2 Minutes 2 Win It: A Perceptual Computing Game

Job Eliezek N. Bangayan
Ateneo de Manila University
job.bangayan@obf.ateneo.edu

Jan Amiel R.Reveche
Ateneo de Manila University
jan.reveche@obf.ateneo.edu

Paul Vincent M. Virrey
Ateneo de Manila University
paul.virrey@obf.ateneo.edu

ABSTRACT

In this paper, the developers will discuss the developed game, *2 Minutes 2 Win It*. The game uses the Creative* Interactive Gesture Camera coupled with the Intel Perceptual Computing Kit (IPCK). Perceptual Computing deals with natural user interaction, focusing on how humans can interact with their computers using perceptual controls, or things that are natural with the human being. It enables users to use senses – eyes, ears, voice, emotions, etc. – for a more intuitive, immersive, challenging and fun experience. Perceptual Computing aims to make the users feel as if using a computer is somewhat like interacting with a person. IPCK is a collection of developer tools that may help people create applications and systems that perform such things.

The IPCK, as described above, is a collection of developer tools. It contains SDK for face, voice and hand recognition; therefore the developers came up with the idea of the *2 Minutes 2 Win It*, wherein they presented a collection of minigames that demonstrates each tool the IPCK has. As they developed and tested the game, several issues came up, such us unstable voice recognition. They have concluded that although the IPCK is a revolutionary tool that may help future users to improve computing experience, it still has a lot of problems that should be addressed.

General Terms

Design, Experimentation

Keywords

Creative Interactive Gesture Camera, Intel Perceptual Computing Kit, Perceptual Interface.

1. INTRODUCTION

1.1 Perceptual User Interface

Perceptual user interfaces are developed to enable users to interact with the technology as we would with non-technological objects that we use daily. Perceptual user interfaces are also designed to simulate human to human interaction. In a paper written by Matthew Turk, a member of the Microsoft Research Team, he defined perceptual user interfaces as:

> *Highly interactive, multimodal interfaces modeled after natural human-to-human interaction, with the goal of enabling people to interact with technology in a similar fashion to how they interact with each other and with the physical world.[2]*

There are certain characteristics that perceptual user interfaces should possess. Devices and sensors used for the technology should be as transparent and passive as it can, and its machines should be able to take human communication channels and interpret it in a way that the user could still understand. These characteristics are required so that multiple levels of technologies can also be implemented such as speech and sound, recognition and generation, computer vision, graphical animation and visualization, language understanding, touch-based sensing and feedback, learning, user modeling and dialog management.

Two words describe perceptual user interfaces: **interactive**, **multimodal [2]**. Mouse and keyboard events are examples of traditional passive interfaces wherein the computer still waits for the user to enter commands, and only then the computer would do that command. Perceptual interfaces are highly sensitive, meaning they scan for input based on fixed intervals regardless of whether the user wanted to enter a command or not. They are also multimodal because unlike the usual interfaces, where all that you need is your hand to type, scroll and navigate, perceptual interfaces use almost all of the perceptual modalities - sight, hearing, touch - for both the user and the computer. The user communicates to the computer via perception, and vice versa.

The Creative* Interactive Gesture Camera Developer Kit (the "Camera") is a small, light-weight, USB-powered camera optimized for close-range interactivity. Designed for ease of setup and portability, it includes an HD webcam, depth sensor and built-in dual-array microphones for capturing and recognizing voice, gestures and images.

The Camera, when paired with the Intel® Perceptual Computing SDK Beta 2013, enables developers to create the next generation of natural, immersive, innovative software applications that incorporate close-range hand tracking, speech recognition, face analysis and 2D/3D object tracking on Intel Core ™ processor-powered Ultrabook™ devices, laptops and PCs.

The Creative Interactive Gesture Camera is bundled up with Intel's Perceptual Computing SDK Beta 2013, which made the device support perceptual user interfaces. As human beings, we develop natural sensing and perception abilities as we interact with other people.

1.2 SDK: Modules

The IPCK SDK is developed by Intel. The SDK has modules which includes hand and finger tracking, face tracking and analysis, voice module, background removal and raw stream capturing.

The SDK finger tracking module tracks hands and finger locations, and performs pose/gesture recognition. The module produces 4 types of processing results: blob information, geometric node tracking result, pose/gesture notification and alert notification. Blob information is the results of intermediate image processing on depth images. Geometric node tracking returns tracked geometric nodes which refer to the skeleton joints in a human body. The pose/gesture notification matches the recognition results to a set of predefined poses and gestures. The alert notification sends notifications whenever an error occurs. [1]

The SDK face tracking module provides a suite of face tracking algorithms including face detection, landmark detection, and face recognition. The face detection locates the rectangular position of a face or multiple faces in a given view. Landmark detection detects the key features of the face (eyes, nose, mouth, etc.) given in the face rectangle. Face recognition recognizes a face, meaning, the face can be given a specific name if the person is previously registered in a database. [1]

The SDK voice modules provide command and control, dictation, and text to speech capabilities. Command and control has a predefined list of words in the command list and the SDK module recognizes speech based on the list. In dictation, the SDK module returns the most likely dictated sentence. [1]

The SDK background removal module takes color and depth streams as input, and produces a segmented image, which outlines the foreground object(s). The module is also capable of masking background pixels with certain color on an image and produce a blended image. [1]

In the raw stream capturing, the device is capable of capturing raw camera data which includes images, audio and coordinates.

1.3 Goal

The group aims to develop a game that tests and demonstrates the features of the IPCK. The group does this with the following research questions.

- How can we design a game that makes authentic use of the features of the IPCK?
- What are some of the limitations of the IPCK?
- How satisfying is the game experience?

2. GAME CONCEPT

2 Minutes 2 Win It is an action puzzle PC game where the player must try to finish 15 minigames while maintaining his life.

The game features a collection of 15 mini games that requires thinking and fast reaction time. In the game, the player has 3 lives and has to finish as many mini games as possible until all the lives are lost or until the game time of 2 minutes has expired. Each mini game lasts about 7 seconds, with a short explanation of what the player needs to do at the start. The player loses a life when he fails to finish the minigame's goal within the time limit.

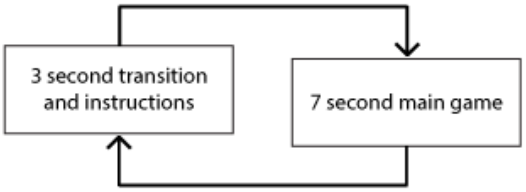


Figure 1: Game Flow

The game is more interactive compared to other games. What makes the game more interactive revolves around the use of the IPCK. With the help of this device, the game can c apture the player's hand gestures, track the player's face and record the player's voice as the main source of input.

The mini games played in the game are independent of each other as each minigame has a goal of its own. The mini games to be played by the player does not follow a sequence and are chosen randomly.

3. LOOK AND FEEL

Playing through the game will have the player be faced with successive minigames. Each minigame have differing settings though the art style will be kept throughout the minigames. At most 5 seconds will be given to the player to finish one minigame, once a minigame is finished a transition screen will be displayed.

3.1 Controls

Since the game aim to exhibit the features and functionalities of Creative and the IPCK, the minigames will be controlled by hand gestures, head movements and voice recognition. The game aims to smoothly incorporate these features within the game and make the controls as intuitive as possible. The game can be compared to other minigame collections such as Nintendo's *WarioWare* Series[4] and Armor Game's *4 Second Frenzy*[5], though *2 Minutes 2 Win It* is heavily gesture controlled through the IPCK.

3.2 Target Audience

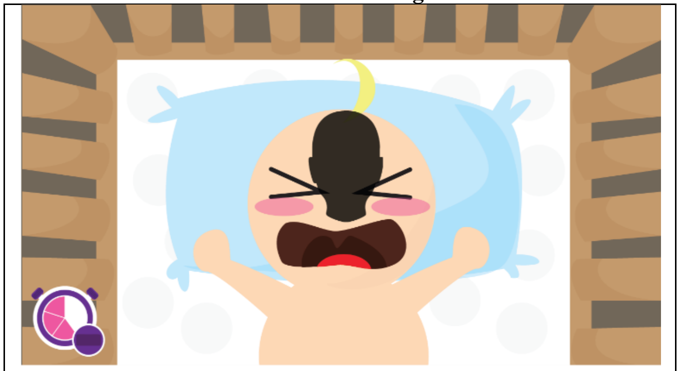
The game is targeted towards the general audience, leaning towards lower age bracket, children going grade school.

3.3 Art Style

Being targeted towards the general audience, the game is designed to accommodate the younger audiences. Cartoonish art style will be used throughout the game and a little splash of comedy and silliness. See the appendix for screenshots.

4. MINIGAMES

Table 1. List of Minigames

 **Figure 2: Peek A Boo**	
Peek A Boo [Modules: Face Tracking]	
Goal:	"Peek a Boo!" - The player must make the crying baby smile
Controls:	The player uses his hands to play peek-a-boo with the child.

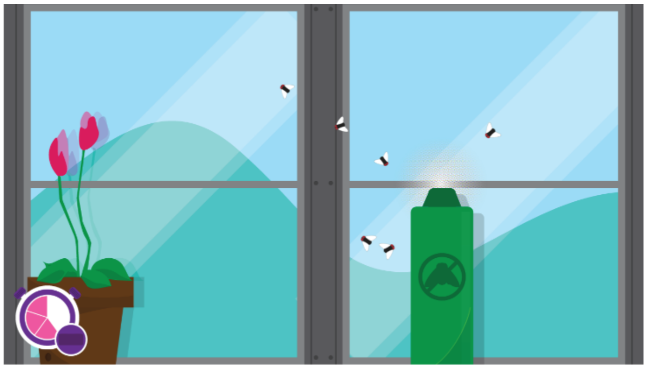


Figure 3: Swat Attack

Swat Attack [Modules: Hand Tracking + Gesture Recognition]	
Goal:	"Kill all the flies!" - The player must kill all of the flies in the screen.
Controls:	The players hand is projected at the screen as a bug spray, the player then gesture as if he/she is spraying.

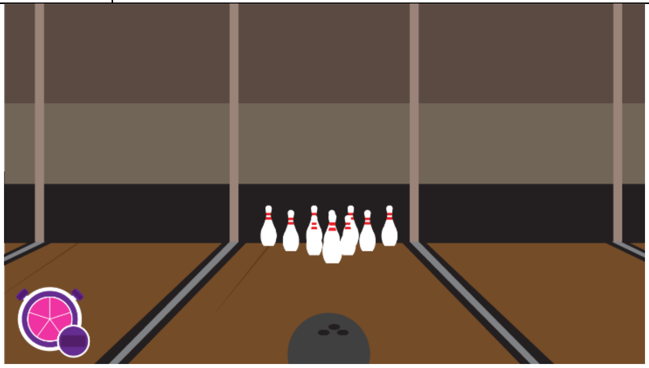


Figure 4: Roll The Ball

Roll The Ball [Modules: Gesture Recognition]	
Goal:	"Knock down the pins!" - The player knocks down the pins with the bowling ball.
Controls:	The player uses his hand that signals the release of the bowling ball in a given direction.

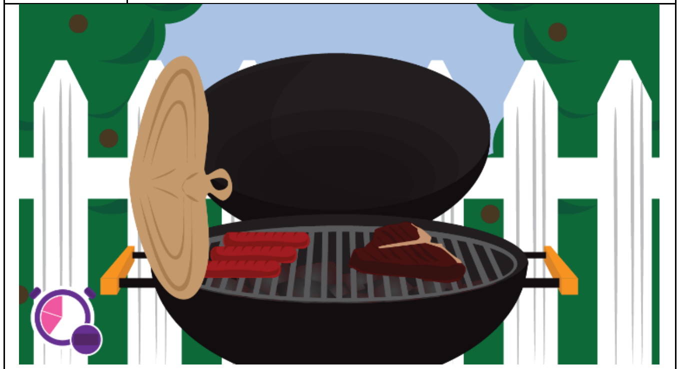


Figure 5: Grill The Meal

Grill the Meal [Modules: Hand Tracking]	
Goal:	"Grill some Barbecue!" - The player must fan the coal to cook some barbeque.
Controls:	The player uses his hand to create gestures like when grilling barbeque.

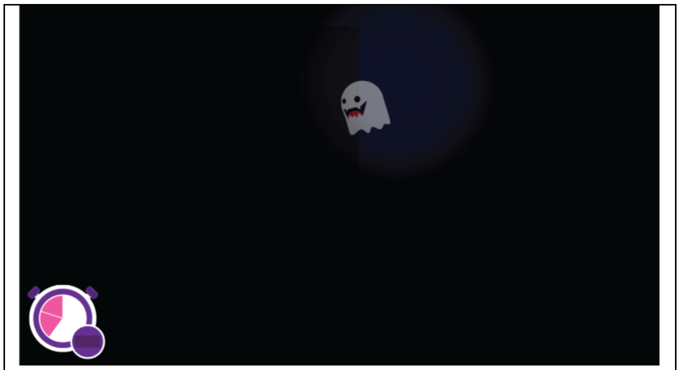


Figure 6: Night Lights

Night Lights [Modules: Hand Tracking]	
Goal:	"Keep up the Light" - Keep the flashlight directed at the ghost for the whole duration.
Controls:	The player points at the screen which is mapped to where the flashlight is directed.

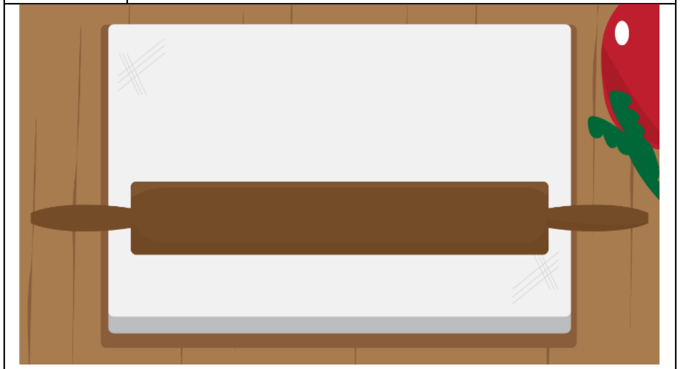


Figure 7: Five Second Diet

Five Second Diet [Modules: Hand Tracking]	
Goal:	"Roll him Up!" - The player must make the person tall and thin by using the rolling pin.
Controls:	The player uses both hands and move them up and down as if using the rolling pin along the screen.

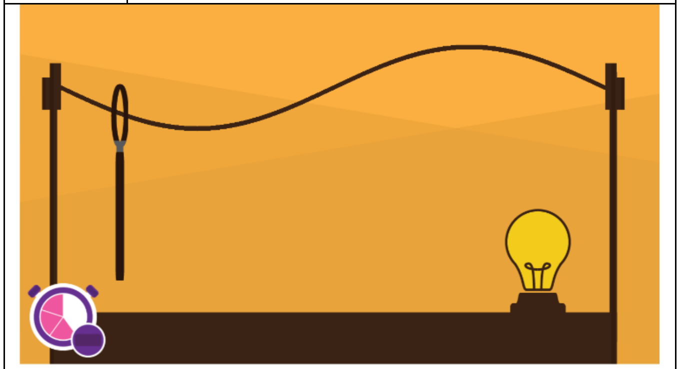


Figure 8: Wire Loop

Wire Loop [Modules: Hand Tracking]	
Goal:	"Must...not...touch" - The player must transfer the metal loop across without touching the curved wire.
Controls:	The player uses his fingers to touch the handle of the metal loop and move it to the other side.

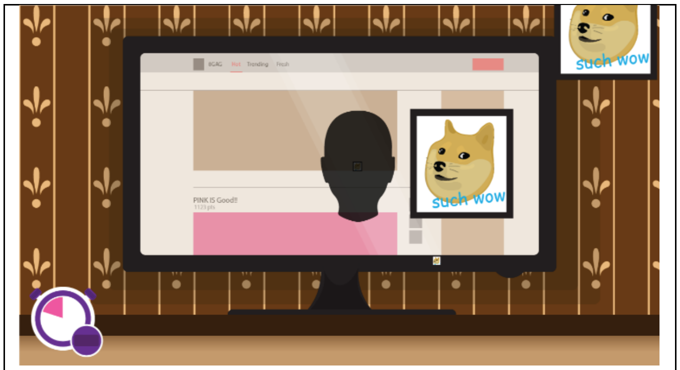


Figure 9: Dodge

Dodge [Modules: Face Tracking]

Goal:	“Wow. Much Agility. Such Speed. Wow” - The player must dodge the obstacles and incoming doges.
Controls:	The player uses his head and body to dodge

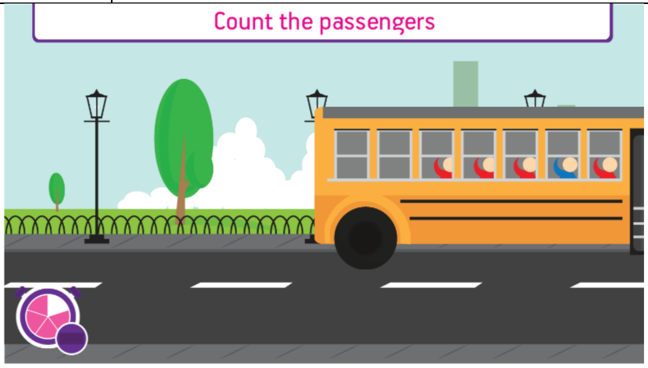


Figure 10: Number the Passenger

Number the Passenger [Modules: Voice Recognition]

Goal:	“Count the People!” - The player must tell the number of people inside the bus.
Controls:	The player uses his voice to tell the number of people of the bus.

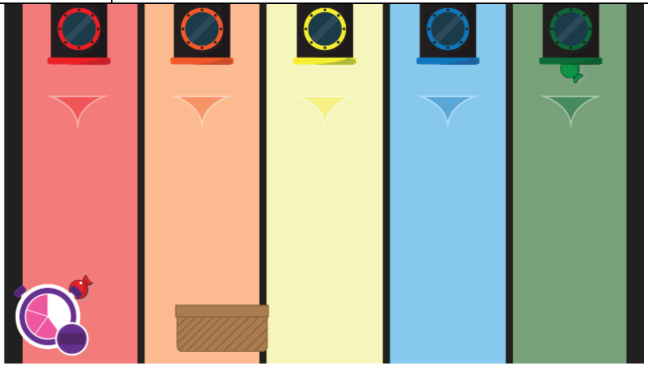


Figure 11: Sweet Treats

Sweet Treats [Modules: Face Tracking]

Goal:	“Catch the Candies!” – The player must catch all falling candies.
Controls:	The player move his head to move the basket sideways along the screen.

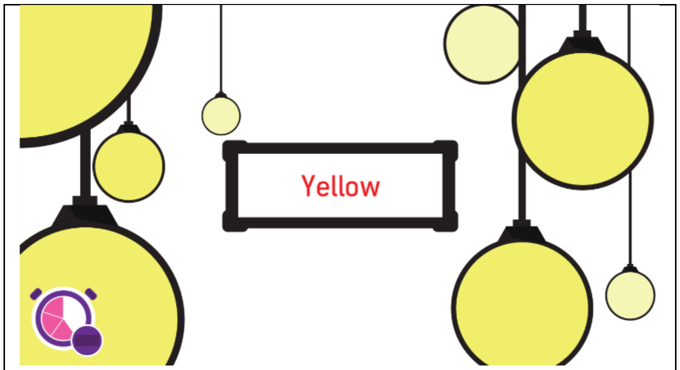


Figure 12: Stroop Effect

Stroop Effect [Modules: Voice Recognition]

Goal:	“Say the COLOR, not the WORD!” - The player must say the color of the text, and not the text itself.
Controls:	The player says the color to move on to the next one.

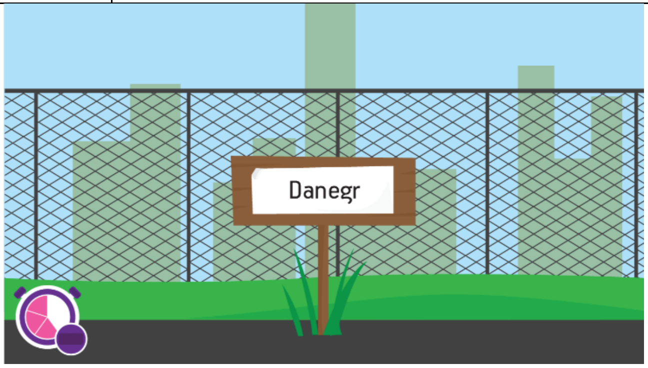


Figure 13: Rumble Jumble

Rumble Jumble [Modules: Voice Recognition]

Goal:	“Guess the Word!” - The player must identify the word given the first and the last letters.
Controls:	The player says the correct word that you will get by rearranging the letters of a given string.

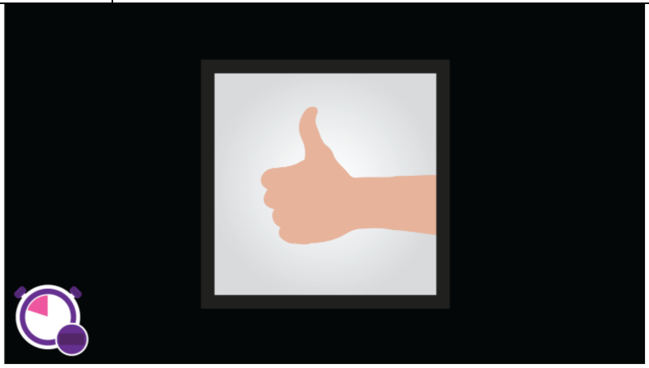


Figure 14: Star Struck

Star Struck [Modules: Gesture Recognition]

Goal:	“Strike the Pose!” - The player must copy the gesture of the person in the screen.
Controls:	The player uses his hands to mimic the pose of the person in the screen.

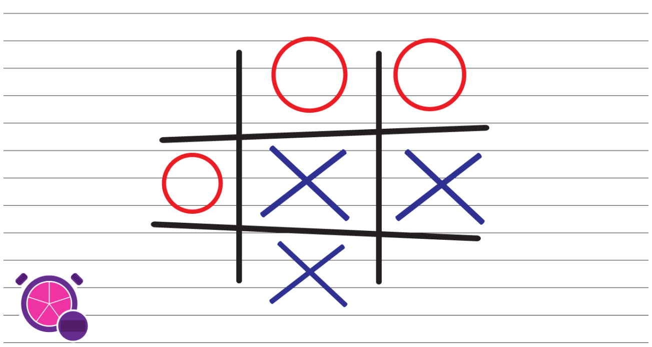


Figure 15: TicToc

TicToc [Modules: Hand Tracking]	
Goal:	"Tic Tac Toe!" - The player must choose where to place the next move in order to win a tic tac toe game.
Controls:	The player hovers his hand over the area where he wants to place his move.

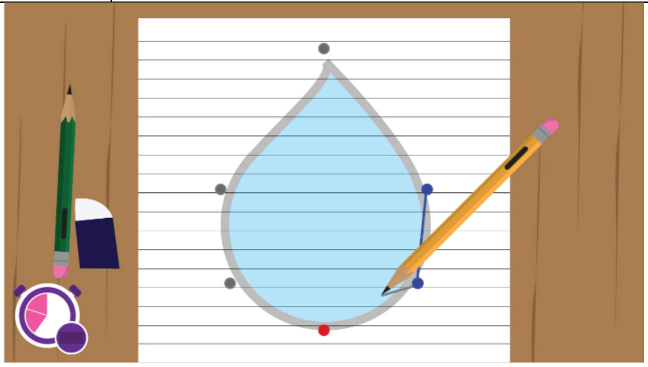


Figure 16: ...Dots...

…Dots… [Modules: Hand Tracking]	
Goal:	"Connect the Dots!" - The player must connects all the dots in order.
Controls:	The player uses his finger to point from one dot to the next.

5. IMPLEMENTATION

In order to make the game possible, the Creative* Interactive Gesture Camera is used as the main device for capturing images, gestures and speech. Using the IPCK SDK, capturing these information are made easy and convenient for the developer. Since the computing kit is in C#, the group will be using Visual Studio as their official IDE while developing the game.

The game also has a lot of illustrations, images and other graphic-related elements, so GIMP and/or Inkscape is also used in developing the game. Sound Effects were altered and edited through Audacity and other sound editing software.

6. TESTING

6.1 Testing Plan

The testing of 2Minutes 2 Win It would be for the purpose of finding out whether the game, using the Interactive Gesture Camera, is intuitive enough to be able to play properly. Since the game controls isn't as simple as a keyboard and a mouse, it would be in their best interest to find out whether the instructions placed all throughout the mini-games would be enough for the testers. The developers do not concern themselves much about finding bugs in the program since the SDK of the camera itself still has some problems and inconsistencies. However, it's still in their interest to fix any bug that they created themselves. And of course, the developers have managed to make the game exciting for its wonderful design and concept, so they would also be observing how much fun the game is for the testers.

The Creative Interactive Gesture Camera is a new technology that might not be that famous yet, however, because of its similarities with the Kinect, a well-known device by Microsoft used by a lot of people nowadays, it would be easier to test the game to people who are familiar with Kinect. The Computer Science Majors in Ateneo de Manila University would be enough to have at least 5 people as testers. It is safe to assume that these people are quite knowledgeable about the Kinect, and that they would be viable testers for the game. However, just in case there would be people in this group that does not know about Interactive Gesture Camera, or even Kinect, the developers would get some of them as testers, so that they could observe how this set of testers cope with the game.

As for the testing proper, the developers requested the tester to play the entire game. Of course, being new to the controls of the Interactive Gesture Camera, testers are not expected to finish the game in their first trial. So, the developers have decided to give each testers 3 trials to finish the game. During the game, a player has 3 lives, and if the player runs out of lives, then that counts as 1 trial. The developers tells the testers to start the game by themselves, giving only the instruction of pointing their finger at the camera. During the first trial, the developers assists the tester by first explaining a little bit of background of the game. After the 1st trial, the developers leave the game with the testers alone. There are instructions per mini-game, but just so the testers would know the feel of the game, the developers demonstrates the game first before a tester's first trial.

After the testing phase, the developers ask the testers to complete a questionnaire about the game experience. The questionnaire asks the testers to rate the game and contains the following questions:

- What gesture-based interfaces have you used?
- How do you find the game? Have you tried other games with the same structure?
- Does the Intel Interactive Gesture Camera fit *2 Minutes 2 Win It*? Rate from 1-10 and explain why.
- If you were the developer of the game, what minigame would you make?

The testers will also rate how true the following statements are from 1 to 5. 5 being extremely and 1 being not at all.

- I felt challenged
- I felt frustrated
- I think the controls are intuitive
- I think there was enough time given
- I think the game had clear instructions
- I think the game had clear objectives

6.2 Test Results

The group decided to test the game with 5 individuals. These individuals are composed of male college students between 17 to 20 years of age. Most of them have tried a gesture-based interface like Kinect, and so they did not find it hard to play the game. As they played the game, all the testers said that they felt challenged, and at the same time, frustrated. Since the group demonstrated the game before letting them test it, the controls of the game became intuitive for them but most of them said that the controls are rather "hard" or "unstable." In fact, all of them said that the games with voice recognition are the hardest, because the game can't register what they are saying while playing the game. The developers have expected this response from the testers because the SDK for voice recognition itself still has some issues. The testers gave a general average of 8.25 out of 10 for the game with regards to its overall integrity and impact.

Presented below are other feedbacks from the questionnaire.

Table 2. Average rating of questionnaire statements

	Rating
I felt challenged	4.2
I felt frustrated	2.4
I think the controls are intuitive	3.8
I think there was enough time given	4.4
I think the game had clear instructions	4.2
I think the game had clear objectives	4.6

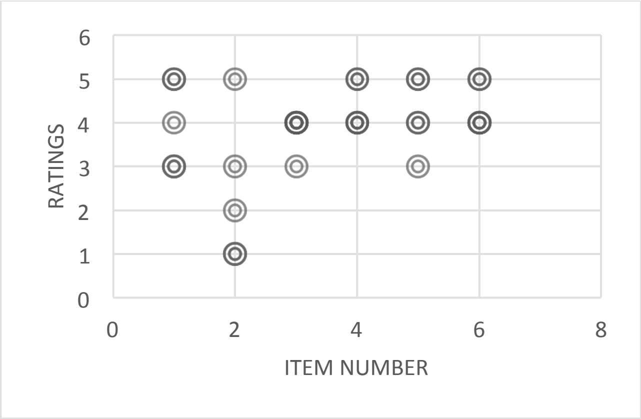


Figure 17: Rating of questionnaire statements

As seen in the table, most testers felt challenged. And despite the minimal time given, they thought it was enough.

When asked what other gesture-based interfaces the testers used, they replied mostly with Kinect, a couple replied none. Most testers found the game fun, interesting and exciting, and haven't heard of *Warioware* nore *Four Second Frenzy*.

The testers believed that the device and the interface fits the game and makes it more interactive, giving it an average rating of 8.6 when asked to rate how the usage of the gesture controls fits the game. Other comments consists mostly of adding more minigames. The testers also pitched some minigame ideas which ranges from a "Cross the road challenge" to "Using the Force" and copying facial expressions.

The developers observed that the testers found hand gestures minigames a lot easier than face tracking and voice recognition ones. This can be due to the fact that most gesture-based interfaces, like the Kinect, emphasizes hand gestures. But it is true that the hand tracking module of the SDK seems to be the most complete. As discussed in section 7.2 the face tracking fails when the face is tilted. Most people tilt there head when moving from side to side, which is the basic premise of the minigames *Dodge* and *Sweet Treats*, making these games specifically difficult. Giving them instructions not to tilt their heads eases the problem but makes the game less intuitive.

7. FINDINGS AND CONCLUSIONS

7.1 Game Design for Controls

When designing a game that uses several complex controls such as hand/face gestures and voice control, making a fast paced game is not recommended. It mostly results to frustration of the players since they need to master the controls and find the game objectives quick.

To design a game that makes authentic use of the features of the IPCK, the controls and features should be implemented intuitively. Try to think of what the players will think the controls will be the moment they were presented with an objective and a graphical view. With this mindset, the developers of the game succeeded in such a way that the players were able to grasp the controls easily. There were times that the players were not able to grasp it mainly due to what they think the limitations of the system would be.

7.2 Difficulties and Limitations Encountered

During the development of the game the SDK was still under construction and, as of the time of this writing, was undergoing some fine tuning from Intel. As such, there were issues encountered within development which should be taken into consideration when planning to use the device.

Tracking. The hand face tracking of the SDK is limited to a 320 x 240 dimension, forcing the developers to scale the measurements by the hardware's screen size. Doing so decreases the accuracy of the measurements and makes the tracking too responsive, making the minigame Wire Loop difficult, see figure 7. Hand tracking also seems to be jittery along the edges, scaling the measurements some more provided a quick fix. For the hand tracking to be effective, the room needed to be well lit, wrong lighting can produce errors in reading hand and the face and can sometimes cause the SDK to see the face as a hand. Face tracking is rendered useless when the face is tilted sideways by a few degrees.

Voice Recognition. The main issue with the voice recognition is its sensitivity to noise. It doesn't work well or not at all when there's too much noise going on. It works best when used within a closed silent room. Another issue is its tendency to confuse words (eg. Blue & Two, Yellow & Zero …) and its failure to pick up some words. The developers change the words used in the game based on how responsive the SDK is.

There are also some components of the SDK presented in the documentation but remains unimplemented, an example of this is the heads rotation.

Intel is still continuously developing the SDK which makes the field an avenue for innovation and development. The said limitations may be addressed and make the development kit more convenient for use.

7.3 Game Experience

Most testers who have played the game expressed satisfaction. Having the controls intuitive and easy to learn makes for a good experience as it immerses the player more in the game and makes the controls transparent. But due to the limitations of the IPCK, there is still much attention given to the controls as such lowering game satisfaction.

8. ACKNOWLEDGEMENT

The group sincerely thanks Ma. Mercedes T. Rodrigo and Jessica O. Sugay for guiding the developers throughout the project. Thanks are also given to the testers of the game as well.

9. REFERENCES

[1] *Intel(R) Perceptual Computing SDK Documentation*. Intel Corporation. Web. 28 Nov 2013. <http://software.intel.com/sites/landingpage/perceptual_computing/documentation/html/>.

[2] Matthew Turk. *Perceptual User Interfaces*. Microsoft Research. Web. <http://www.cs.ucsb.edu/~mturk/pubs/TurkEC-NSFWorshop.pdf>.

[3] Turk, Matthew, Kolsch, Mathias. 2003. *Perceptual Interfaces*. Technical Report. University of California, Santa Barbara. <https://www.cs.ucsb.edu/research/tech_reports/reports/2003-33.pdf>

[4] *Warioware*. 2003. Nintendo. Video game

[5] *4 Second Frenzy*. 2006. Armor Games. Video game