Bridging Tangible and Virtual Interaction: Rapid Prototyping of a Gaming Idea

Thov Reime, Heikki Sjöman, Achim Gerstenberg, Pekka Abrahamsson, Martin Steinert

► To cite this version:

Thov Reime, Heikki Sjöman, Achim Gerstenberg, Pekka Abrahamsson, Martin Steinert. Bridging Tangible and Virtual Interaction: Rapid Prototyping of a Gaming Idea. 14th International Conference on Entertainment Computing (ICEC), Sep 2015, Trondheim, Norway. pp.523-528, 10.1007/978-3-319-24589-8_50 . hal-01758412

HAL Id: hal-01758412

https://inria.hal.science/hal-01758412

Submitted on 4 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Bridging Tangible and Virtual Interaction: Rapid Prototyping of a Gaming Idea

Thov Reime[a], Heikki Sjöman[a], Achim Gerstenberg[a], Pekka Abrahamsson[b], Martin Steinert[a]

[a] Department of Engineering Design and Materials, NTNU, Richard Birkelands vei 2B, 7492
Trondheim, Norway
{Heikki.Sjoman, Achim.Gerstenberg, Martin.Steinert}@ntnu.no, Thov@stud.ntnu.no
[b] Department of Computer and Information Science, NTNU, Sem Sælands vei 9, 7491
Trondheim, Norway
pekkaa@ntnu.no

Abstract. The Fibo Car is an example for a game interface that allows a user to modify a virtual car in a racing game through assembling tangible car parts. This paper describes the 6 week development journey towards a fully functional proof of concept prototype, reflections on the process as well as the technical details of the prototype.

1 Introduction

The basic idea of the Fibo Car game project is that the player can construct a real world car model out of tangible building blocks. The structure of the model is digitally recognized and it influences the properties of the virtual model in the car racing game.

In this paper we focus solely on the development of the tangible objects, the structure recognition, and a virtual representation without any gameplay. The game idea is related to games like Kerbal Space Program [1] or Besiege [2] except that the constructing takes place tangibly like in LEGO Mindstorms [3].

The solution presented here has one central part that can detect the attached neighboring parts. The identification is realized by measuring and identifying part-specific resistances with an Arduino Uno microcontroller board [4] wired to a PC. A virtual representation of the identified tangible model is then shown on a screen. The latest version of the prototype is shown in figure 1.

The upcoming section describes the development journey of the first 6 weeks. It includes failures, dead ends, and gives reasons for the actions taken. We concentrated on development speed using a process with rapid iteration cycles in favour of fast learning and quick improvement without project control by predefining requirements and a priori budgeting.

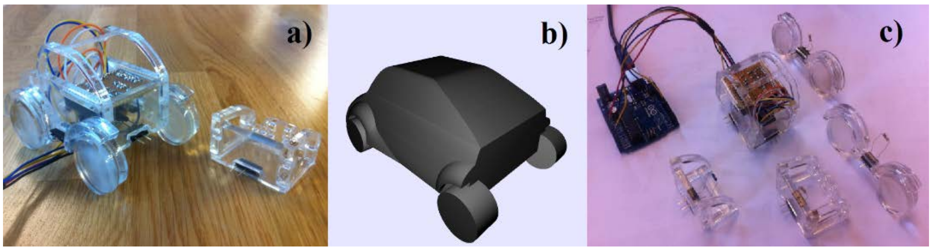

Fig. 1. Latest proof of concept prototype showing the tangible model in a) and the corresponding virtual representation in b). All available tangible car parts of the latest prototype are shown in c).

2 Development Journey

The project started with a presentation of the basic game idea by the problem owner to the developers. The aim was to reach a common ground on the project vision and the reasons behind the idea. This initiated a brainstorming about possible solutions. The main challenge was perceived to be the structure recognition. Therefore, we started by exploring possible technical solutions on paper. When considering radio communication and information through light signals, measuring resistances turned out to be the easiest to develop and cheapest alternative. Concerning the algorithm for structure recognition, we realized that it is simpler if each car part is only detecting its nearest neighbor instead of all parts detecting the entire structure. Therefore, resistance identification and neighbor detection were chosen to be pursued. The resistor solution was tested with resistors on a solderless prototyping board measured with ohmmeters. The principle was confirmed as functional. The idea here was to make sure as early, simple and fast as possible that principles worked with components already available in the lab in order to minimize the amount of time wasted in case it did not work. In the beginning of the **second week** of the development journey, we determined that we require three electrical connection points for connecting two car parts. This fit with the already existing BitSnap connectors [5] that had three electric connection points. They used magnets and their own shape to ensure a consistent and non-ambiguous electrical connection. Those BitSnap connectors were designed to be soldered onto a circuit board, and solderless prototyping boards were too large to fit into the car parts. Knowing that the principle worked, we decided that spending time for manufacturing a soldered circuit board version was a safe investment. The result is shown in figure 2 on the left. From this prototype we learned that the BitSnap connectors were mechanically not rigid enough to support the weight of the car parts. Furthermore, the connectors were not symmetrical, meaning that only matching pairs were combinable. This was in conflict to the fundamental idea of allowing any given car piece to connect. Anyhow, the structure recognition technically worked. Therefore, we continued developing the remaining critical functions such as measuring the resistors with a microcontroller, sending this data to a PC and displaying the measurement on a screen. We decided to not bother with improving the connectors at this point in time to save time towards achieving the critical functions of our envisioned game idea. The microcontroller measurement was prototyped using an Arduino Uno because it is easy to develop, immediately accessible in the lab, and already offers the software to display the results of the measurement on a computer screen. After merging the existing development stages and fulfilling the critical functions as early as possible (digitally recognizing a structure of mechanically attached objects, transferring this data to a PC, and displaying it) we could now focus on improving the existing solution. During **week three**, we intended to focus on shrinking the Arduino microcontroller solution to a size that is suitable for embedding in a car part. Light Blue Beans [6] appeared to be a suitable solution that is already available in the lab. They also had the advantage of replacing the wire connection between the Arduino and the PC by wireless Bluetooth communication. One upcoming problem with Light Blue Beans was that they only have two analogue inputs. This lead to the use of shift registers to channel many measurements through few input pins on the microcontroller. The shift register also work in combination with the Arduino Uno and the Arduino was kept because it is more convenient to program.

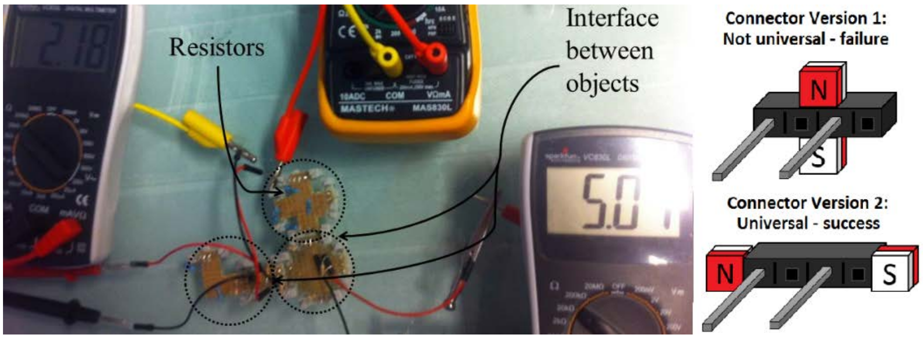

Fig. 2. Left: the resistances on the connected circuit boards in the middle are unique for each connection and measured by ohmmeters. Right: two sequent designs of a mechanically more stable connector.

Week four started with developing mechanically stronger and symmetrical connectors. This was implemented by using larger and stronger magnets and using pin connectors to further stabilize mechanically. The first design is shown in figure 2 in the top right. However, this design turned out to be impossible to connect to an identical connector because the magnet orientation would not match. It required a matching counter piece and was thus no improvement to the previous solution with the BitSnaps. The bottom right design solved this problem. This design flaw was discovered by building and testing the design in a very rough way instead of technically drawing and machine producing the parts. This decreases the risk of design errors and thereby saves more costly resources at a later development stage when such errors have more profound implications.

All electrical components were now on two large breadboards that required a lot of space. The components had to be merged on one platform so that they would all fit safely inside a physical shell. This was accomplished by soldering all components (transistors to control a shift register, reference resistors and header connectors) compactly onto a custom circuit board.

The next issue to be tackled was to advance the virtual representation from a line of text to a car look-a-like representation. We took two approaches into account: The first was a photograph based version where the PC would display a corresponding picture for every possible combination of parts. The second was using 3D models for representation of the car structure. We decided to develop the second option because the number of pictures needed for the first was inconveniently large when scaling up the number of car parts. We used Processing [7] to process the data coming from the Arduino, determining the structure and displaying the models on the PC screen. The system was first tested by displaying a rocket and a chair as substitutes for the virtual car model. Only after verifying the concept, we continued to make virtual representations of the car parts using a CAD software and importing those models to Processing. After confirming that Processing was a reasonable option for displaying a digital representation, **week five** began by drawing the car model parts and implementing them within Processing. At the same time, we also pursued the implementation of Light Blue Beans to make the physical model wireless. However, we experienced that Windows 8.1 did not allow importing serial data via Bluetooth. We could not instantly resolve this problem with the resources at hand and therefore decided to move back to the proven technology to not lose more time with this issue. During **week six** we explored switches, buttons and potentiometers as extra tangible inputs to alter the car parts. Since the gameplay did not yet exist, the visual representation was the only possibility to make adjustments to. We showed that this extension was technically functional and could also be used to change non-visual properties in the game later on. But there was no meaningful reason to develop something further that had no use at the current development stage. Therefore, we stopped after the proof of concept and continued to make a laser cut physical car model in acrylic. The acrylic car model was combined with the existing technology and combined all aspects from physical model to structure recognition and virtual representation. Figure 1 shows the prototype after these six weeks of development.

3 Reflections on the development process

It turned out that our process was very similar to the wayfaring process described by Steinert and Leifer [8]. Both processes are largely based on rapid iteration cycles of design, build and testing ideas as early and quickly as possible. We tested the most critical functions with the resources that are readily available in the lab to fail early and mitigate the risk of losing advances that become unusable due to a later design changes. The early testing lead to learnings that shaped the development journey; the design emerged over time.

4 Detailed description of the latest prototype

The final prototype consists of one central part connected to a PC, and four external objects that can be attached to the central part. The central part has four connectors, one on each vertical side, on which external parts can be attached; each external part has only one connector. When no external parts are attached to the central part, a 3D model resembling the central part is displayed on the connected PC screen. Upon attaching an external part to the central part, a virtual 3D model resembling the attached part is automatically updated.

The identification of the neighboring car parts is achieved by the measurement of resistors through the connectors on the sides of the car parts. All connectors are made from 4 pin headers where two alternating pins are pulled out (see figure 2, bottom right). In the external parts' headers, a resistor is placed with one pin hole between its two legs. In the central part headers, two wires are connected to the female pins that the external connectors will fit into. Thus, when an external part connector is connected to a central part connector, we get a closed loop that runs through one wire into one of the central part header pins, through the male pin on the external part header, through the resistor, and back across to the other wire. This design is made with the intention of having multiple 'central parts' in the future that can measure each other's resistors. So far in this prototype, the central part pins that connect to the external header serve only for structural integrity.

The connector wires are connected to an analogue gate and ground on an Arduino Uno. The Arduino is able to calculate the resistance between ground and the analogue gate by comparing it to a reference resistor between its 5 volt supply and ground. Because there are four connectors and we use only one analogue gate, a shift register is used to control which connector has current at any time. The shift register is placed on a custom made circuit board along with the reference resistor, four transistors, two rows of headers for the connector wires, and a series of headers for easy connection of wires from the Arduino. Three wires connect three digital pins on the Arduino to the shift register. The shift register is connected to the gate pins on the transistors which open the current through the various connectors. Thus, the loop through ground, resistor, and analog gate is controlled. The circuit board is placed inside the central part and connected to the Arduino through a total of six wires (three digital, 5V, ground, and analogue).

When measuring the resistors, the Arduino uses as sequence of North, West, South, and East when the central part is seen from above. For each measurement, the value is serial printed, and a semicolon is added between the values. Processing 2.2.1 imports the string through the COM port on a PC. Before Processing can use the data for anything, it must convert the string into integers and store them in an array. The semicolons act as delimiters for the values. Processing then takes each value in the array and compares it to a set of thresholds.

Processing displays a rotating 3D model resembling the central part in a window. Depending on which interval between thresholds a certain measured value is, Processing displays a corresponding 3D model next to the central part. The correct position is acquired by the position of the value in the array, thus the reason for the compass sequence in the Arduino.

All 3D models are made in Autodesk Inventor [9] and converted into an .obj format. Processing loads the models in the setup of the script, and only displays them when receiving not NULL values from the Arduino.

The physical objects are made from laser cut pieces of 5mm thick acrylic plastic sheets. All pieces are modeled and assembled in Inventor before converting to a 2D format fit for cutting. The pieces are then assembled together with circuit board, central part connectors, and external connector. The pieces are held together with hot glue and clear tape so that broken pieces can be removed and retrofitting is easier.

5 Future plans

In the near future, we will focus on improving the existing prototype by including wireless communication, universally orientable connectors, alternate modes of inter-object communication, more than one ‘smart part’, and how to merge our tangible programming prototype with actual gameplay. We will continue to use a wayfaring mind set as we are satisfied with the results it has yielded so far. Looking further ahead, developing and testing of real gameplay is needed before we can undergo user testing and subsequent reiterations.

Acknowledgments. We extend our most sincere gratitude to project owner Leonore Alexandra Nilsen, CEO of Metis Productions AS, for giving us this challenge and introducing us to the Fibo Car Project. This research is supported by the Research Council of Norway (RCN) through its user-driven research (BIA) funding scheme, project number 236739/O30.

References

1. Web page about “Kerbal Space Program”. Available at: https://kerbalspaceprogram.com/en/?page_id=7 [Retrieved April 28, 2015]
2. “Besiege” sandbox game. Available at: http://www.besiege.spiderlinggames.co.uk/ [Retrieved April 28, 2015]
3. LEGO Mind Storms EV3. Available at: http://www.lego.com/en-/mindstorms/?domainredir=mindstorms.lego.com [Retrieved April 28, 2015]
4. Arduino Uno microcontroller board. Available at: http://www.arduino.cc/en/Main/HomePage [Retrieved April 28, 2015]
5. Little Bits Electronics, BitSnaps. Available at: littlebits.cc/accessories/bitsnaps [Retrieved April 28, 2015]
6. LightBlue Beans. Available at: https://punchthrough.com/bean/ [Retrieved April 28, 2015]
7. Processing programming language. Available at: https://processing.org/ [Retrieved April 28, 2015]
8. Steinert, M., Leifer, L.: ‘Finding One’s Way’: Re-Discovering a Hunter-Gatherer Model based on Wayfaring. Int. J. Eng. Educ. 28, 1, 251-252 (2012).
9. Autodesk Inventor. Available at: http://www.autodesk.com/products/inventor/overview [Retrieved April 28, 2015]