User-Generated Voice Navigation Editing System Using Block-Type Visual Programming Language

Daisuke Yamamoto, Sekiya Hiroki, Shinsuke Kajioka

Nagoya Institute of Technology, Japan.

IARIA

Biography

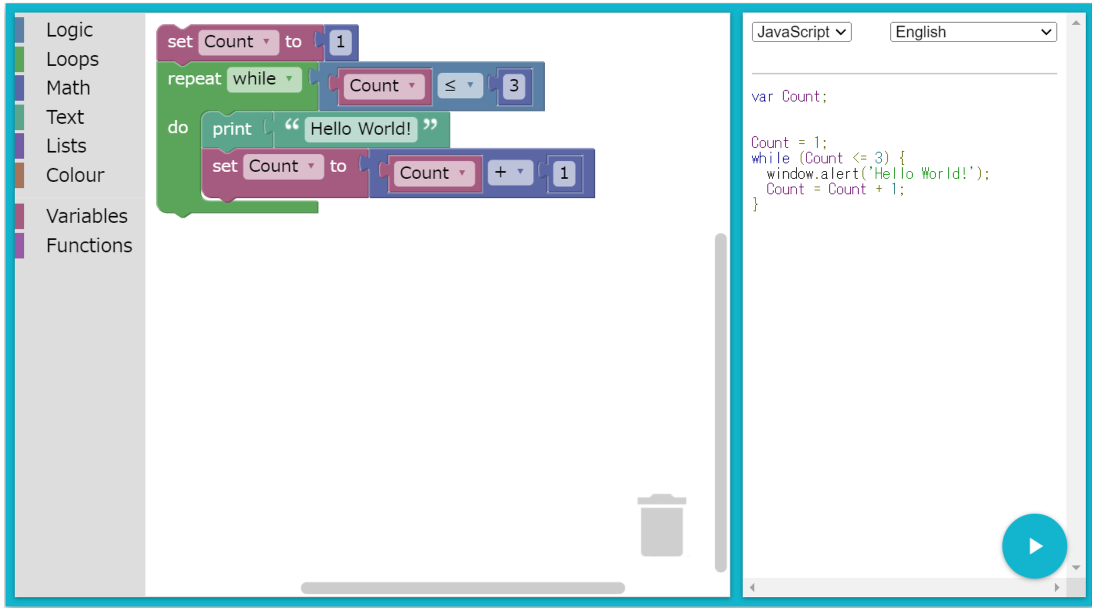
- Daisuke Yamamoto
  - Associate professor at Nagoya Institute of Technology, Japan.
  - Web services, content technologies, geographical information systems, and multimedia systems.
- Call for international joint research!
  - daisuke@nitech.ac.jp

Introduction

- **Voice navigation systems** on smartphones, like Google Maps, have become popular in recent years.
- They provide one-way navigation based on standard phrases from a map database.
- If **two-way voice navigation** were possible, it is expected that the instructions would be as polite and detailed as a human tour guide.
  - Interactive voice navigation systems for tourist areas can benefit tourism and help people reach their destination safely.

Approach

- Maps lack tourist information or supplementary navigation, making it hard to generate voice navigation based on them.
- Our focus is on **User-Generated Content** (UGC).
- UGC lets users edit content like maps and dictionaries, such as Wikipedia and OpenStreetMap.
- **Google Blockly** is a visual programming tool used in Scratch, an environment for kids to learn coding.



```
var Count;


Count = 1;
while (Count <= 3) {
  window.alert('Hello World!');
  Count = Count + 1;
}
```

Google Blockly

Aims and Requirements

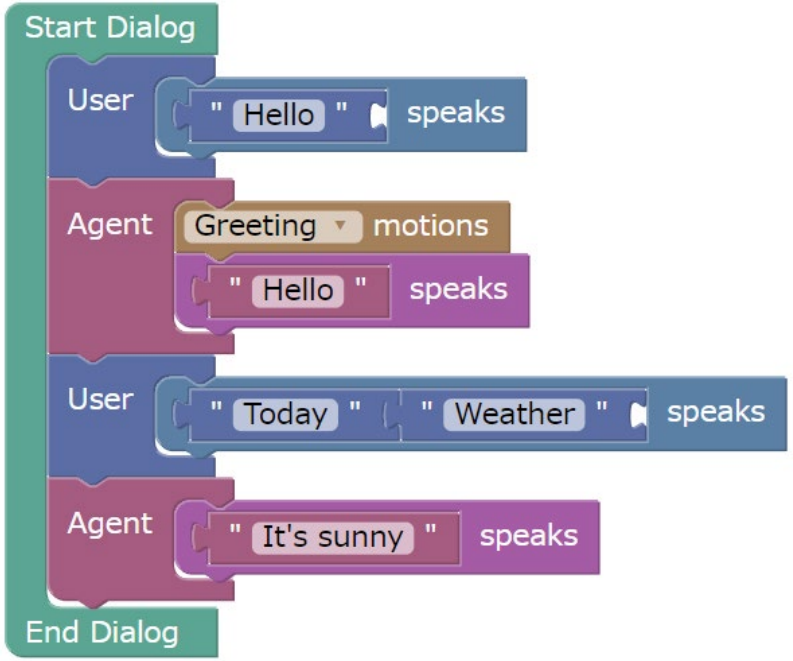
- Our study aims to develop a UGC-based mechanism that lets people easily edit two-way voice navigation content.
- Three requirements for this mechanism:
  1. Anyone should be able to edit the navigation content easily, using UGC.
  2. In tourist areas, the shortest route might not be the best. The user should be able to change the route.
  3. Pedestrians might get lost. The mechanism should still be able to direct them to their destination, even if they don't follow the suggested route.

Features

- We suggest "**spot blocks**," an extension of Google Blockly, which match blocks to map nodes called "spots."
- This lets people edit voice navigation content using a visual programming language.
- The route can be easily edited by changing **the layout and combination of spot blocks.**
  - The default state of spot blocks is the shortest route.

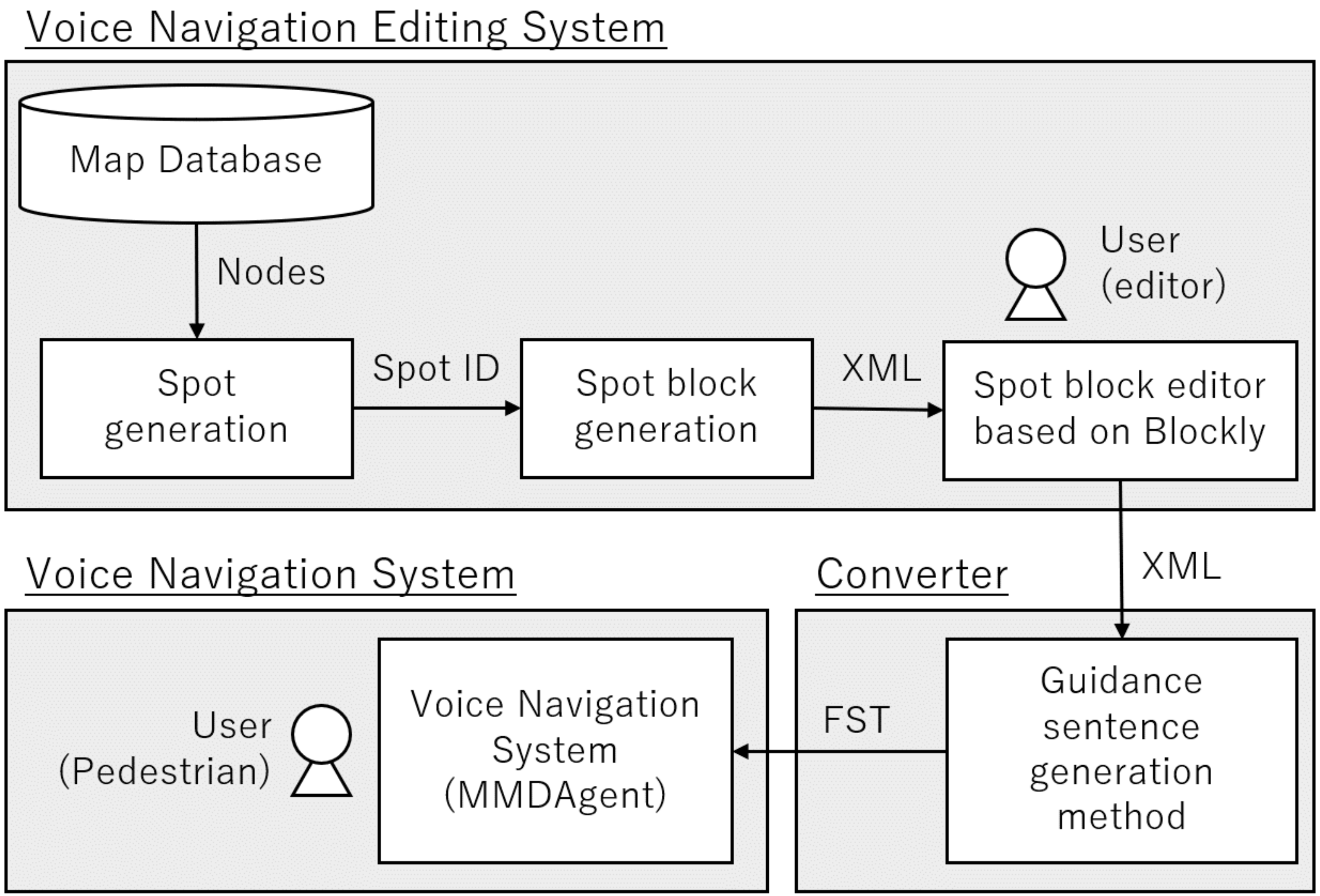
Related Work

- **Furuichi[1]** method
  - Google Blockly blocks were used to create voice interaction systems. By combining these blocks, **two-way voice interaction content** could be created.
  - An example of this is shown in Figure 1, where if a user says "Hello", the agent responds with "Hello" and bows. Similarly, if the user asks about the weather, the agent responds with "It is sunny".
  - However, this study **did not focus on navigation**.



[1] M. Furuichi, D. Yamamoto, and N. Takahashi, "Voice Interaction Scenarios Editor with Block-based Visual Programming Facilities," Transaction on IPSJ, DCON, vol. 8, No.2, pp.1–15, 2020.

System Configuration



Voice Navigation Editing System

Map Database

Nodes

Spot generation

Spot ID

Spot block generation

XML

Spot block editor based on Blockly

User (editor)

XML

Voice Navigation System

Converter

User (Pedestrian)

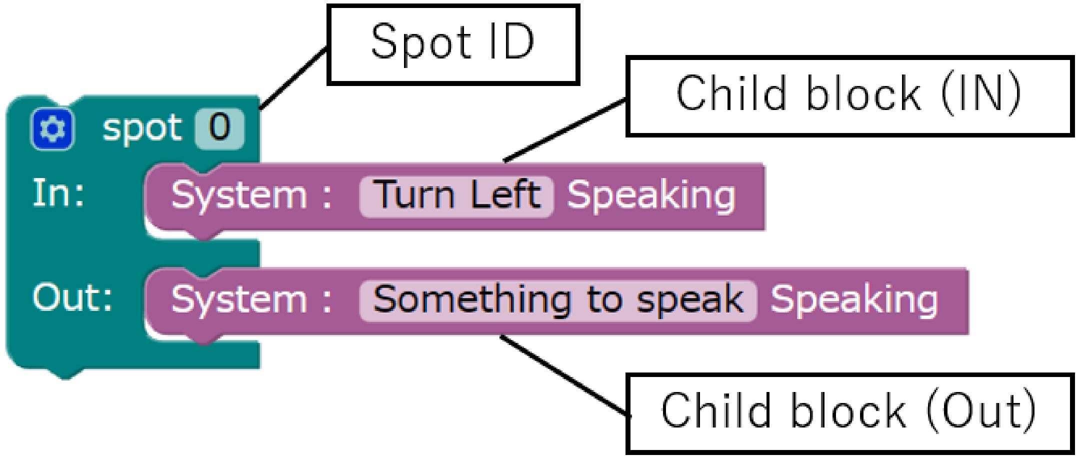
Voice Navigation System (MMDAgent)

FST

Guidance sentence generation method

Spot Generation

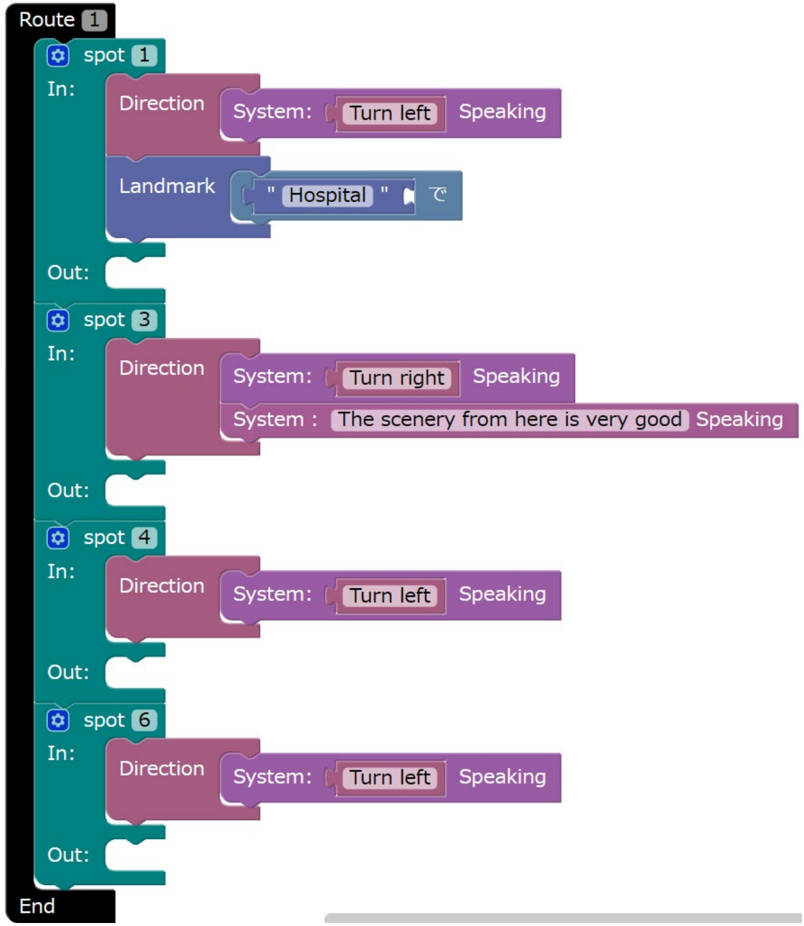
- A **spot** is the smallest unit for navigation and corresponds to an intersection node
- Construction of a **spot network**
  - Enables the calculation of the distance and direction
- Minimum amount of voice navigation can be generated
  - without using the Spot Block Editing function


Spot ID
Child block (IN)
spot 0
In:
System : Turn Left Speaking
Out:
System : Something to speak Speaking
Child block (Out)

Spot Block

Spot Block Editing

- Spot block proposed as a new block for describing navigation, corresponding in a one-to-one manner with map spots
- Spot block has corresponding spot ID, child block that fires when user approaches spot, and child block that fires when user leaves spot
- Spot blocks can be linked back-to-back to determine order of navigation and user can link spots to create their own route
- Child blocks can be added to enable voice interactions at spots



Route 1

spot 1
In:
Direction System: Turn left Speaking
Landmark " Hospital "
Out:

spot 3
In:
Direction System: Turn right Speaking
System : The scenery from here is very good Speaking
Out:

spot 4
In:
Direction System: Turn left Speaking
Out:

spot 6
In:
Direction System: Turn left Speaking
Out:

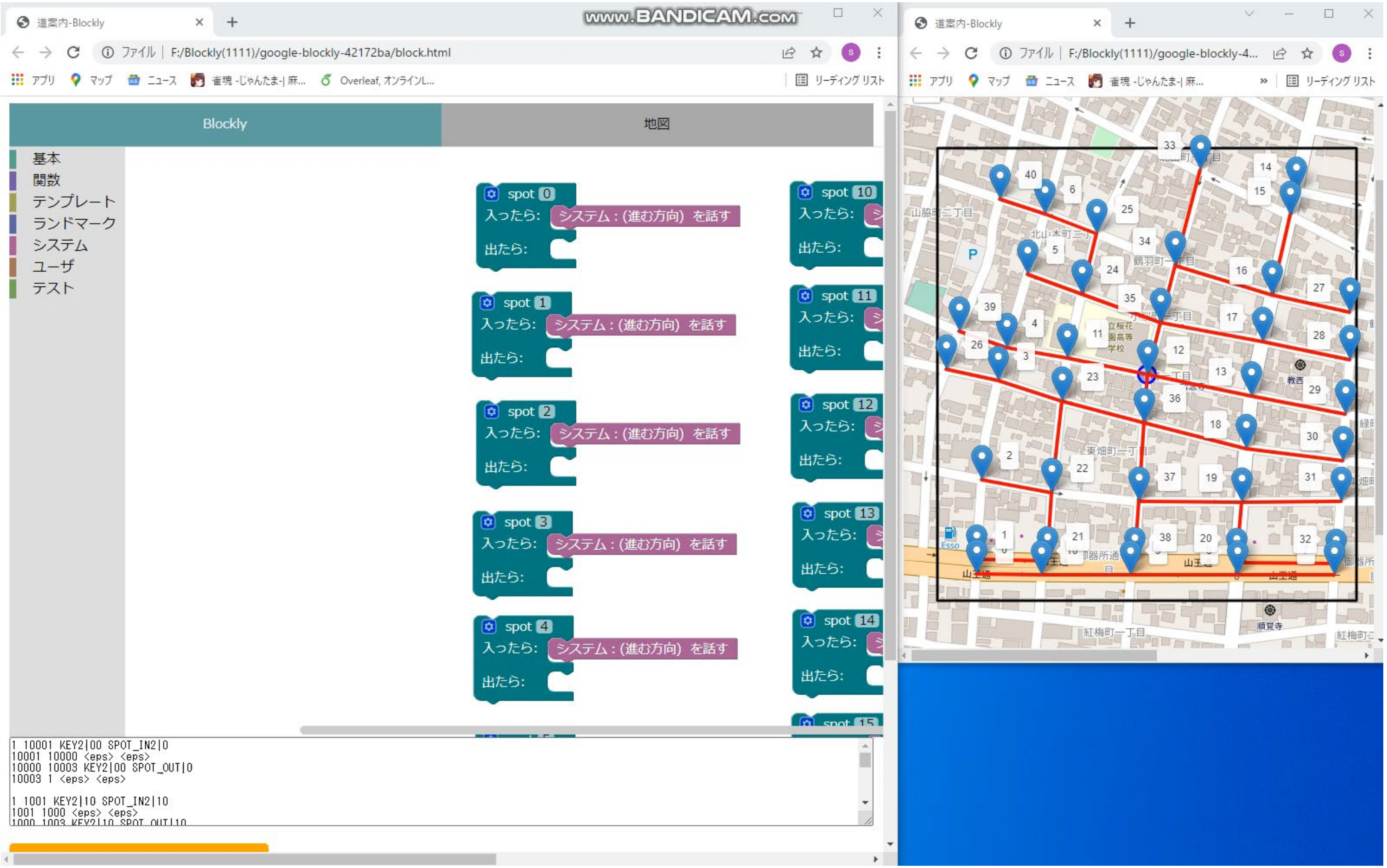
End

Figure 4. Example of linked spot blocks with navigation set in order of spot ID 1, 3, 4, and 6 and supplementary interaction becoming possible at spot ID 3

Navigation Sentence Generation

- This slide explains how navigation sentences are generated for a Japanese voice interaction system using spot blocks.
- The direction and distance to the next spot are determined based on the IDs of the current, next, and preceding spots.
- The navigation route can be changed by connecting or separating spot blocks, which updates the NextSpotID of the corresponding spots.
- The shortest route is used to set the NextSpotID when spot blocks are separated.

Demo (in Japanese)



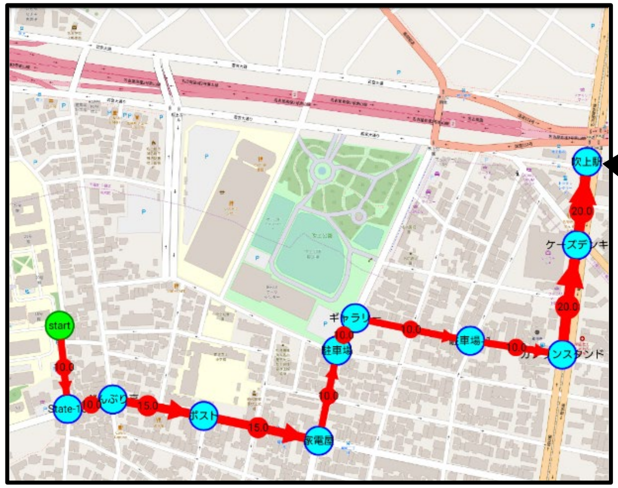
```
1 10001 KEY2|00 SPOT_IN2|0
10001 10000 <eps> <eps>
10000 10003 KEY2|00 SPOT_OUT|0
10003 1 <eps> <eps>

1 1001 KEY2|10 SPOT_IN2|10
1001 1000 <eps> <eps>
1000 1003 KEY2|10 SPOT_OUT|10
```

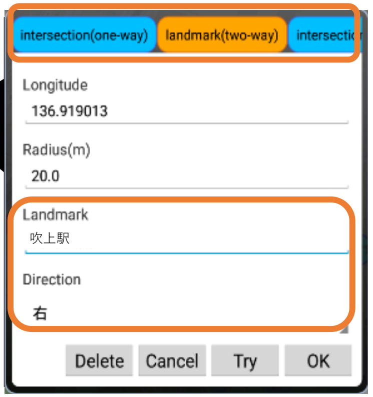
Experiment

- Purpose:
  - To evaluate the user-friendliness of the proposed system for creating voice navigation.
- Method:
  - Participants: 8 students from our university.
- Evaluation criteria:
  - Evaluation based on the time taken to create the scenarios, SUS score, and a 5-point questionnaire.
  - The SUS score is used as an index to measure the satisfaction with the system (the average score is 68).

Experimental Method

- Conventional method
  - Create a state transition by connecting the turning points on the map with roads.
  - Edit templates to describe the content of the dialogue.




intersection(one-way) | landmark(two-way) | intersectio

Longitude
136.919013

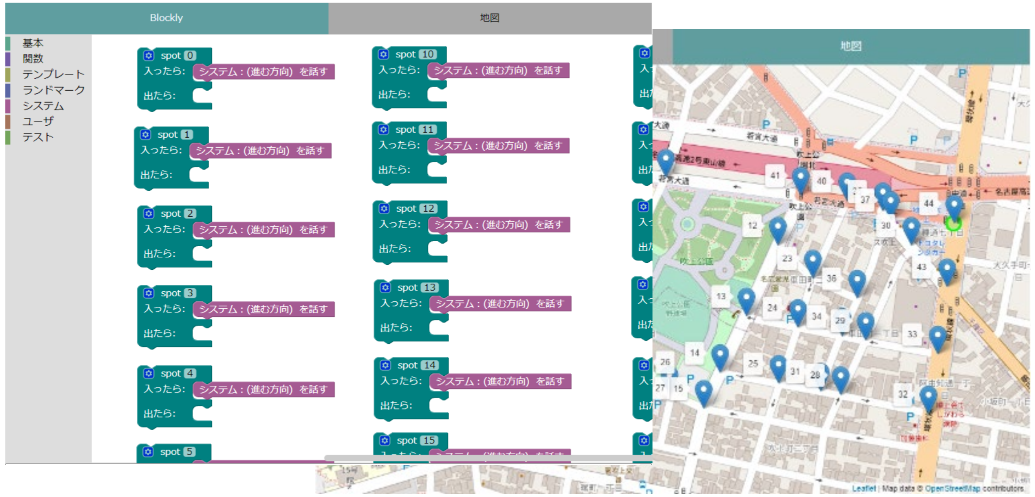
Radius(m)
20.0

Landmark
吹上駅

Direction
右

Delete | Cancel | Try | OK

- Proposed method
  - Present a map with 44 spots, including the start and goal locations.
  - Use a navigation editing system with Blockly to create the navigation guide.



Experimental Result (1/2)

- Methodology:
  - Prototype system used to determine 10 destination points and place spots within 500m, 1km, 1.5km, and 2km range around each destination.
  - Dijkstra's algorithm used to automatically generate routes from arbitrary spots to destinations.
  - Verification of possibility to reach destinations using automatically generated routes.
  - Average arrival rate for each distance recorded and presented in Table II.

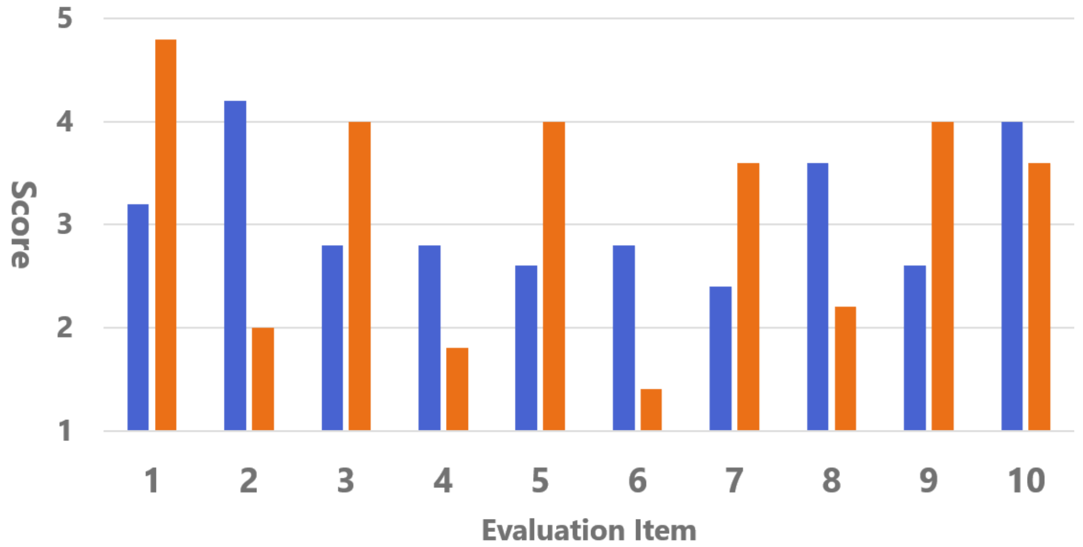
	500 m	1 km	1.5 km	2 km
Arrival rate(%)	98.75	99.01	99.35	99.81

- Results:
  - Arrival rate exceeded 98% for all distances.
  - Minimal voice navigation needed without user editing.
  - Allows users to concentrate on tourism and supplementary navigation.

Experimental Result (2/2)

- Result
  - Proposed method scored higher than the conventional method.
  - Proposed method also showed a decrease in the creation time, and an increase in the number of turning points.
- Discussion
  - The block-type description format of the proposed method is effective in creating voice navigation scenarios.

	Conventional	Proposed
SUS score	45.5	61.5
Creation time	170 seconds	157 seconds
Waypoint	4.8	6.2



Details of SUS score relating to editing interface.
Left: Hayashi et al. method, right: proposed method.

Conclusion

- User-generated voice navigation using visual programming.
  - It allows users to easily edit voice navigation scenarios with two-way voice interaction.
  - It to be effective in generating directions, with a 98% or higher arrival rate within a 2 km range.
  - The proposed method also scored higher in usability tests.
- Future work:
  - Improving the editing function's UI
  - Integrating multiple navigation content.