

# Camera Motion Editor Tool and its Editing Method of CMF and MOT by Using Interactive Ride Game Contents Development

Hyungjin Jeon<sup>1</sup>, Eeljin Chae<sup>2</sup>, Hongsik Pak<sup>2</sup> and Taesoo Yun<sup>2</sup>

*Department of Visual Contents, Graduate School,  
Dongseo University, Busan, South Korea*

*H01086090095@gmail.com<sup>1</sup>, {cinetree, hspak, tsyun}<sup>2</sup>@gdsu.dongseo.ac.kr*

## **Abstract**

*Representation method of interactive ride game contents can be characterized as a synchronize process between the X, Y, Z coordinate value - camera path data of stereophonic imaginary world of which spreading through digital screen, and physical movement of motion chair. This paper introduced a motion frame editor tool and method of interactive ride game contents which allows modifying motion frame, and shows appropriate adaptability on multimedia content developing program. While data being modified by motion frame editing tool inside of UNITY 3D Game Engine. Data can be directly identified as the coordinate changes of X, Y, Z follow by the movement of camera, it is possible to shorten the production period and cut spending. In addition, comparing with using input operation of mouse drawing method during the editing period of a whole project, we hope to present a more convenient editing operation that can help improving interactive ride game contents service.*

**Keywords:** *Interactive Ride Game, UNITY 3D, CMF, MOT*

## **1. Introduction**

Despite recent global economic recession, theme park industry has globally recorded with over 6 percent of growth rate a year between 2004 and 2014. In 2013, the top 25 theme park around the world recorded total 214,708,299 visitors and it is the increased number with 4.3 percent. Including China, Central and South America, overall market of theme park industry seems to be growing. In China, in recent 2 years, 12 theme parks including two as large as Disney land have been opened and there was an announcement about construction of 59 large and medium theme parks. The rides, the main element of theme park industry, have caught huge popularity because they allow virtual experience based on the 3D/4D interactive ride attraction with IT convergence and the software tech inducing interaction with user by perceiving their voices or movements. In these days, our lives are used to digital technology and virtual experience so that the development of theme park industry and all the rides goes with such digitalized virtual reality [1]. The theme park based on interactive content has shown the expansion from 3D/4D interactive theatre and urban theme park business to the interactive ride game content [2]. The interactive content makes the virtual reality through the three-dimensional image and delivers it to the users with software and hardware creating sensory experience for the purpose of strong engagement [3].

The Interactive ride game content is made up of H/W for operating system, S/W and 3D image content. The system consists of major hardware such as digital screen, motion chair, gun controller-type input device, IR LED receiver, network hum system, PC system, and 4D impactor and so on. The technique to create virtual reality in Interactive ride game content has its purpose to make players engaged and immersed in digital screen [4]. The media, used for this purpose, determines the way of perception [5]. As the game

content with camera movement reacting players' eyes and physical movement in motion chair offers strong engagement, the visual sense is the most critical factor to be engaged.

As the degree of reality in playing game depends on interaction process, the players playing ride content are likely to correspond their eyes with the camera and it is visualized on screen through the camera path inside the program for multi-media content development. In addition, the movement and direction of players' eyes is converted to motion file which controls motion chair. The strong engagement based on visual and physical sense works as a catalyst to make players feel as if they exist in the virtual world. As the critical point of making the Interactive ride game content amusing is the dynamic movement visualized with various kinds of vehicles on screen, it is important to create and correct the camera movement path of X, Y, and Z in detail while in design process [6].

This study researches the system that makes it possible to create and correct any correction data by checking the real-time movement of UI screen with the tool available to edit motion frame and also to work those process only with mouse drawing. This system, having simple development procedure as a result of the study, is able to shorten time consumption in content development and to quicken correction of camera direction. Moreover, it offers the camera motion frame editor tool and the way to edit CMF and MOT file by using it. Since it is possible to see the motion frame data with correction, the improvement in camera adjustment and display composition is achieved to save production time and cost.

## 2. Related Works

UNITY3D game engine has become a high practical tool of various game developing process [7]. The Interactive ride game mainly produced into UNITY3D game engine is based on technique for game content development in the way of using the synesthesia with vision from X, Y, and Z camera movement data in UI screen and the secondhand movement effects from synchronizing the camera motion data between motion simulators with a machine able to move along X, Y, and Z axis.

The Interactive ride game content features a displaying method by sync the camera motion data from the movement of UI screen with the movement of motion chair. However, in case of the existing camera input approach such UNITY3D, every value of X, Y, and Z has to be input as a figure by each coordinate and it causes inconvenience [8]. Moreover, whenever correcting, the detailed correction in specific camera path affects the others' and it is hard to check the result of correction on screen, causing inefficiency in terms of time and cost.

Therefore, there is need to present any motion data from X, Y, and Z camera in the form of graphic image and make it able to correct the presented graph or the new one by mouse operation [9]. Furthermore, it is also necessary to study vibration, rolling, angle variation, rotation, camera movement, zooming screen in/out, setting the given speed in each frame and correcting and extra effects in specific section of each axis. Also, there is need for more efficient way to see every data with correction on screen through the tool for motion frame editing and to cut the production time and cost by improving the camera modification and composition process.

## 3. System Construction

### 3.1. Camera Motion Graph Information Creation

Figure 1 is the composition of the camera motion frame editor tool this study suggests. This camera motion frame editor tool is made up of 4 sections; 1) camera motion graph data creation, 2) camera motion graph data correction, 3) camera motion graph data saving, and 4) camera motion graph data variation. The output of HMLP network with

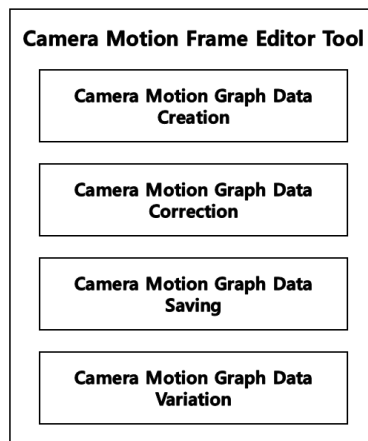
one hidden layer can be expressed by the following equation:

Among the camera motion frame editor tools in Figure 1, Figure 2 is the UI screen with the motion frame and the scene for Chair Game Object, the targeted material of camera motion graph data creation. The camera motion graph data creation section collects the rotation value of motion frame referring to the relative value of camera X, Y, and Z so as to edit the rotation value for the axis of camera X, Y, and Z while playing. In this process, the way to collect the relative rotation value of camera is to subtract the original angle from the current angle and it becomes the data for camera motion graph just like “Vector3 angle = Vector3 (CurmainCamera-OriginMainCamera);” [10].

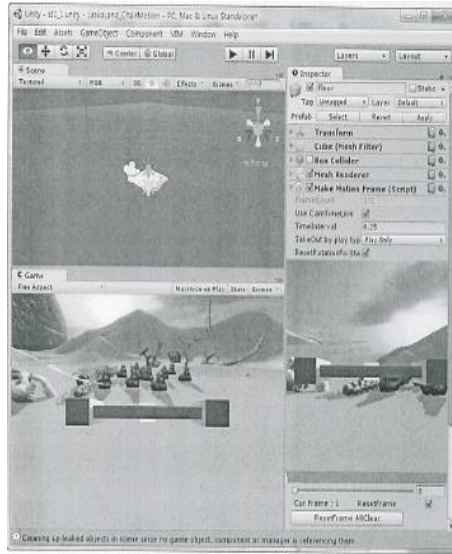
The camera motion graph data creation section makes an assumption that there is no rotation of the axis Y in chair game object. Then it replaces the camera axis Y with the axis Z in motion frame, converting the rotation of the axis Z to create motion frame data of camera axis Y. In this conversion, the function is referred as “Vector3 angle = Vector3 (CurmainCamera-OriginMainCamera);”

Stating in a brief way, in converting the axis Z rotation against the camera axis Y, its value is found by comparing angles between the current camera axis Y and the original one with the function: “floatZangle = Quaternion.Angle (CurMainCamera.Y,OriginMainCamera.Y);” and then presented in quaternion. The value from each axis in automatic method is saved as CMF file and <Table 1> shows its structure.

When in the interactive ride game content, it is able to see the movements of motion frame based on the data from CMF file. Generating the paused frame, it refers the possible rotation value of each axis when playing, compares it with the current set value of camera, then saves that relative value as CMF file.



**Figure 1. Composition of Camera Motion Frame Editor Tool**



**Figure 2. UI Screen with the Motion Frame**

**Table 1. CMF File Saving Structure**

- float timeInterval <sup>↵</sup>
- int frameCount <sup>↵</sup>
↵
- int _resetRotationForStart : 1 or 0 <sup>↵</sup>
- [LOOP frameCount] <sup>↵</sup>
float pos_x, pos_y, pos_z <sup>↵</sup>
float rot_x, rot_y, rot_z <sup>↵</sup>
int reset <sup>↵</sup>
float rot_interval_x, rot_interval_y, rot_interval_z <sup>↵</sup>
float up_down_pos <sup>↵</sup>
float SmallVibPos <sup>↵</sup>
float play_time_point <sup>↵</sup>
- [LOOP end] <sup>↵</sup>

**Table 2. MOT File Structure**

- float timeInterval <sup>↵</sup>
- int frameCount <sup>↵</sup>
- int _resetRotationForStart : 1 or 0 <sup>↵</sup>
- [LOOP frameCount] <sup>↵</sup>
float leg1_y, leg2_y, leg3_y <sup>↵</sup>
Float norTimeLine <sup>↵</sup>
- [LOPP end] <sup>↵</sup>

### 3.2. Camera Motion Graph Information Modification

Figure 3 is the UI screen by the camera motion graph data correction section. The camera motion graph data correction section takes charge of correcting and editing the motion frame in the generated CMF file and creates the final motion file, MOT (MOTION data) file. The structure of MOT file is presented in Table 2.

The rotation graph on axis for MOT file creation has three types; the axis X rotation, the axis Z rotation and the frame Up and Down. The axis Y rotation is not available. It applies micro-vibration to the whole graph in random and uses the data of up and down movements in motion frame. It edits the graph of axis X by applying “rot\_interval\_x”, the rotation value of axis X in motion frame, to the rotation graph of X axis in the CMF file

from the Table 1 structure. In case of graph edition, likewise, the rotation value of motion frame axis Z from CMF file is applied to perform the edition. In the motion frame up and down, “up\_down\_pos” value is applied to edit

By editing the axis X, Y and Up/Down of motion data, the camera motion graph data correction section applies the final rotation value and location variation, calculated by Euler operator such as the first function in Table 3, to perform the motion frame edition as a final output of each frame. As shown in the followed procedure, the data in CMF file offers critical information to form the graph editor. When in the editing process, the correction work is processed by mouse drawing to establish easier and quicker working condition.

### 3.3 Camera Motion Graph Information Saving

The camera motion graph data saving section accounts for saving the current editing data, creating and saving motion data. In saving the current editing data, it saves the data corrected by the correction section as CMF file. In case of motion data, it does in MOT file referring to CMF data.

Figure 4 is the screen shot when the saving section is collecting motion ride data within motion frame. As the height of Leg1, Leg2 and Leg3 box by rotating of motion frame is used for data of each axis in motion chair, this value is saved as MOT file data. The saving section, at first, collects the axis Y value to each Leg, the perceived object by motion frame, and then uses the function in Table 4 to get the initial value of Y in motion frame and the relative location value of every frame.



Figure 3. UI Screen by the Camera Motion Graph Data Correction

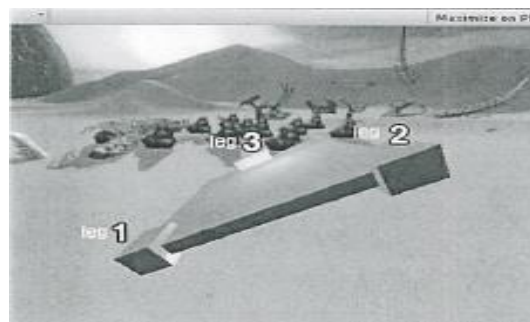


Figure 4. Screen Shot of Motion Ride Section Data Collection

Table 3. Functions of Motion Graph Data Correction

```

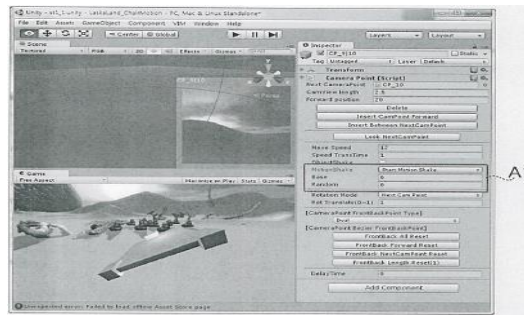
1st Function:
rot_floor = Quaternion.Euler(rot_interval_x, rot_interval_y, rot_interval_z);
2nd Function:
transform.position += updown;
    
```

**Table 4. Functions of Motion Frame Value Creation**

```
Float leg1 = ( floor_leg1.transform.osition.y )-parent_YPOS;
Float leg2 = ( floor_leg2.transform.osition.y )-parent_YPOS;
Float leg3 = ( floor_leg3.transform.osition.y )-parent_YPOS;
```

### 3.4 Camera Motion Graph Information Variation

As the Motion Rider needs the real-time micro vibration, the camera motion graph data variation section finds the function for micro vibration from the camera pointer and then adds or subtracts micro vibration value from the current value of each axis in Rider. The function is divided into two cases, the beginning and the pause of micro vibration, and each function is presented in Table 5. Figure 5 shows the UI screen by the variation section and AI part is the setting screen to find each function.



**Figure 5. UI Screen by the Variation Section and AI Part**

**Table 5. Functions of Micro Vibration**

```
3rd Function:
small_vibration_start(shake_bas,shake_rnd); // Micro
vibration starts
|
4th Function:
small_vibration_enable( 0 ); // Micro vibration stops
```

## 4. CMF and MOT File Editing

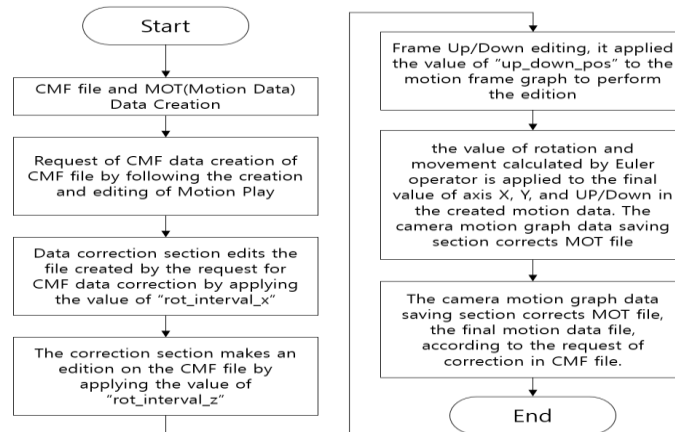
Figure 6 is the flow chart, showing how to edit CMF and MOT files by using the camera motion frame editor tool for multi-media content development program. In a brief description, the camera motion graph data creation section creates CMF file. That is, while playing, it gets the rotation value of motion frame with the relative rotation value of camera X, Y, and Z. In this process, it draws the graph with the rotation value from subtraction between current camera angle and original one. Also, in case of the chair game object, it replaces the axis Y with the axis Z in motion frame and creates graph data after rotating and converting. At last, it saves the final value of each axis as CMF file.

As the next stage, the camera motion graph data correction section edits the file created by the request for CMF data correction by applying the value of “rot\_interval\_x”, which presents the rotation value of axis X in motion frame, as a rotation graph of axis X. In other words, the correction section performs an edition of the axis X graph for creating MOT file, the final motion file, according to the request of correction and edition the motion frame from the CMF file by the creation section.

The correction section makes an edition on the CMF file by applying the value of “rot\_interval\_z”, the rotation value of axis Z in motion frame, as the axis Z rotation graph data. As for the frame Up/Down editing, it applied the value of “up\_down\_pos” to the motion frame graph to perform the edition. In drawing the graph with micro-vibration as

random value, the data of movement up and down against motion frame is available.

At the last stage of motion frame edition, the value of rotation and movement calculated by Euler operator is applied to the final value of axis X, Y, and UP/Down in the created motion data. The camera motion graph data saving section corrects MOT file, the final motion data file, according to the request of correction in CMF file. In other words, the saving section accounts for the function of saving the current editing data and creating and saving motion data and also saves MOT file based on CMF data.



**Figure 6. Flow Chart of Editing CMF and MOT Files**

## 5. Conclusion

This study suggest the motion frame creation tool in the development program for multi-media content including the Interactive ride game content and the system with efficiency and convenience in creating CMF and MOT file using that tool. This system has a composition that makes it possible to see every movement of camera X, Y, and Z as coordinate when designing content such as the Interactive ride game and also edit by mouse drawing with ease. It suggests an advantage of improvement in overall quality of multi-media content due to the simplified developing procedure, the shorten time consumption and quicken correction process for camera direction [11].

It would be significant contribution to cut time and cost when updating game content having demand in the urban theme park and the interactive theatre. Above all, it gives an implication that the convenient condition of creating and modifying the motion frame with the variation value of axis X, Y and Up/Down makes players far more engaged in the Interactive ride game content. In the following study, there would be the research for the control communication system of such camera motion frame so as to achieve far better quality in engagement when playing multi-media content.

## Acknowledgements

This work was supported by the Brain Busan 21 Project in 2015. The authors would like to acknowledge Dongseo Univ. for supporting this research through DSU project.

## References

- [1] J. Robinett, "2013 Theme Park Index Museum Index", Themed Entertainment Association (TEA), (2011).
- [2] S. Kwon, G. Kim, S. Chung, S. Ko and G. Lee, "The Analysis of Physical Effects on Presence in the 4D Attraction Film using the Data Mining", International Journal of Software Engineering and Its Applications. vol. 9, no. 2, (2015).

- [3] E. Chae, C. Choi, K. Choi and K. Kim, "Developing a Sensory Ride Film 'Dragon Dungeon Racing'", Journal of the Korea Contents Association, vol. 11, no. 2, (2011).
- [4] J. Kim and S. Nam, "A study of Immersive Game Contents System Design and Modeling for Virtual Reality technology", International Journal of Control and Automation, vol. 7, no. 10, (2014).
- [5] E. Shin and H. Yoo, "Study on Immersive Installation that Use Interactive Media Technology", International Journal of Multimedia & Ubiquitous Engineering, vol. 10, no. 2, (2015).
- [6] AECOM, "China Theme Park Pipeline Report 2013", Inpark Magazine, (2013).
- [7] J. Bae and A. Kim, "Design and Development of Unity3D Game Engine-based Smart SNG (Social Network Game)", International Journal of Multimedia & Ubiquitous Engineering, vol. 9, no. 8, (2014).
- [8] Y. Xu, E. Kim, K. Lee, J. Ki and B. Lee, "FDS Simulation High Rise Building Model for Unity 3D Game Engine", International Journal of Smart Home, vol. 7, no. 5, (2013).
- [9] H. Jeon, E. Chae, H. Pak and T. Yoon, "Study of Camera Motion Frame Creation Tool and CMF File Creation for Interactive Ride Game Content", International Journal of Applied Engineering Research, vol. 9, no. 24, (2014).
- [10] H. Jeon, E. Chae and H. Pak, "Application of Camera Motion Frame Editor Tool for Unity 3D Game Engine", Advanced Science and Technology Letters, vol. 87, (2015).
- [11] D. G. Hawkins, "Virtual Reality and Passive Simulators: The future of Fun", Communication in the age of virtual reality, New Jersey (1995)

## Authors



**Hyungjin Jeon**, BFA & MFA of Photography from Chung-ang Univ. Seoul, Korea (2009, 2012) Ph.D. Course at Department of Visual Contents, Graduate School, Dongseo Univ. Busan, Korea (2013~) Research interests: Digital Media, Interactive Contents, Virtual Reality.



**Eeljin Chae**, BFA of Photography from Chung-ang Univ. Seoul, Korea (1998) MFA of Motion picture from Academy of Arts Univ. USA (2001) Producer for TUBE PICTURES, FX Digital. (2001~2004) Associate Professor at Division Research interests: Digital Media, VFX, Contents Business.



**Hongsik Pak**, MFA of Computer arts from Academy of Arts Univ. USA (2000) Associate Professor at Division of Digital Contents, Dongseo Univ. Busan, Korea (2010~) Research interests: Interactive Contents, Game Design.



**Taesoo Yun**, PhD of Computer Engineering from Kyungpook National University Daegu, Korea (2001) Researcher for DAEWOO Electronics R&D Center (1993~1998) Full Professor at Division of Digital Contents, Dongseo Univ. Busan, Korea (2001~) Research interests: Game Development, A.I.