recommendations for a tutor | ●      study the material in advance;  ●      be able to easily explain the meaning of each block |

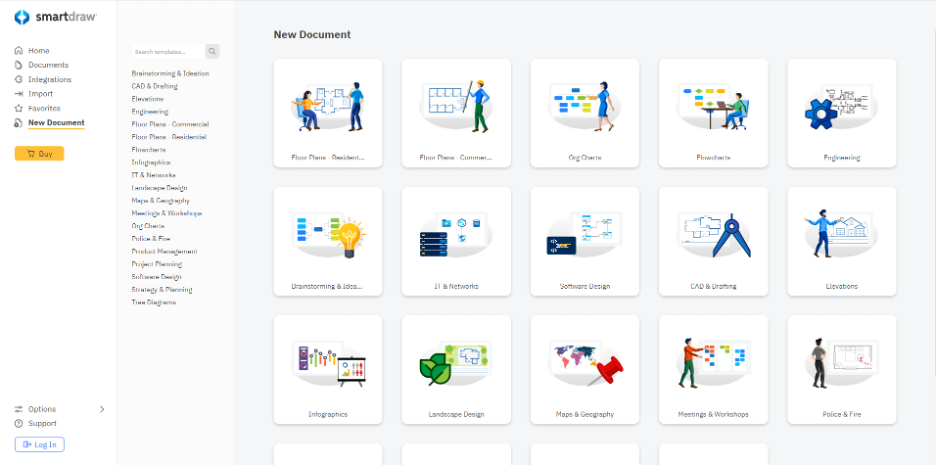
Hello, guys! Today we are going to learn how to work with flowcharts.

A flowchart is a graphical way of representing an algorithm or program. It helps to visualize the process of performing a task and make it more understandable.

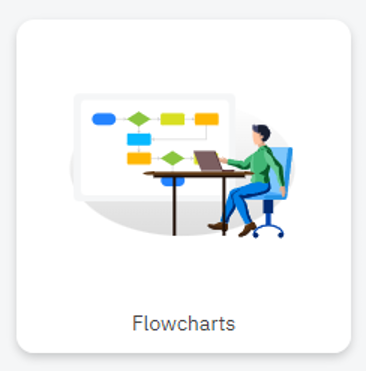
You will ask me: "Why do we need to learn to work with flowcharts?"

And I will answer you - to be able to develop and plan your own project! After all, in the next lesson we'll create our own game, and we will need this skill. We will use neural networks to design the game.

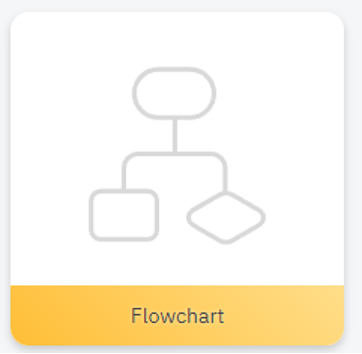
To get started, we start a browser and open this website: [smartdraw.com](https://app.smartdraw.com/?nsu=1&_gl=1*bxkatj*_ga*MzA0OTE0Nzg1LjE3MTY1MzI5NTI.*_ga_9796Z3R0VG*MTcxNjUzMjk1MS4xLjEuMTcxNjUzNjgwNy42MC4wLjA.*_gcl_au*MTIxNTQyNjEyNy4xNzE2NTMyOTUy).

[](https://kiber.one/#5b10929c-5bd6-4282-b24c-d06eb20d9a86)

Select the "Flowcharts" block:

[](https://kiber.one/#e91aa80b-edc6-4d9b-b274-00e3cbabc480)

Select the "Flowchart" block again:

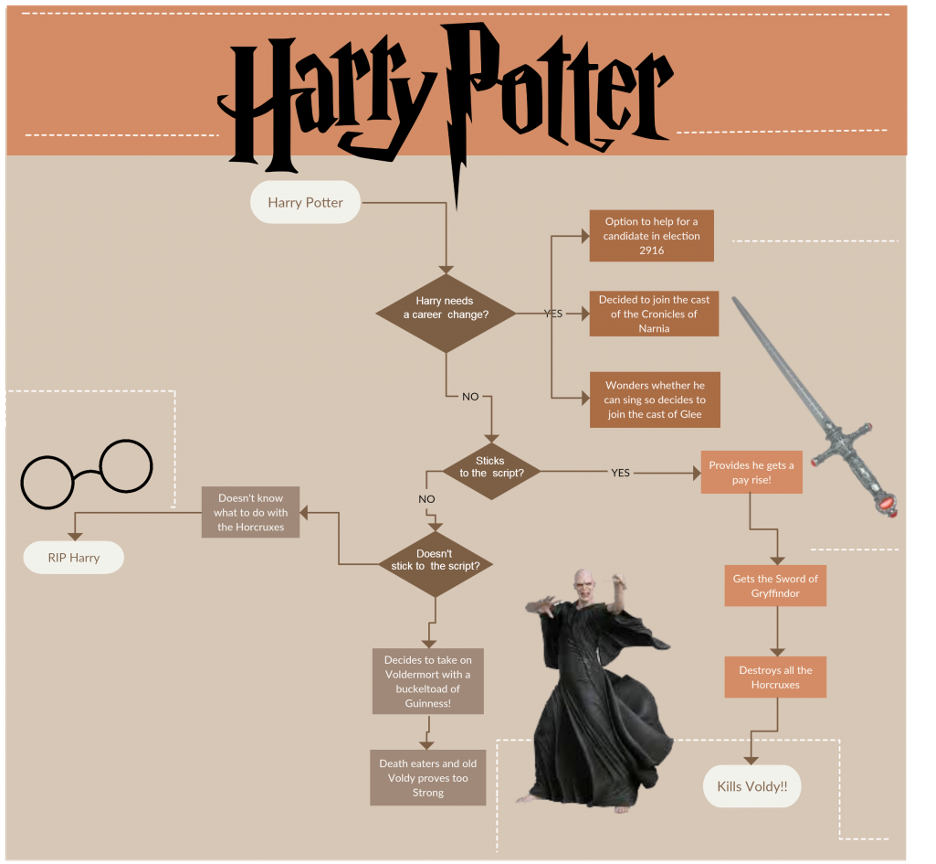
[](https://kiber.one/#9faf19cc-3460-4989-a4d4-6596d0758a45)

Before we begin, I would like to ask you a few questions. What kind of tasks would you like to solve using flowcharts? What ideas do you already have for creating your own flowchart?

*Give the children time to think and discuss.*

Great! Now that we have some ideas, we can move on to creating our first flowchart.

Let's start by developing a game plan. We will use a flowchart to do this. The game plan should have many twists and ends to make the game more interesting:

[](https://kiber.one/#e489dfb3-b58c-4231-85c3-72ce44bf6c3b)

Our game will be an adventure game in which the player must go through several challenges to reach the goal. The goal of the game can be anything. It could be finding a treasure, rescuing a princess, or accomplishing another task.

To create such a game, we can use several types of flowchart blocks.

The first type of flowcharts are blocks that help define the order in which actions are performed. For example, the block "start" (an oval) - shows where the game starts, and the block "end" (also an oval) - shows where the game ends:

Another type of flowcharts are conditional blocks, which allow you to determine which actions should be performed depending on certain conditions. For example, conditional block "if-then" (rhombus) - shows that a certain action should be performed only if a certain condition is true:

The third type of flowcharts are blocks that allow repeating actions several times. For example, the block "loop" (parallelogram) - shows that it is necessary to perform certain actions several times until a certain condition is met:

For loop

We can use all of these types of flowchart blocks to create our game. For example, we can use a "start" block to show where the game starts, then add a conditional "if-then" block to show that the player should only perform a certain action if a certain condition is true. We can then add a "loop" block to show that the player must repeat certain actions multiple times until a certain condition is met.

In order to create an interesting game, we can also use branching and endings. For example, we can add several different paths that the player can choose, and each path can have a different ending.

Thus, flowcharts are a powerful tool to help you create a complex and interesting game. They help visualize the process of creating a game and allow us to better understand how each action affects the game as a whole. In addition, flowcharts help us to anticipate possible errors and look for ways to fix them.

In order to create a game, we must start by planning its scenario. We need to determine what challenges the player has to go through, what problems he has to solve, and what obstacles he has to overcome. We can use flowcharts to describe all the actions the player has to perform and identify the possible paths he can choose.

Let's come up with a game genre together, and write out the game's story

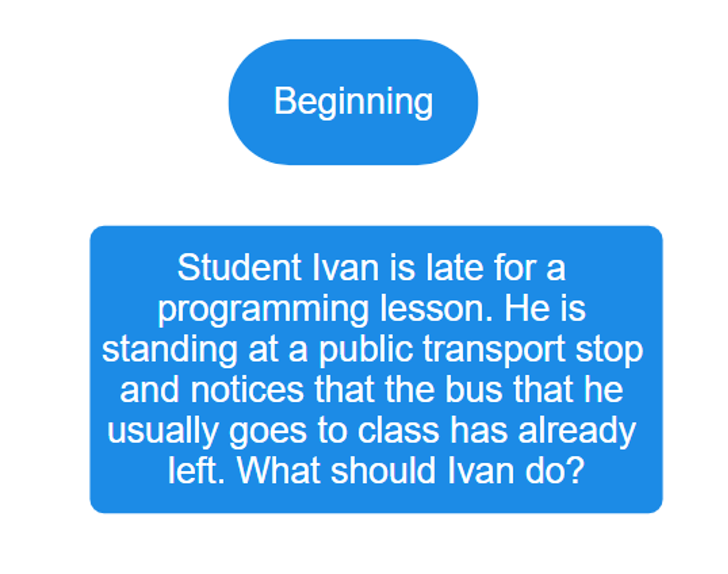
*(Writing a story about John who is late for class)*

Okay. Now that we know what kind of game we want to create, let's start working with the flowchart.

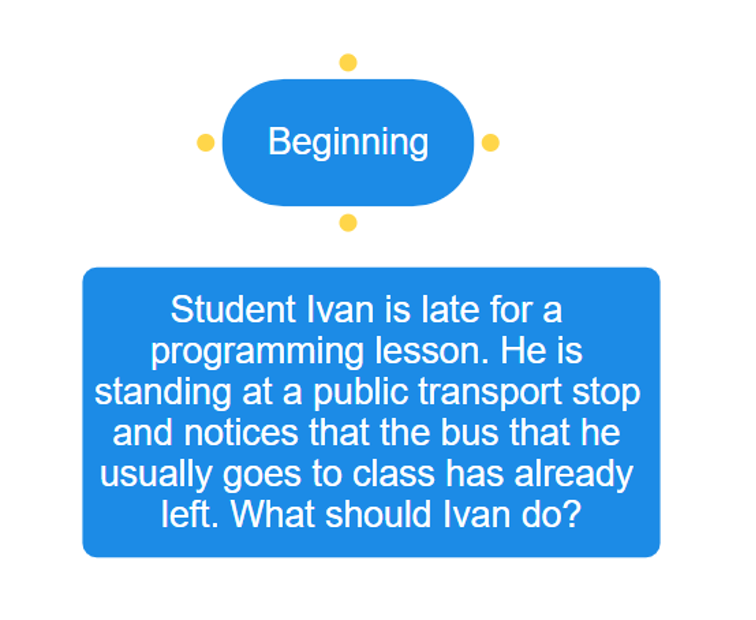
We'll start with the main block - this will be the start of the game. Then we'll add branches that will include player actions and conditions that will lead to different endings.

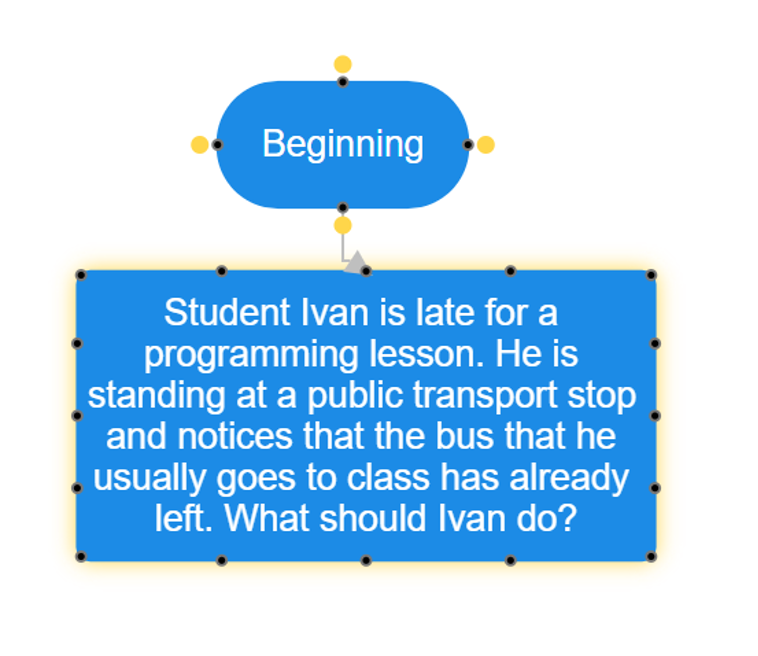
Game start

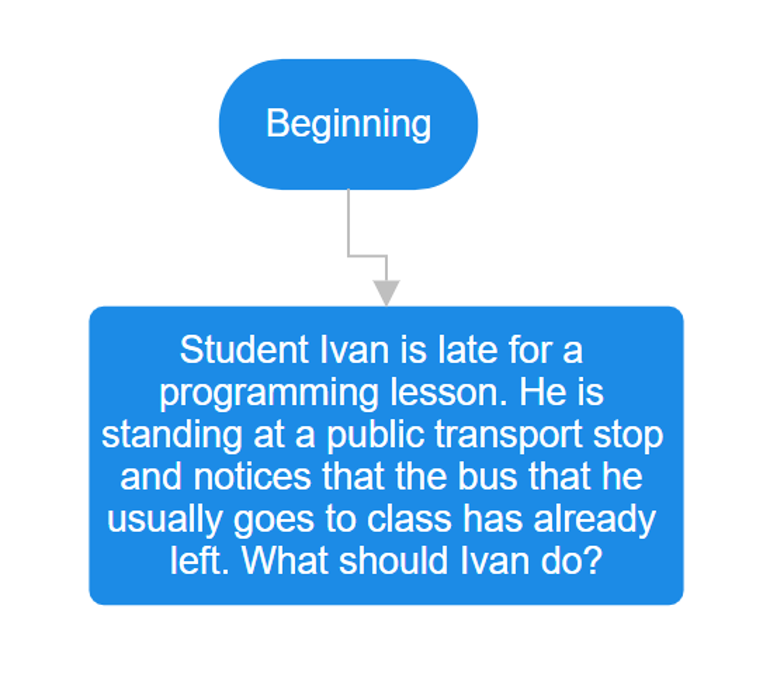
A student named John is late for his programming class. He stands at a public transportation stop and notices that the bus he usually takes to class has already left. What should John do?

[](https://kiber.one/#ca57759d-10fd-46dd-82c7-74869e052fca)

To connect these items, point to the "Start" block, click on the dot and connect it to the text block:

[](https://kiber.one/#e18913cc-8263-48a7-ab87-062e48736798)

[](https://kiber.one/#3f9b3fef-b18b-43ea-9a8c-c5d43deceacc)

[](https://kiber.one/#1eee42fe-f5d0-4abb-8c25-dfbcb01c26dd)

Now let's write the script next!

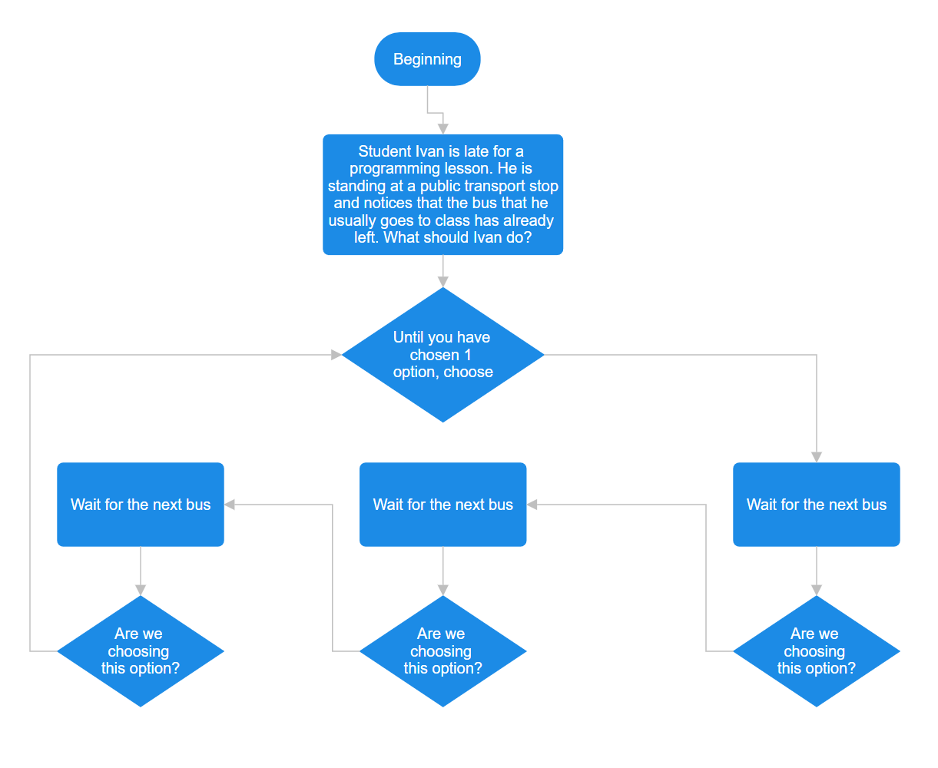
A little hint - if you don't know what exactly you can come up with next, you can use the neural network, it will always help you!

Here are the options the neural network suggested to me:

1) ask his mom to drive him to the class;

2) walk to the class;

3) wait for the next bus.

[](https://kiber.one/#009468bf-a388-44cd-9d14-d6aeab66ea23)

Option 1 - The player chooses "Ask his mom to drive him to the class."

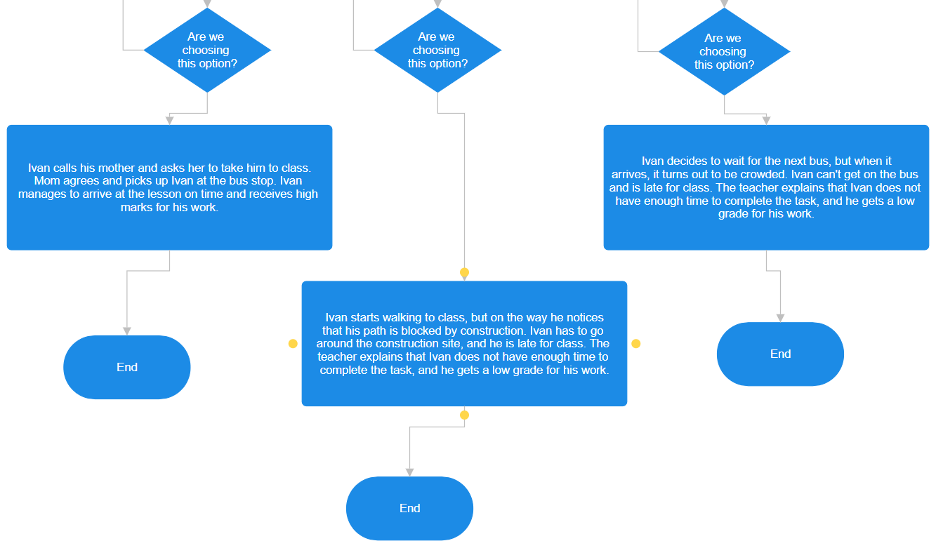
John calls his mom and asks her to take him to the class. Mom agrees and picks John up at the bus stop. John makes it to the class on time and gets a high mark for his work.

Option 2 - The player chooses "Walk to the class".

John walks to the class, but on the way he notices that his path is blocked by construction. John has to go around the construction site and is late for class. The teacher points out that John has not enough time to complete the task and receives a failing grade for his work.

Option 3 - The player chooses "Wait for the next bus".

John decides to wait for the next bus, but when the bus arrives, it is full. John cannot get on the bus and is late for the class. The teacher tells John that there is not enough time for him to complete the assignment, and he receives a failing grade for his work.

[](https://kiber.one/#d40c4a33-3753-43a5-b804-58554230aa03)

Game Ending

The player has completed the game and seen all three possible endings depending on his choices. He may decide to play the game again and try to choose a different path to get a better ending.

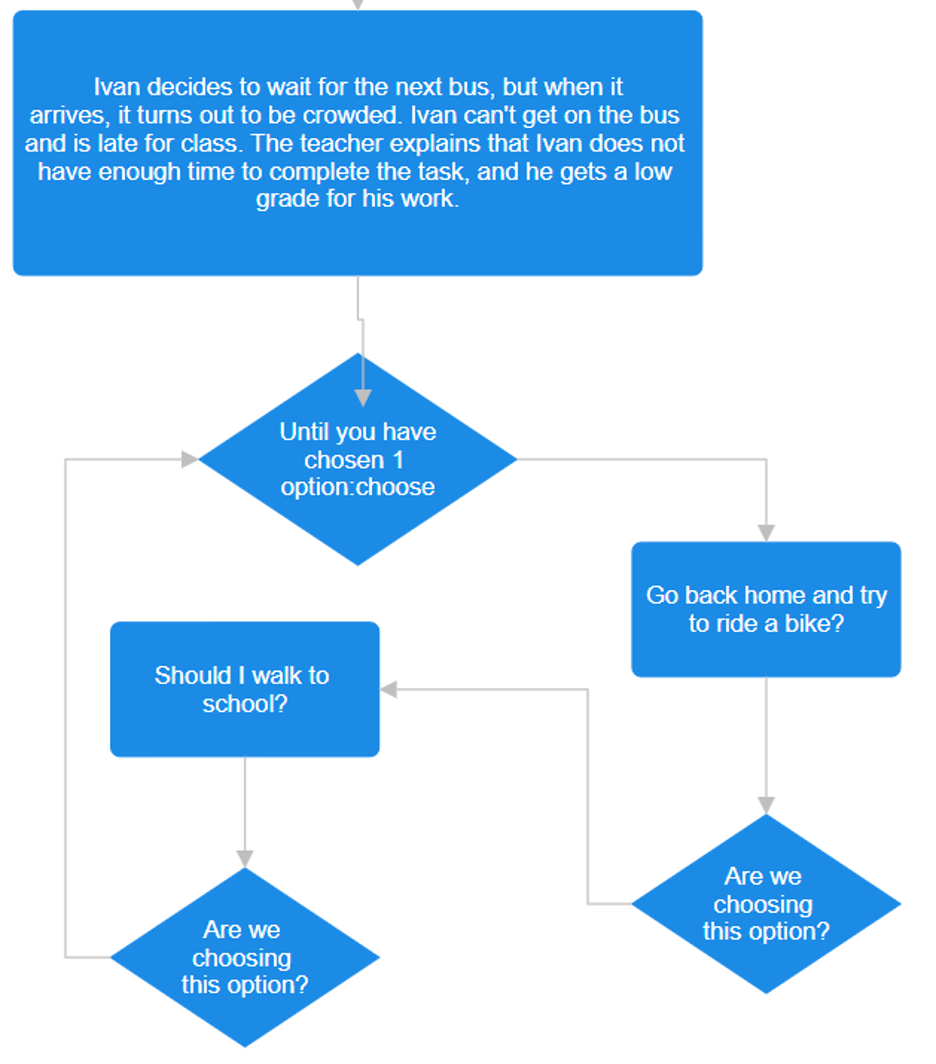
Let's try to complicate the scheme.

After choosing the "Wait for the next bus" path, the main character leaves the house and walks to the bus stop. On the way, he notices that there is a heavy fog outside and the cars are moving very slowly. Finally, the bus arrives, but it turns out that it is already full and does not stop.

Now the player has to make a choice:

1) go home and try to ride a bicycle to get there;

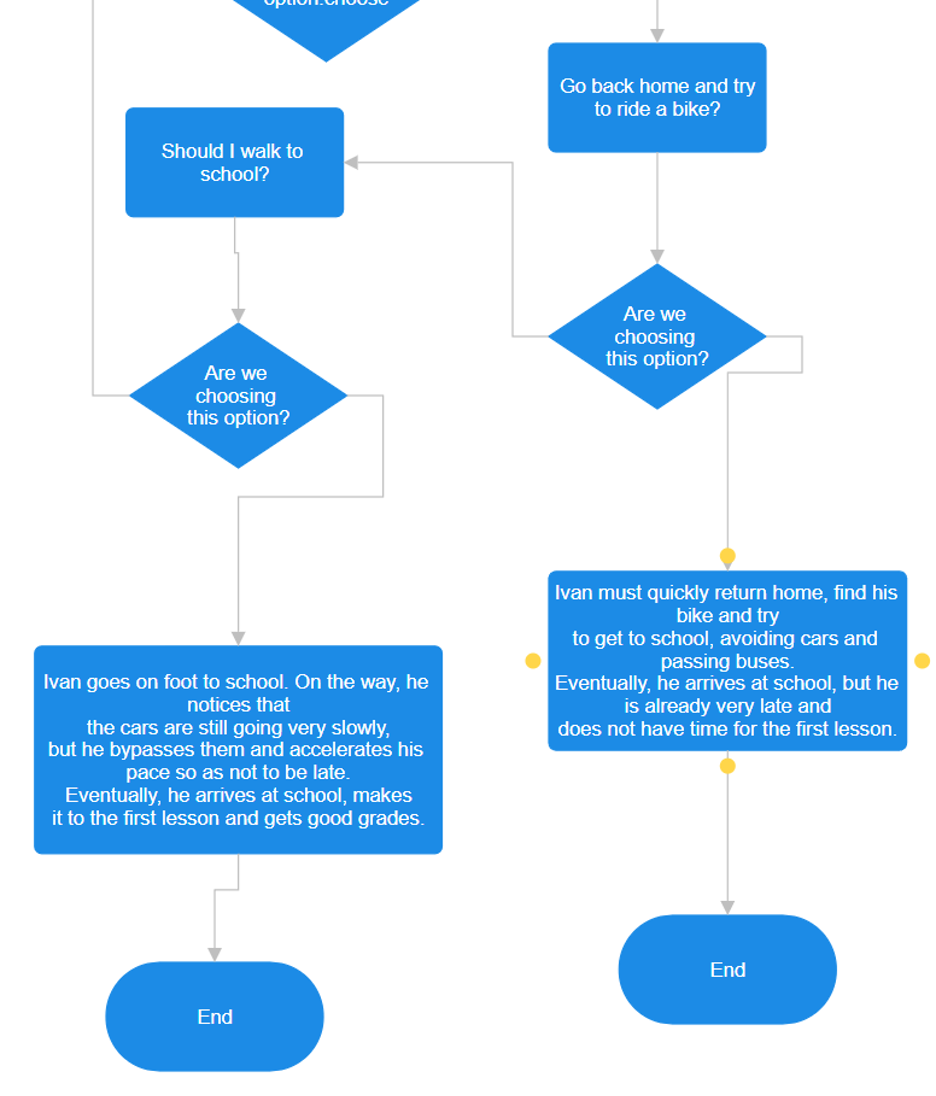
2) walk to the school.

[](https://kiber.one/#20ac4a49-5b43-4709-ba5c-8980ce29b2d9)

If the player chooses the first option, he must quickly return home, find his bike and try to ride to the school, avoiding cars and passing buses. Eventually, he arrives at the school, but he is already very late and does not make it to the first lesson.

If the player chooses the second option, he walks to the school. On the way, he notices that cars are still driving very slowly, but he goes around them and speeds up his pace to avoid being late. Eventually, he arrives at school, makes it to his first class on time and gets good grades.

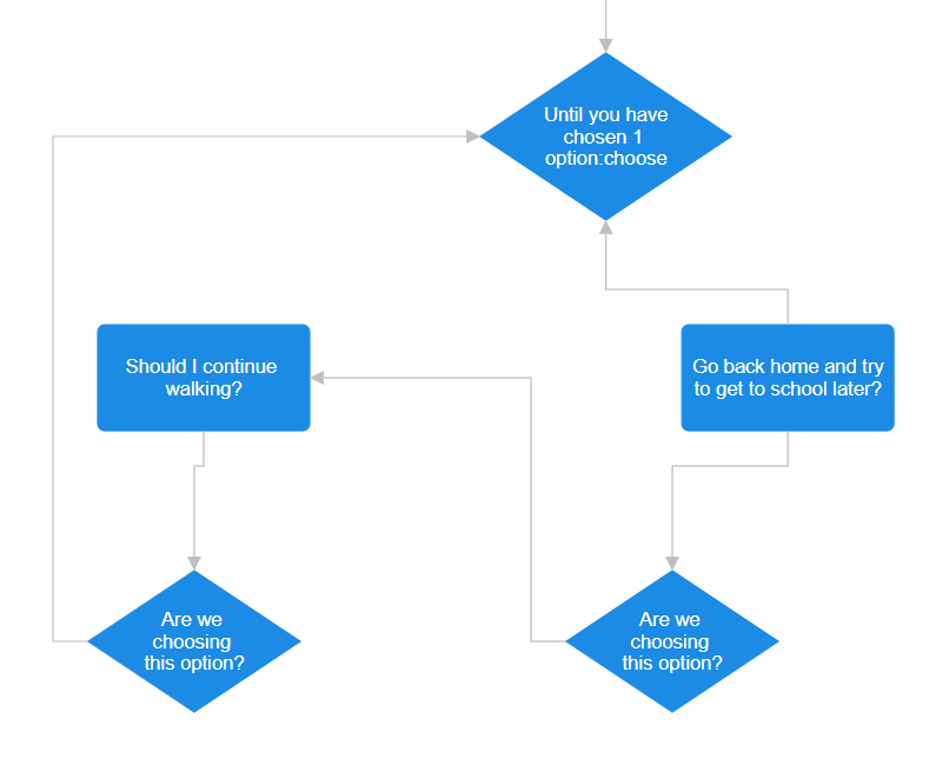
In this way, the player is given choices that affect the story and ending of the game. It is important to remember that every decision has consequences, and the player must carefully evaluate his actions to achieve the best possible result.

[](https://kiber.one/#da198340-145d-4d0f-8e5b-cc1842480d8c)

Let's try to complicate the game even more by adding an "if-then" condition.

If a player chooses the option "walk to the school" and on the way notices that the fog is getting thicker, he will have another choice:

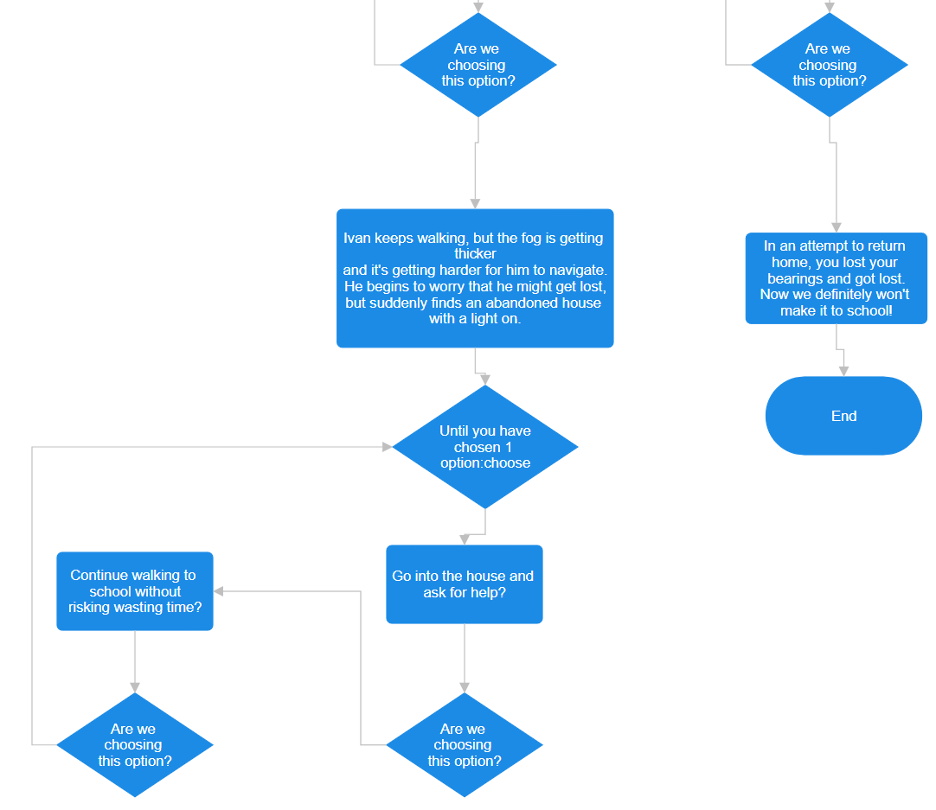
* continue walking;
* return home and try to get to the school later:

[](https://kiber.one/#658f3620-b0d4-4253-9e46-8ecb55ea3ca3)

If the player chooses the first option, he continues walking, but the fog is getting thicker and thicker, and he finds it increasingly difficult to navigate. He starts to worry that he might get lost, but suddenly he finds an abandoned house with a light on.

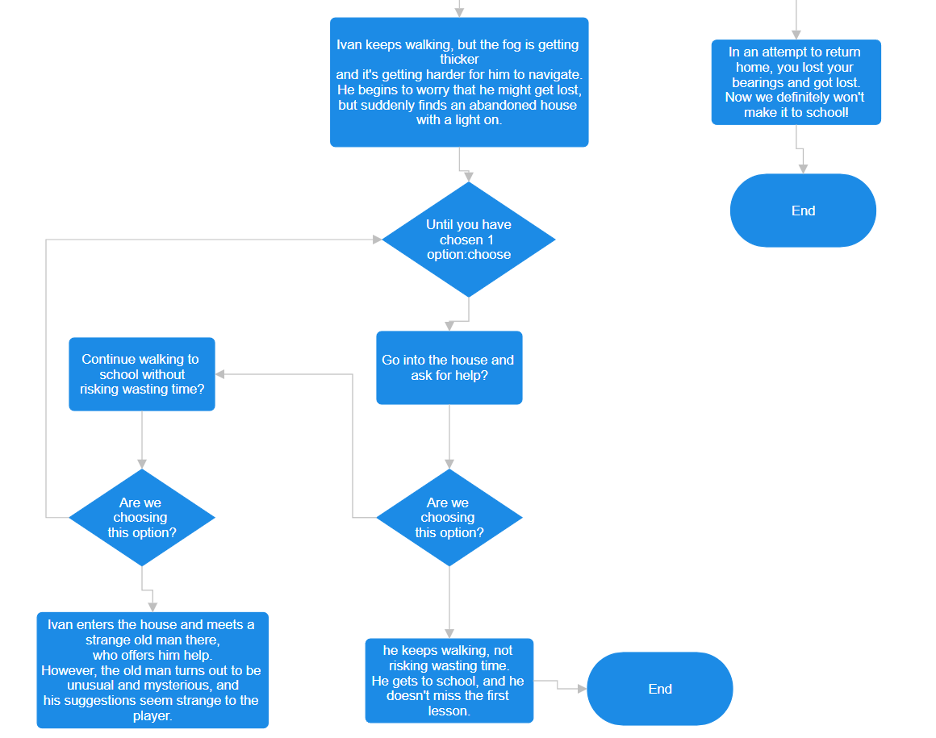
Now the player must choose what to do next:

* go into the house and ask for help;
* continue walking to the school without risk of wasting time:

[](https://kiber.one/#a96983ac-d6e9-4665-9e42-f748a76e5560)

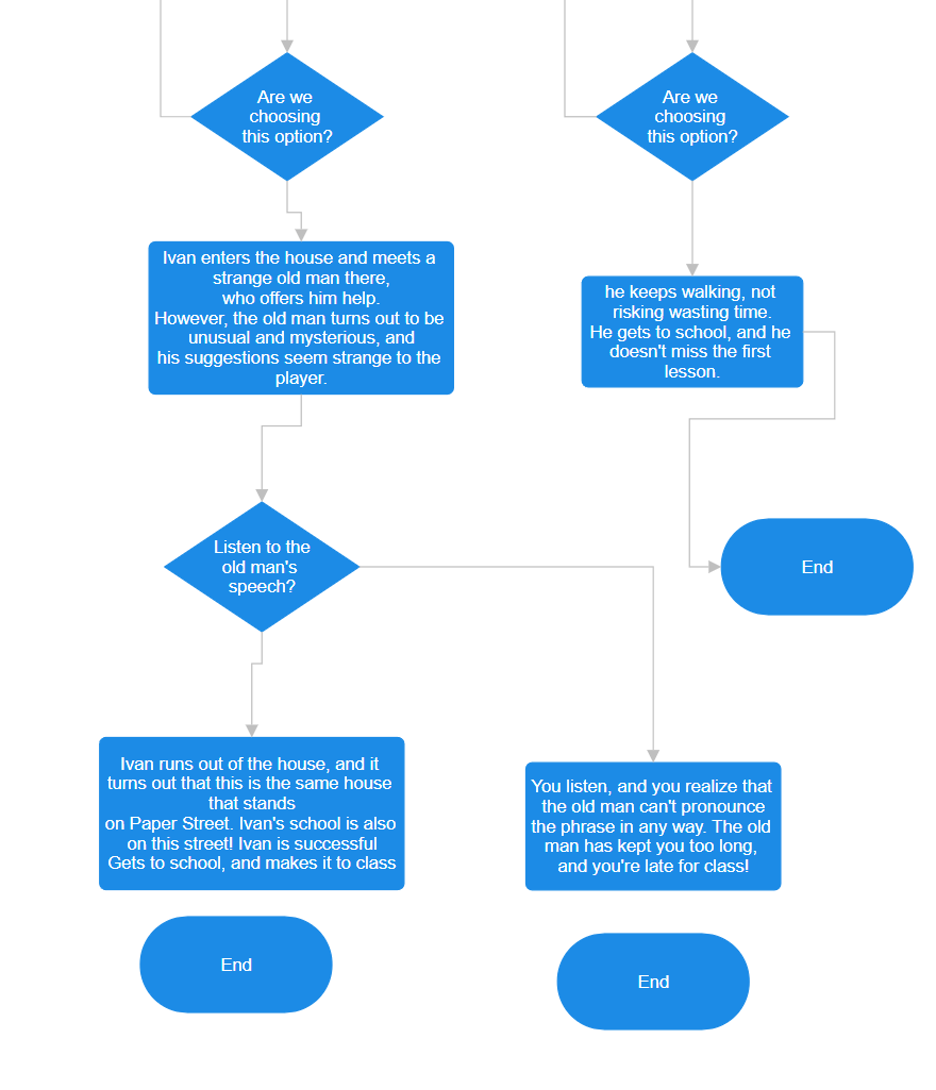
If the player chooses the first option, he enters the house and meets a strange old man who offers to help him. However, the old man turns out to be unusual and mysterious, and his offers seem strange to the player. Eventually, the player realizes that he is wasting his time and leaves the house, continuing his journey to the school.

If the player chooses the second option, he keeps walking without risking wasting time. He reaches the school and does not miss the first lesson:

[](https://kiber.one/#b30c87e4-e5bd-4fdc-8651-a7bde2fd9f7d)

Thus, the player is given even more choices that affect the story and outcome of the game. Every decision has consequences, and the player must carefully evaluate their actions to achieve the best possible outcome.

Let's finish the last twists and turns!

[](https://kiber.one/#6ae88882-b311-4514-bc58-f8a3d83dc3d8)

Now we are ready for the next step - writing a program based on our flowchart. To do this, we will use a programming language. But we will get to that in the next lessons.

*Give the children time to discuss.*

Great! We learned today how to create flowcharts and use them to write programs. I hope you have learned a lot and can use this knowledge in the future!

**Summing up - 5 min.**

Don't forget to save your projects, because we will need them in the next lesson. *(It is better to save your projects in JSON format so that we can open them in the next lesson).*

Thank you for your attention! If you have any questions, feel free to ask me.

Overall, I think the lesson went well because you were attentive and worked hard.

Summarize the results of the entire training module.

**Additional task**

Invite the children to write a story for their own game!