Interactive Learning Media of the Solar System for Children using Kinect

R F Rahmat¹, N Amalia¹, S Faza², M A Muchtar¹, R Budiarto³

¹Department of Information Technology, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia
²Politeknik Negeri Medan, Medan, Indonesia
³Albaha University, Saudi Arabia

*Correspondece Email: romi.fadillah@usu.ac.id

Abstract. Children education adheres to the emphasis of playing while learning or learning while playing. Playing using the body muscles can stimulate the body senses of a child and assist the child in exploring the surroundings. Children discover and learn new things or skills through playing a game. In creating an interactive learning experience to stimulate the brain of a child, a solar system is included in the formal education curriculum packed in an interactive media using Kinect where the children will play an active role in studying the solar system. The Kinect sensor detects 23 joints of a child to be used to move the 3-dimensional astronaut object directly on the body (coordinates mapper) and uses the background removal on the image to separate the background and the child pixels, so the child seems to be beside the planet. Based on the test results using the respondents of 50 children, it was shown that the system effectiveness level for the children is 86%.

1. Introduction

The solar system consists of the Sun, eight official planets, at least three "dwarf planets," more than 130 satellites on the planet, many comets and asteroids, and interplanetary media. The inner solar system contains the Sun, Mercury, Venus, Earth, and Mars. The main asteroid belt is located between the orbits of Mars and Jupiter. The outer planets in the solar system are Jupiter, Saturn, Uranus, and Neptune. The first thing to note is that the solar system is mostly empty space. The planets are atomic compared to the space of the solar system. It is even too large to be measured on a scale in relation to the orbit size.

The golden age is the reason for the importance of early childhood education. Beeker, in his book, classified the golden age as the children in the age of 3-6 years. This group of children usually attends pre-school or kindergarten. In the book, child education must be prepared in a planned and holistic manner so, in the golden age, the development of the children can be complete to develop and showcase various potentials.

An interactive media for the introduction to solar system using Kinect is required to reach the peak point of the children potential in the golden age. The media provides the children with a learning-by-doing method to learn actively.

A research was conducted to create a robot that can be controlled using Kinect sensors so the robot can move according to user movement. The robot can follow the user and calculate the distance between it and the user using the depth value of the coordinates mapping component and the Kinect...
component. The limit ranges of the robot movement are 2-4 meters for forward motion, 1.5-2 meters for stopping and 0.5-1.5 meters for backward movement [1].

Another study was performed to create a virtual fitting room system that can change the texture and color of the users using Kinect 2. The proposed method of the study had been jointly developed focusing on virtual fitting room applications. This research replaced the shirt texture with a new costume texture from an image. This was made using the camera of Microsoft Kinect 2 to obtain the depth image of the scene. In the study, the shirt segmentation was carried out using Kinect depth information to effectively eliminate the background and manually mark some boundaries of the garment and body. The texture coordinates were calculated using the information of the real object coordinates that matches with the pixel coordinates in the image. The system created a realistic perspective by only considering the real world coordinates of X and Y when making the maps. The effect and result look very realistic. The lighting estimation was achieved by equalizing the lighting of the original image. The estimated lighting was used to shade the texture color of the new image in order to acquire more realistic results where some parts of the surface are in shadow while the other parts are in bright [2].

Another study was conducted on Kinect project mapping using the combination of augmented reality and the sensor and projector of the Kinect. In the research, a framework was developed to calculate the calibration data and achieve dynamic mapping using Microsoft Kinect. The ultimate goal was for the graphical applications to be able to utilize data derived from the Kinect-Calibration projector and apply it to a real environment which results in the visualizations of Augmented Reality from the captured scenes in the runtime [3].

A research on virtual fitting room using Kinect sensor was conducted in 2012. The system allowed the users to choose their clothes in the form of a sleeveless dress and save the result. The input image of the system is a dataset consisting of dress images that have been separated from the background. Kinect recorded the user movements and identified the head, shoulder center, shoulder left, shoulder right and hip center. The processed image will be mapped out, so the user seemed to wear a different outfit [4]. Several interactive application has been developed before such as Augmented reality (AR) for Hijaiyah Learning [5], AR for children with special needs [6], activity learning using traditional game in Android [7].

The solar system is a collection of celestial bodies that are bound by gravitational forces to form a system. A system is required to simplify the learning process that can be used as an interactive learning media about the Solar System where children can play a direct role through actions and visuals in real-time (learning by doing).

2. Methodology

The proposed method of this study consists of several processes. The initial step is to connect the system with a Kinect sensor. There are several processes in the system; the first is to detect the presence of a user within the distance of Kinect scope, then transform the image of the user into a skeleton form using SDK function in the Kinect SDK and background removal to detect the joints of the user. After the joints are detected, the GameObject (Astronaut) is mapped (Coordinates Mapper) to the joints, and the motions are adjusted to the joint motion (Body Joint Orientation). The general architecture of this study is shown in Figure 1.
2.1. Research Data

The system collected the data of the joints in the Kinect to use the coordinates mapper. Of all the 25 detected joints, the system selected 23 joints to be applied. The list of joints is shown in Table 1.

| Joint Order | Joint Name         | Joints Type       |
|-------------|--------------------|-------------------|
| 0           | Spine base         | Spine Base        |
| 3           | Head               | Head              |
| 4           | Shoulder left      | Shoulder Left     |
| 5           | Elbow left         | Elbow Left        |
| 6           | Wrist left         | Wrist Left        |
| 7           | Hand left          | Hand Left         |
| 8           | Shoulder right     | Shoulder Right    |
| 9           | Elbow right        | Elbow Right       |
| 10          | Wrist right        | Wrist Right       |
| 11          | Hand right         | Hand Right        |
| 12          | Hip left           | Hip Left          |
| 13          | Knee left          | Knee Left         |
| 14          | Ankle left         | Ankle Left        |
| 15          | Foot left          | Foot Left         |
| 16          | Hip Right          | Hip Right         |
| 17          | Knee right         | Knee Right        |
| 18          | Ankle right        | Ankle Right       |
| 19          | Foot right         | Foot Right        |
| 20          | Spine shoulder     | Spine Shoulder    |
| 21          | Hand tip left      | Hand Tip Left     |
| 22          | Thumb left         | Thumb Left        |
| 23          | Hand tip right     | Hand Tip Right    |
| 24          | Thumb right        | Thumb Right       |
In addition to joints data, the system requires 3D astronaut and planetary objects that are given a texturing image of a planet. Planet texturing aims for a planet to look like its actual state. The image texture of the planetary can be seen in Table 2.

| Object       | Texture |
|--------------|---------|
| Sun          | ![Sun Texture](image) |
| Mercury      | ![Mercury Texture](image) |
| Venus        | ![Venus Texture](image) |
| Earth        | ![Earth Texture](image) |
| Moon         | ![Moon Texture](image) |
| Mars         | ![Mars Texture](image) |
| Jupiter      | ![Jupiter Texture](image) |
| Saturn       | ![Saturn Texture](image) |
| Ring of Saturn | ![Ring of Saturn Texture](image) |
| Neptune      | ![Neptune Texture](image) |
2.2. Detection of User Skeleton
The skeleton in the frame can have a tracking status of "Tracked" or "Position Only." The tracked framework provides detailed information about the position in the camera viewpoint consisting of twenty-five joints of the user's body. A framework with tracking status "Position Only" has information about the user's position, but there are no details about the joint.

![Figure 2. Skeleton and Joints Tracking.](image)

The application can specify the framework to be tracked using the tracking ID as shown in the Active User Tracking section. For the tracked skeletons, a joint array provides the position of twenty-five human joints in space. In an example, the application can use the hand joint to guide the cursor on the screen or pull the position of the user body on the screen. The joint also has a tracking status of "Tracked" for a connection that is visible, "inferred" if the connection is not visible and Kinect will decide its position, and "non-tracked," for the bottom joint when the user is sitting down.

2.3. Background Removal
Background removal is a separation process between user pixels and background pixels. When we wipe out the background, it is necessary to keep the pixels representing the user and remove the other redundant pixels. The depth camera of Kinect sensor is utile in determining the user body. However, instead of the distance, it is compulsory to find the value of RGB. We need to determine the RGB value that matches with the depth value of the user. Using Kinect, every point in the space has the following information: Color value: Red (Red) + Green (Green) + Blue (Blue). Depth Value: Distance from the sensor.

The depth camera provides the depth values, and RGB camera produces the color value. CoordinateMapper is used to map these values. CoordinateMapper is a useful Kinect property that determines which color value suits each distance and vice versa. Note that the RGB frame (1920 × 1080) is wider than the frame depth (512 × 424). As a result, not every color pixel has the appropriate depth mapping. However, body tracking is mainly performed using depth sensors, so there is no need to worry about missing values. Background removal has a result like the green screen.

2.4. Coordinate Mapper and Body Joints Orientation
The Coordinate Mapper is to identify whether the dots of 3D space correspond to 2D color or depth space and vice versa. CoordinateMapper is part of the KinectSensor class. In the Kinect, there is a Camera Space referring to the 3D coordinate system used by Kinect. The coordinate system is defined as follows: Origin (x = 0, y = 0, z = 0) is located in the center of the IR sensor on the Kinect, X grows to the left of the sensor, Y grows up above the sensor (note that this direction is based on the sensor slope), Z grows in the direction of the sensor and 1 unit = 1 meter. Depth Space is a term used to describe 2D locations in depth images. Consider this as the location of row or column of a pixel where x is the column and y is the row. So x = 0, y = 0 corresponds to the top left corner of the image and x = 511, y = 423 (width-1, height-1) is the lower right corner of the image. In some cases, a z value is
needed to map the inner space. In this case, it is sufficient only with the sample of the depth image in
the row or column and uses that value (in millimeters) directly as z.

A common operation in depth image is to produce 3D points at the target point. In this case, it will
not project from the depth space to the camera space. The operation uses one of the array-based
functions (MapDepthPointsToCameraSpace) or the mapping table (GetDepthFrameToCameraSpaceTable) then the multiplication can be performed. After the skeleton
and joints of the user are obtained, the next task is to map 3D objects into the joint coordinates. It is
necessary to divide the 3D objects according to the types of mapped joints. The joints of the user can
be accessed using BodyJoint handleft = body.Joints [JointType.HandLeft], and carried out to each
joint to be accessed.

The Body Joints Orientation is applied for the mapped 3D objects to move according to the type of
the mapped joint. Kinect reads the orientation value of the joints as a quaternion. A quaternion is a set
of 4 values: X, Y, Z, and W. Kinect SDK encapsulate the quaternion into a structure called Vector4.
This Vector4 needs to be converted into a set of 3 numerical values. The rotation of the joints around
the axes X, Y, and Z can be calculated using quaternion orientation. The rotation around the X-axis is
called Pitch, the rotation around the Y-axis is called Yaw, and the rotation around the Z-axis is called
Roll. The body joints orientation can be summoned using the command of BodyJointOrientation
handLeftRot = body.JointsOrientations[JointType.HandLeft].

![Figure 3. Joint Direction.](image)

2.5. Activity Diagram

Figure 4 describes the process of system activity. First, the user runs the application and checks the
connection of the Kinect sensor. If the Kinect is connected, the system displays the user interface, and
If not connected, it will go back to the start screen. After the system connected, it will detect whether
there is any user within the distance range of the Kinect. If no user is detected, the display remains
unchanged. The system detects the user but only when the user raises both hands. These users will be
tracked as the player. Every movement of the detected player will be tracked, and the player's hand
will serve as the cursor to run the system. The cursor will move according to the hand movements, and
the user can change which hand is made for the cursor. The user should put his current hand cursor
behind and put the other hand upfront to change the hand for the cursor.

After the player is detected, the player appears in the system. The user can choose to appear as
himself or an astronaut in 3D while playing in real-time. When a player uses the system, the system
can detect and display up to 6 players on the system screen in background removal mode. Players can
choose the selection of the solar system and planet to learn and see the information of the selected
objected in detail. When a player is in the solar system screen, the player can see how each planet
moves around the sun. The player then can move to the next view, which is the detailed view of the
planet. The user is required to touch his head to see the details of the planet when in astronaut mode.
The users can see the planet and the direction of its rotation up close. They can also see the
information on each planet in the information bar which is presented in the form of text and sound.
When the information bar shows, the voice information will also be heard by the user. Players can exit
the system by selecting the exit button symbolized by an X. However, players can also return to the
previous Solar System view with a menu symbolized by a triangle.
3. Result and Analysis

3.1. System Requirements
Several specifications were required in building an interactive learning system using a Kinect sensor to support the application running smoothly. The minimum requirements are as follows:

1. Processor Intel(R) Core(TM) i3-2348M
2. 64-bit (x64) processor
3. Dual-core 3.1 GHz
4. Memory (RAM) of 4 GB
5. Kinect adapter
6. USB 3.0 host controller
7. The graphical adaptor of DX11:
   - Intel HD 4400 integrated display adapter
   - ATI Radeon HD 5400 series
   - ATI Radeon HD 6570
   - ATI Radeon HD 7800 (256-bit GDDR5 2GB/1000Mhz)
   - NVidia Quadro 600
   - NVidia GeForce GT 640
   - NVidia GeForce GTX 660
   - NVidia Quadro K1000M
8. The operating system of Windows 8, Windows 8.1 or Windows 10
9. Visual Studio 2013 and Unity 5
10. Kinect SDK V 2

3.2. Implementation of System Design

The first page that appears is the opening screen. The camera runs from the direction of the sun to the last planet, Neptune. This view aims to introduce the whole planet. The opening display can be seen in Figure 5.

![Figure 5. Welcome Page.](image)

The second display of this application is to present the images of the orbital movements of the entire planet around the sun or the revolution. This has the purpose of introducing the direction of each planet in the revolution process. Figure 6 shows the orbit of the planet in the three-dimensional model in the astronaut mode, while Figure 7 shows the planet's orbit with the background removal mode or using the image of the player himself. For the background removal mode, the system can display up to 6 people, but only one person is tracked as a pointer.

![Figure 6. Planet Orbit View in Three-Dimensional Object Mode.](image)

![Figure 7. Planet Orbit View in Background Removal Mode.](image)

The next one is a screen containing information about each planet and the sun. Figure 8 shows a planetary information display in three-dimensional Object mode, while Figure 9 presents the display in background removal mode. In this mode, Kinect can detect up to 6 users.
3.3. Application Testing
Authors conducted a questionnaire on 50 respondents of all elementary school students in SDN 105287 with ages ranging from 6 to 9 years to obtain a more accurate result. The respondents were initially guided in the practice of using the system; then they would try the features in the system independently. After the respondents tried all the features available, they were invited to fill out a questionnaire containing a poll on the level of agreement between interaction and gesture. The statements in the poll are as follows:

1. The solar system is easy to use.
2. The solar system is very interactive.
3. Convenience in using the solar system application.
4. The body movements on the screen are synchronized with mine.
5. Able to see the image of oneself on the screen.
6. Able to change the body to an astronaut on the screen.
7. Can use the hand as the pointer.
8. The design of the planets is intriguing.
9. Can remember the planet names in the solar system.

All of these statements are linked with a Likert scale to obtain the results. The scale marks the strongly agreed (SA) points as 5 and the strongly disagreed (SD) as 1. The full marks can be seen in Table 3.

| Table 3. List of questionnaire value |
|--------------------------------------|
| Scale                  | Value |
| Strongly Disagree (SD)     | 1     |
| Disagree (D)              | 2     |
| Neutral (N)               | 3     |
| Agree (A)                 | 4     |
| Strongly Agree (SA)       | 5     |
Based on the values in Table 3, the average value of each question can be calculated. The average value of each statement at the questionnaire for interaction and gesture can be seen in Figure 10.

Figure 10. Graphic of User Satisfaction on Interaction and Gesture.

Furthermore, the system testing was carried out on the user height with a predetermined distance. The test result can be seen in Table 4.

| User Height | Distance from the Kinect | Detection Time |
|-------------|--------------------------|----------------|
| 98 cm       | ± 1 m                    | 2.60 Seconds   |
| 102 cm      | ± 1.15 m                 | 5.5 Seconds    |
| 106 cm      | ± 1.15 m                 | 2.39 Seconds   |
| 111 cm      | ± 1.15 m                 | 1.75 Seconds   |
| 123 cm      | ± 1.5 m                  | 1.05 Seconds   |
| 137 cm      | ± 1.5 m                  | 1.44 Seconds   |
| 142 cm      | ± 1.5 m                  | 3.76 Seconds   |
| 160 cm      | ± 2 m                    | 1.18 Seconds   |
| 166 cm      | ± 2 m                    | 6.21 Seconds   |
| 180 cm      | ± 2.10 m                 | 4.43 Seconds   |
| **Total Time** |                         | **30.31 Seconds** |
| **Average Detection Time** |                     | **3.031 Seconds** |

Based on Table 4, the test was conducted on users with different heights. The players were ranged in age of 6 to 25 years old. Every function in the system performed well. The distance between Kinect and user was adjusted to be able to show the entire body of the user on the screen. If the screen only displayed half of the user body, the leg will fall apart in the astronaut mode. However, it is not a problem in background removal mode. NPS (Net Promoter Score) method was used to measure the level of user satisfaction on the system. The method uses the scale of 0 - 10 where 0 - 6 indicates dissatisfaction and detractors, 7 - 8 represents the passives who are quite satisfied but still indifferent to the system, while 9 - 10 indicates the promoters, users who are satisfied with the system. NPS score can be obtained by subtracting the scores of satisfied users with the unsatisfied ones. Net Promoter Score is always illustrated in absolute number and not a percentage. Net Promoter Score can range
from as low as -100 (if all the users are detractors) to as high as 100 (if all the users are Promoters). The NPS data of this study are shown in Table 5.

| Score | Number of Users |
|-------|-----------------|
| 10    | 7               |
| 9     | 14              |
| 8     | 13              |
| 7     | 11              |
| 6     | 5               |
| 5     | 0               |
| 4     | 0               |

Total User 50

Percentage of the promoters = \( \frac{21}{50} \times 100 \) = 42%

Percentage of the passives = \( \frac{24}{50} \times 100 \) = 48%

Percentage of the detractors = \( \frac{5}{50} \times 100 \) = 10%

NPS Score = Percentage of the promoters − Percentage of the detractors
= 42% − 10%
= 32%

There are many reasons in the necessity to measure the system functions. The most common one is the need to communicate effectively with the interests of the evaluated system. Other is to meet the requirements of comparing the benefits of two or more systems and to identify any malfunction in the system. Of the total 50 users, 43 of them understood the solar system from the system trial. These users answered questions about the solar system correctly. The effectiveness of the system is shown in a percentage using the following equation.

\[
\text{Effectiveness} = \frac{\text{Successful task}}{\text{Total task}} \times 100 \%
\]

\[
= \frac{43}{50} \times 100\% = 86\%
\]

3.4. Kinect Limitations
During the period of building and testing the system, the authors noted some limitations in using Kinect sensors. First is that the Kinect sensor cannot function properly in a room with a full background. Kinect cannot detect users if the background has lots of patterns or other objects. Second, Kinect cannot detect users if the room is dark or in minimum light condition.

4. Conclusion and future research
Based on the test results, some conclusions can be drawn as follows:
1. The system can detect the users with a height of 98 cm - 180 cm in a predetermined distance.
2. Although the system is created for the children, it is also suitable for adults.
3. The system can detect up to 6 people but only those who raise their hands to be tracked and considered as users.

References

[1] Gahlot A, Agarwal P, Agarwal A, Singh V and Gautam A K 2016 Skeleton based human action recognition using Kinect Intl. J. of Computer Applications 0975 – 8887.
[2] Traumann A, Anbarjafari G H and Escalera S 2015 A new retexturing method for a virtual fitting room using Kinect 2 camera CVPR.
[3] Motta T, Loaiza M and Raposo A 2014 Kinect projection mapping SBC J. on Interactive Systems vol. 5 3.
[4] Lan Z 2012 Augmented reality: virtual fitting room using Kinect Singapore: University of Singapore.
[5] Rahmat R F, Akbar F, Syahputra M F, Budiman M A, Hizriadi A 2018 An Interactive Augmented Reality Implementation of Hijaiyah Alphabet for Children Education, Journal of Physics: Conference Series 978(1) 012102
[6] Syaputra M F, Sari PP, Arisandi D, Abdullah D, Napitupulu D, Setiawan M I, Albra W, Asnawi, Andayani U 2018 Implementation of augmented reality to train focus on children with special needs Journal of Physics: Conference Series 978(1) 012109
[7] Rahmat R F, Ramadhan R, Arisandi D, Syahputra M F, Sheta O 2018 Rangku Alu – A Traditional East Nusa Tenggara Game in Android Platform Journal of Physics: Conference Series 978(1) 012103