Missing Pieces: A Spatially Interactive Puzzle Game

Jessica Lee, Chris Shaw and Philippe Pasquier

School of Interactive Arts and Technology, Simon Fraser University
yichiehl@sfu.ca, shaw@sfu.ca and pasquier@sfu.ca

Abstract

In this project, I designed and implemented a spatially interactive puzzle game. In the game, player's position is detected by the system and player is allowed to interact with the puzzle by physically moving in the space, as if the puzzle exists in the real world. I believe spatial interaction provides a more immersive user experience and, in certain applications, is more intuitive than traditional interfaces in which users rely on mouse and keyboard inputs to interact with the system. In the future, I envision the system to be extended to fully support three dimensional interactions, including tilt gestures and acceleration movements.

Introduction

Ever since personal computers became prevalent in homes and offices, mouse and keyboard-based GUIs have been the major way of human-computer interaction. In past decades, with the growing popularity of mobile devices, new interaction styles such as multi-touch gestures have been investigated and put into commercial products. Meanwhile, researchers have been studying and exploring possibilities of next generation human computer interaction interfaces, for example virtual reality (VR), augmented reality (AR), wearable interfaces.

In this project, I am particularly interested in AR and spatial computing, in which users use the space around them as an interface to interact with computer systems. I implemented a spatially interactive puzzle game to explore the possibilities of leveraging spatial interaction in 3D human-machine interaction, where users interact with computer systems based on three-dimensional spatial inputs.

The source code of this project is hosted on the github repo: *https://github.com/qpiu/Missing-Pieces*. A short demo video of the project can be found on Vimeo: *https://vimeo.com/377880896*.

Motivations

This project is motivated by several previous works related to using spatial inputs to interact with digital systems and augmented reality applications.

Kaufmann, et al. [4] studied the use of spatial interaction in navigation tasks. They found the users who used spatial input to perform navigation tasks had 41% better spatial memory performance compared to users who used classic touch input. In [5], Büschel, et al. investigated ways to leverage spatially tracked mobile devices to support 3D data visualization tasks.

In Augmented Reality (AR), virtual objects are integrated into real world in real time [7]. AR technology provides an immersive user experience, as it permits users to interact with virtual objects using gestures, actions and movements in real world. Mathews, M. et al. [6] designed a memory game with an 2D version and an AR version. The results showed that participants were more accurate in the AR version memory game. Moreover, the technology is particularly suitable for applications related to the spatial environment. For example, the use of projected AR to support real world navigation [8], using an AR assistant to help search for products in a supermarket [9].

These works have shown the advantages and potential of spatial interaction and AR.

System description

There are three components in the system: The Missing Pieces web application that runs in a browser, a position tracking component, and a game controller (as shown in Figure 1). The components are connected through wireless local network.

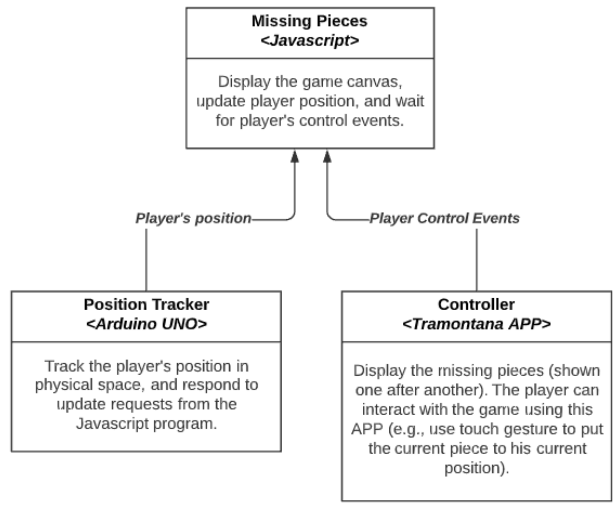


Figure 1. The three components in the system.

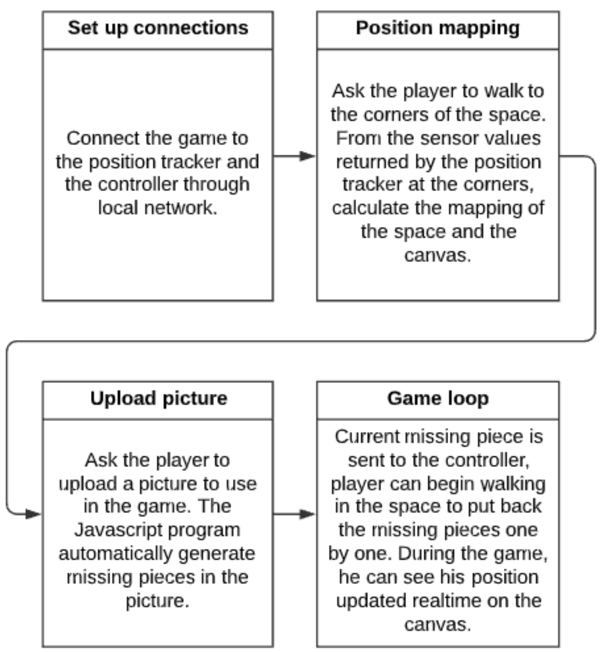


Figure 2. The system execution flow.

Execution Flow

The brief system execution flow is shown in Figure 2. There are four phases in the system.

In phase 1, player enters the IP addresses of the position tracker and the controller to connect the three components.

In phase 2, player is asked to walk to the upper-left corner and the lower-right corner (as shown in Figure 3). Missing Pieces uses the sensor values at these two corners to calculate the space dimensions and the mapping between the space and the canvas.

In phase 3, player chooses a picture from the local disk to play in the game. Missing Pieces automatically generates pieces from the selected picture, and randomly selects three pieces to be the missing pieces.

In phase 4, the program enters the game loop where the game begins. Player can start to walk around the space to put each missing piece back. At each iteration of the loop, Missing Pieces sends requests to the position tracker and update player position to the canvas, and listens to the control events from the controller. When a control event is triggered, Missing Pieces program checks if the player is at the correct position of the current piece. If he is at the correct position, the current piece will be placed back to the canvas and next missing piece will be sent to the controller. When the last missing piece is put back, the game ends.

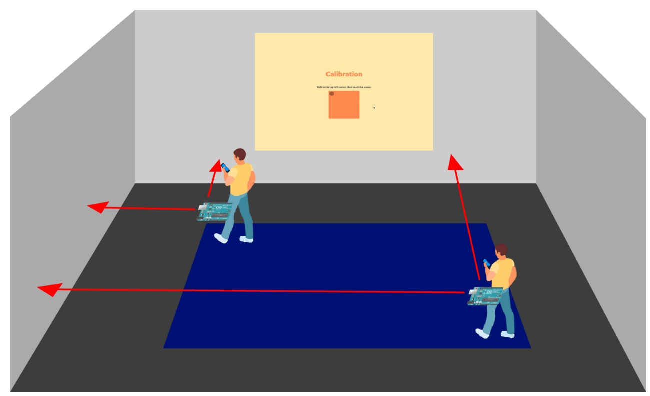


Figure 3. Position mapping phase.

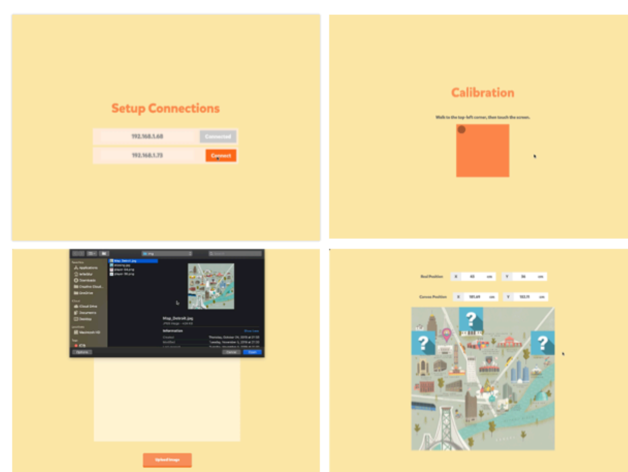


Figure 4. Missing Pieces web application.

Missing Pieces

The Missing Pieces web application is the main part of the system, where the player connects the game to other two components, performs position mapping, uploads picture, and uses spatial inputs to play the game (as shown in Figure 4). Missing pieces is implemented in Javascript. I used several Javascript libraries in the system, including JQuery[1] (for sending the http get requests to position tracker and handling the response and manipulating DOM elements), GSAP [2] (for creating animations in the web application), Tramontana [3] (for subscribing to sensor events on mobile phones).

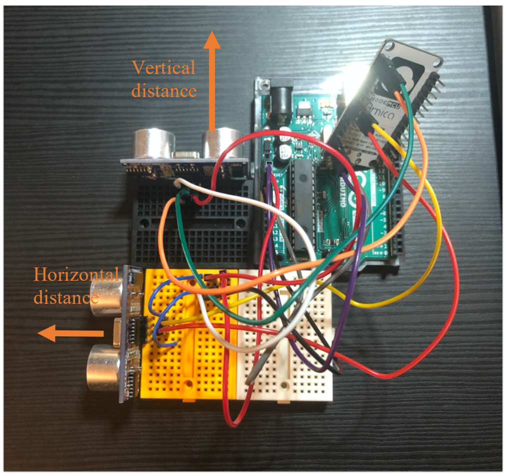


Figure 5. The position tracker is implemented using an arduino board with a WIFI shield and two ultra-sonic sensors.

Position Tracker

The position tracker component is implemented using an Arduino UNO board with a WIFI shield. Two ultra-sonic sensors (model HC-SR04) are connected to the WIFI shield, one for measuring the vertical distance to the upper boundary of the space, another for measuring the horizontal distance to the left boundary of the space (see Figure 5).

Currently, the information of the WIFI network connection (i.e., SSID and password) is hardcoded in the arduino program. After the position tracker is powered on, it first connects to the WIFI network, shows the IP address of the position tracker, then starts a server on port 80 (sample log outputs can be found in Figure 6). The position tracker continuously listens to client requests and returns the values of the two ultra-sonic sensors in JSON format.

Controller

In the project, I chose to use a mobile device as a tangible game controller. The player needs to first download and install Tramontana APP for iOS or Android (see Figure 7). Tramontana APP is a companion APP of the Tramontana Javascript library [3], which I used in the Missing Pieces web application. Through the APP, the sensor events on mobile devices become accessible to web applications.

In Missing Pieces web application, I used Tramontana API to subscribe to touch events on the connected mobile device and defined a handler for the event. When a touch event is triggered, Missing Pieces will be notified, and the handler is called to handle the event.

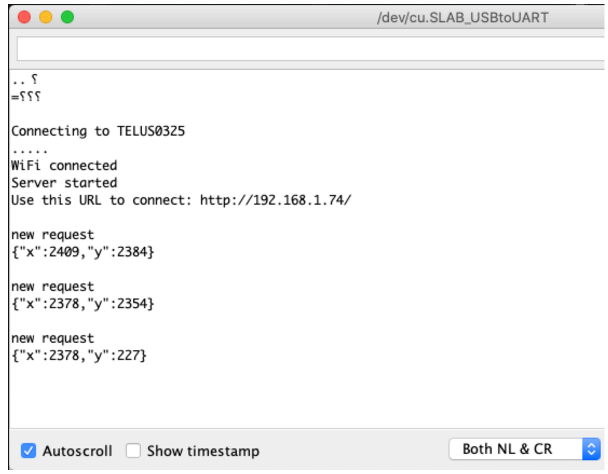
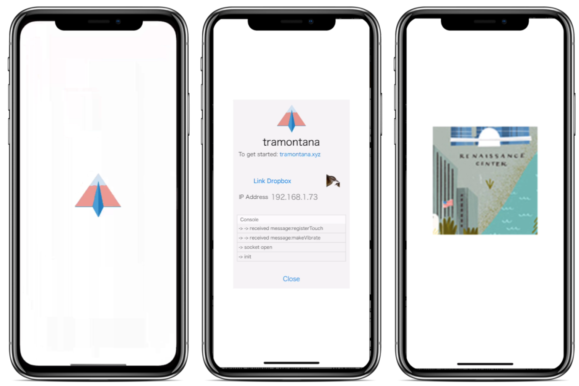


Figure 6. Log messages in Arduino console.



(a) (b) (c)

Figure 7. (a) Tramontana APP is used as the controller of the system. (b) After starting the APP, LAN IP address of the device is displayed on the screen. (c) Current missing piece is displayed on the screen during the game. Player can put the piece to his current position by touching the screen.

System implementation (UML)

The Missing Pieces web application is object-oriented, the UML class diagram can be found in Figure 8.

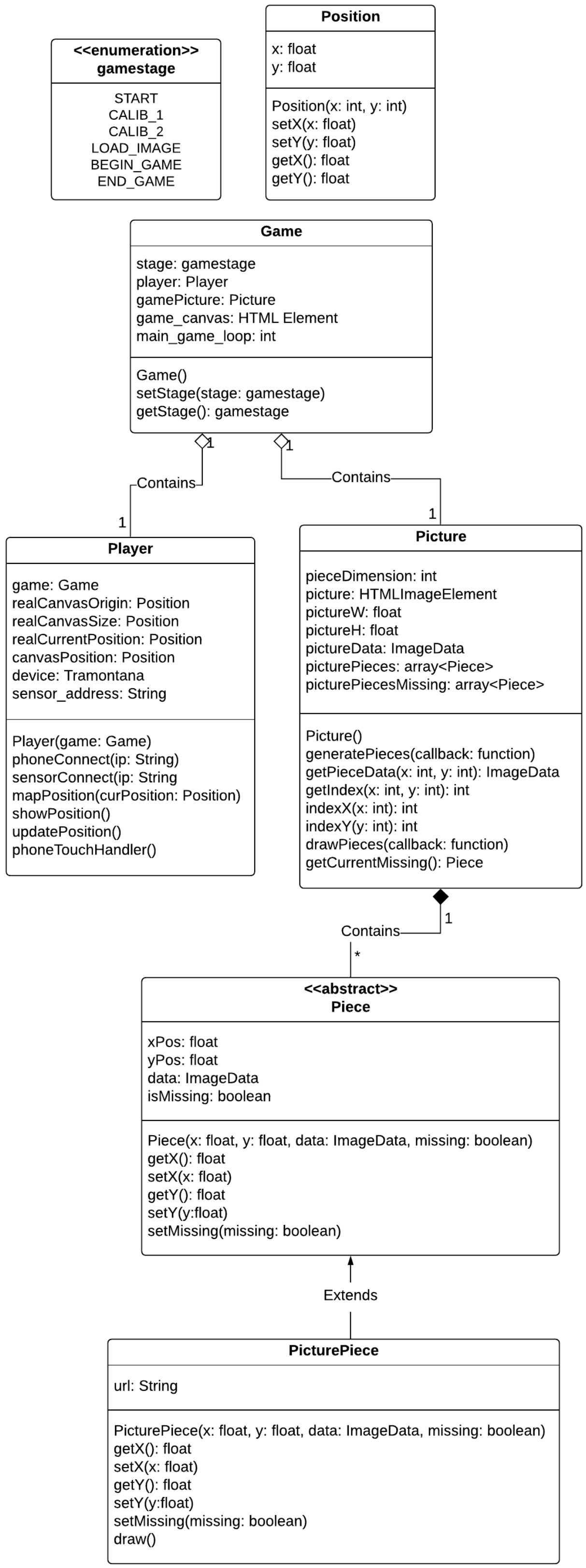


Figure 8. UML diagram of the system.

First, a **Game** object will be created for the game, which contains an instance of **Player** class and an instance of **Picture** class. The **Player** class keeps track of the Tramontana APP (*device*), the position tracker (*sensor_address*), and information for performing position mapping. The **Picture** class stores data about the picture uploaded to use in the game, generate pieces from the picture, and stores generated pieces in arrays (*picturePieces* and *picturePiecesMissing*). The generated pieces are instances of class **PicturePiece**, which is a subclass of the abstract class **Piece**. In the **Piece** abstract class, properties are defined to store the correct position of the piece, image data of the piece, and a Boolean value indicating whether the piece is one of the missing pieces. **PicturePieces** extends the abstract **Piece** class and methods are implemented to override the abstract methods.

Limitation

First of all, currently the position tracking component of the system is limited to two-dimensional. Therefore, users can only generate spatial inputs by moving within two-dimensional space. Moreover, the ranging distance of the ultra-sonic sensor (HC-SR04) is 2 centimeters to 500 centimeters, which means the space dimensions are limited to within 5 meters by 5 meters.

Secondly, ultra-sonic sensors emit high frequency sound which travels in the air, and use the time between transmission and reception to calculate the distance to an object. Therefore, to use ultra-sonic sensors to measure the distances and calculate position mapping, it is required that two fixed flat surfaces are present at the front and left side of the area (as shown in Figure 3).

Conclusion & Future work

In this project, I propose a prototype of an interactive system using spatial computing.

In the future, I envision the project can be further extended to support three-dimensional position tracking and a variety of gestures that can be detected on mobile devices (e.g., accelerometer events, gyroscope events, force of touch events). Moreover, various alternative display types (e.g., large wall display, floor display) can be tested in the system.

Acknowledgements

I would like to show my gratitude to Tom Froese, who kindly granted the use of one of his map works (*https://www.tomfroese.com/maps*) in the Missing Pieces demo video. The use of the map is only for educational use.

References

[1] jQuery. (n.d.). https://jquery.com/

[2] GreenSock. (n.d.). https://greensock.com/

[3] Tramontana. (n.d.). https://tramontana.xyz

[4] Kaufmann, B., & Ahlström, D. (2013). Studying spatial memory and map navigation performance on projector phones with peephole interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 3173–3176. https://doi.org/10.1145/2470654.2466434

[5] Büschel, W., Reipschläger, P., Langner, R., & Dachselt, R. (2017). Investigating the use of spatial interaction for 3d data visualization on mobile devices. *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, 62–71. https://doi.org/10.1145/3132272.3134125

[6] Mathews, M., Challa, M., Chu, C.-T., Jian, G., Seichter, H., & Grasset, R. (2007). Evaluation of spatial abilities through tabletop AR. *Proceedings of the 8th ACM SIGCHI New Zealand Chapter's International Conference on Computer-Human Interaction: Design Centered HCI*, 17–24. https://doi.org/10.1145/1278960.1278963

[7] Azuma, R.T. (1997). A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments Vol. 6, Issue 4*, 355-385.

[8] Colley, A., Häkkilä, J., Forsman, M.-T., Pfleging, B., & Alt, F. (2018). Car exterior surface displays: Exploration in a real-world context. *Proceedings of the 7th ACM International Symposium on Pervasive Displays*, 7:1–7:8. https://doi.org/10.1145/3205873.3205880

[9] Li, M., Arning, K., Bremen, L., Sack, O., Ziefle, M., & Kobbelt, L. (2013). ProFi: Design and evaluation of a product finder in a supermarket scenario. *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, 977–984. https://doi.org/10.1145/2494091.2496007