

REALITATEA VIRTUALĂ PENTRU VIZUALIZAREA UNEI CELULE ROBOTIZATE

THE USE OF VIRTUAL REALITY IN VISUALIZING A ROBOT CELL

NEAGU Mihai

Facultatea: I.M.S.T., Specializarea: M.I.I.V., Anul de studii: 1, e-mail: mihai.nm@yahoo.com

Conducător științific: Prof. Dr. Ing. **Diana POPESCU**

SUMMARY: The study aims to provide a view of a robotic cell in a virtual reality environment (VR) using an application developed in Unity. Cell view in VR offers a wider range of interactions with the objects present in the cell compared to their presentation in a classic way as an image or 3D virtual environment. The advantage can include simply observing the models inside a 3D environment allowing 360 degrees movement around the model for a better understanding of available space and dimensions. Also, by rendering the cell in VR, the user can take a closer look to the interactions between the elements of the cell, which, in addition to being able to move around the objects, can lead to a better learning experience.

KEY WORDS: Virtual Reality, Robotic Cell, Unity.

1.Introducere

Prin convertirea unei celule robotizate din format 3D CAD într-un format .apk suportat de telefoanele mobile, se dorește punerea la dispoziția unui public mai larg a informațiilor legate de aceste celule. Astfel, nu mai este necesar ca utilizatorul să folosească un program dedicat de modelare asistată pe calculator pentru a putea vizualiza și interacționa cu elementele prezente în cadrul celulei. În schimb, poate opta pentru varianta folosirii realității virtuale (RV) pentru a vizualiza celula la 360°, reducând timpul și cerințele minime de calcul și hardware pentru a putea analiza vizual o astfel de celulă. Pentru a realiza acest lucru, s-a folosit programul Unity prin intermediul căruia s-a realizat celula în mediul virtual, precum și aplicația pentru telefonul mobil.

Avantajele pe care le oferă această abordare sunt posibilitatea vizualizării la scara 1:1 a întregii celule, posibilitatea utilizatorului de a se deplasa virtual în cadrul acesteia și de a observa interacțiunile elementelor precum și facilitarea modului de a accesa astfel de celule și sisteme tehnice de către persoane ce nu dețin cunoștințe în utilizarea programelor de modelare asistată pe calculator.

Pentru a realiza proiectul, au fost preluate modele 3D puse la dispoziție de firma ABB Robotics pentru modelele 3D ale robotului, sistemului periroboți, centrul de mentenanță al torței de sudare și unitățile de procesare ale robotului.

2.Stadiul actual

În momentul actual, pentru realizarea și vizualizarea celulelor robotizate, cât și a altor sisteme tehnice, se folosesc programe dedicate de tip 3D CAD. Aceste programe permit proiectarea și dezvoltarea sistemelor tehnice, însă datorită formatelor în care pot salva aceste

programe este nevoie de programul nativ în care s-a dezvoltat celula, un program similar ce poate accesa același tip de fișier sau un model în format neutru pentru a putea deschide și vizualiza produsul.

În plus, utilizatorul este nevoit să dețină cunoștințe minime în astfel de programe pentru a putea deschide și vizualiza un produs. De aceea, prin intermediul programului Unity, se dorește ca după realizarea proiectului într-un program 3D CAD, acesta să fie convertit într-un format accesibil telefoanelor mobile astfel eliminându-se necesitatea unui program 3D CAD pentru deschiderea proiectului, cât și nevoia de cunoștințe minime necesare. Astfel, proiectul poate fi accesat de către un număr mai mare de persoane; de asemenea modul de prezentare al proiectului pentru persoanelor nespecializate în inginerie este mai accesibil, detaliile fiind mai ușor de explicat și de înțeles într-un mediu VR.

3.Dezvoltarea aplicației

3.1Elemente necesare

Elemente necesar a fi preinstalate pentru a putea dezvolta aplicații pentru Google Cardboard sau Android sunt următoarele:

- Kit-ul Android.sdk ce permite dezvoltarea aplicațiilor pentru platforma Android.

Acest kit se poate descărca de pe site-ul pus la dispoziție de Google pentru dezvoltare: <https://developer.android.com/studio> . De pe acest site se poate descărca programul Android Studio cu care se descarcă toate fișierele adiționale Android.sdk .

- După instalarea SDK-ului trebuie ales fișierul din Unity. Pentru acest lucru se urmează calea EDIT-PREFERENCES-EXTERNAL TOOLS.

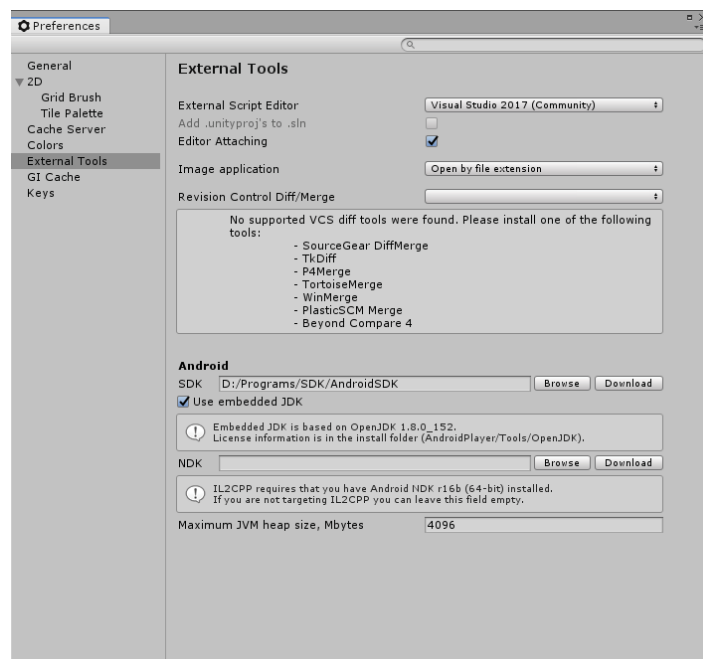


Fig. 1. Inserare SDK

- Următorul pas constă în a debloca Developer Mode pe telefon. Acest lucru diferă de la telefon la telefon, dar în general se poate debloca prin apăsarea de 5 ori a versiunii de telefon din meniul setări (Settings). Odată deblocat Developer Mode trebuie activat USB Debugging.

După ce toate aceste lucruri au fost instalate, se poate lansa Unity.

3.2 Dezvoltarea aplicației în Unity

După crearea proiectului, putem modifica setările acestuia. În primul rând, trebuie schimbată platforma pentru care se dezvoltă aplicația. Implicit, proiectul a fost creat pentru platforma PC, Mac&Linux Standalone. Pentru a schimba platforma, din tab-ul File se selectează Build Settings. De aici, se poate selecta platforma dorită, în acest caz Android.

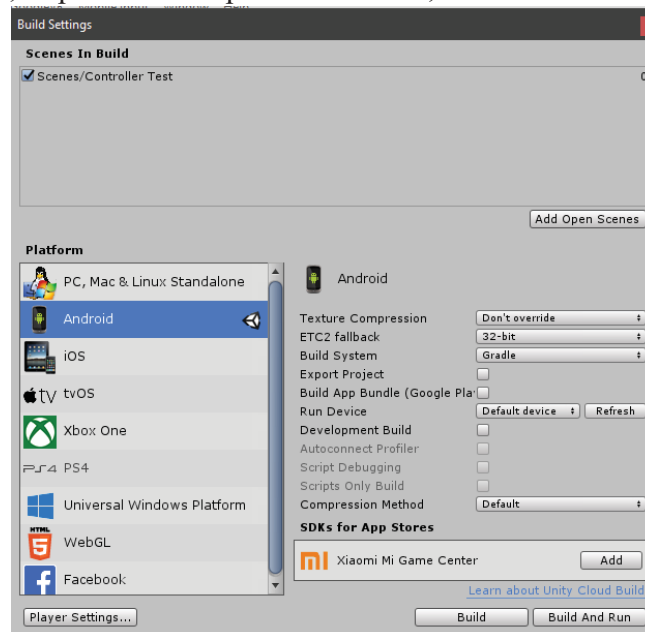


Fig. 2. Selectarea Platformei

Din tab-ul Player Settings se pot modifica setări adiționale. Acestea constau în setări precum Company Name și Product Name cae trebuie completate obligatoriu pentru a putea continua dezvoltarea aplicației. Prin aceste setări, Unity identifică aplicațiile create și se poate merge mai departe la compilarea acestora.

În meniul Other Settings la tab-ul Identification trebuie setat Package Name. Acesta are o formă fixă: com.CompanyName.ProductName. În cazul de față, Package Name va fi: com.UPB.CelulaTest. De asemenea, din acest meniu se poate seta pe ce sistem Android minim va rula aplicația. În acest caz, 4.4 KitKat deoarece este primul sistem Android dezvoltat cu suport pentru Google Cardboard.

Primul lucru creat a fost un plan 3D ce are rolul podea și pe care vor fi amplasate obiectele. Pentru aceasta se urmează path-ul Game Object – 3D – Plane.

REALITATEA VIRTUALĂ PENTRU VIZUALIZAREA UNEI CELULE ROBOTIZATE

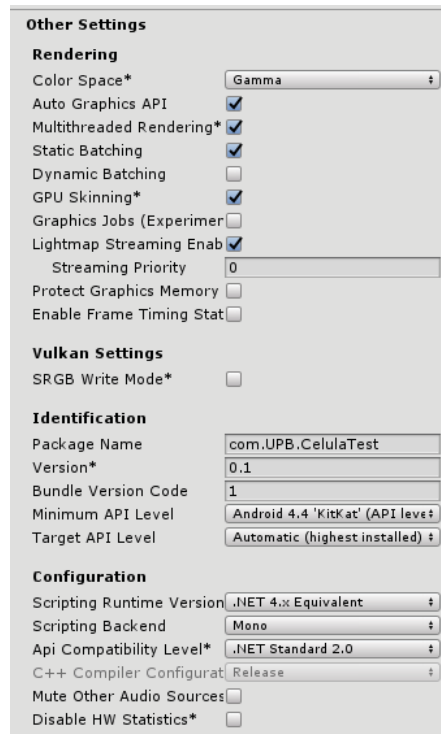


Fig. 3. Setări Adiționale

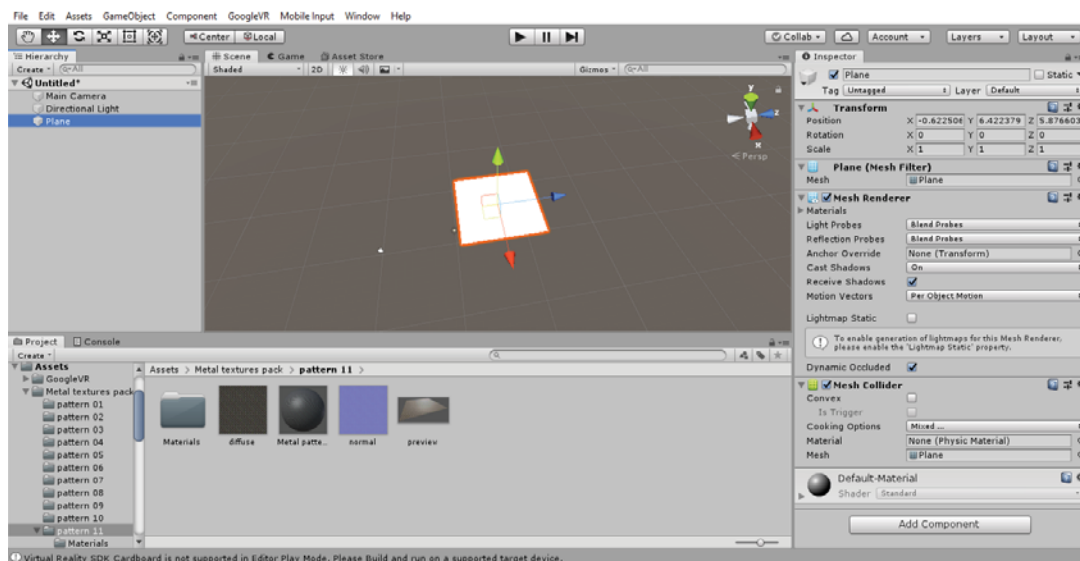


Fig. 4. Interfața Unity

Cu planul selectat din meniul din stânga, în meniul “Inspector” din dreapta se pot modifica anumite setări ale acestuia. De aici s-au modificat valorile pentru pozițiile pe axele X,Y,Z la 0 și valorile pentru Scale la 5 pentru axele X,Y,Z.

Pentru a avea acces la Standard Assets oferite gratis de către Unity, acestea trebuie coborâte din Store.

Pentru acest lucru se merge pe tab-ul Asset Store unde se caută și se coboară pack-ul Standard Assets. Acest pack oferă prefab.-uri simple gratuit.

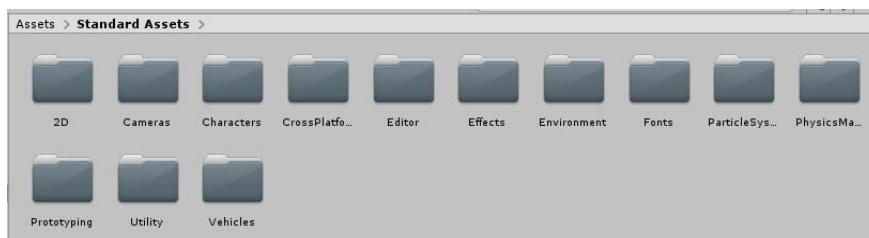


Fig. 5. Librăria de prefabricate

De aici, se va folosi un prefabricat numit FPSController. Acest prefabricat este un ansamblu format dintr-o cameră, un audio source ce oferă sunetul pașilor, un modifier Rigidbody ce permite obiectului să fie afectat de gravitație și un script ce conține maparea controalelor de mișcare, cât și a camerei.

Pentru a putea controla caracterul cu ajutorul Controllerului, este necesară configurarea acestuia. Pentru a seta controalele de Input, se deschide path-ul: Edit – Project Settings - Input. Aici se pot găsi setările pentru axele verticale și orizontale. Se poate observa că sunt 2 seturi de axe Vertical/Horizontal. Primele sunt setate ca și Key and Mouse Button, fiind atribuite controalelor de pe mouse și tastatură. Al doilea set de axe se poate edita și se poate modifica tipul axei pe Joystick Axes atât pentru Vertical, cât și pentru Horizontal. După, putem selecta ce axe disponibile de pe controller putem alocă axelor vertical/orizontal.

În acest proiect s-a folosit un Controller DualShock 4. Datorită suportului redus pentru acest controller (lipsă drivere etc.) s-a mapat axa verticală pe axa verticală a stick-ului din stânga și axa orizontală pe axa orizontală a stick-ului din dreapta. Din cauza lipsei de drivere dedicate, nu s-a reușit maparea ambelor axe pe același stick.

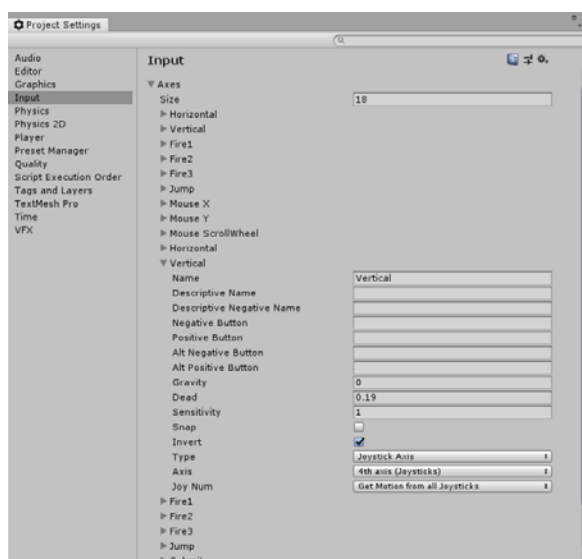


Fig. 6. Setări Input

De asemenea, este nevoie a se modifica programul pentru FPSController pentru a primi input de la axele controllerului. Pentru acest lucru, se dă dublu click pe FirstPersonController în tab-ul Scrip. Acest lucru va lansa Microsoft Visual pentru a putea edita codul.

```

203
204     private void GetInput(out float speed)
205     {
206         // Read input
207         float horizontal = Input.GetAxis("Horizontal");
208         float vertical = Input.GetAxis("Vertical");
209
210         bool waswalking = m_IsWalking;
211
212     #if !MOBILE_INPUT
213         // On standalone builds, walk/run speed is modified by a key press.
214         // keep track of whether or not the character is walking or running
215         m_IsWalking = !Input.GetKey(KeyCode.LeftShift);
216     #endif

```

Fig. 7. Codul FPSController

În program, la secțiunea //Read Input trebuie modificate comenzile în Input.GetAxis pentru a putea primi input de la axele Controller-ului. Trebuie modificate atât axa orizontală cât și cea verticală. De asemenea, pentru a putea deplasa caracterul înainte folosind Cardboard-ul pe post de cameră trebuie modificate setările camerei. În secțiunea FixUpdate() trebuie modificată linia:

Vector3 desiredMove = transform.forward*m_Input.y + transform.right*m_Input.x;

în

Vector3 desiredMove= Camera.main.transform.forward*m_Input.y + transform.right*m_Input.x;

Astfel, axele de control ale caracterului sunt actualizate în funcție de poziția camerei.

```

private void FixedUpdate()
{
    float speed;
    GetInput(out speed);
    // always move along the camera forward as it is the direction that it being aimed at
    Vector3 desiredMove = transform.forward*m_Input.y + transform.right*m_Input.x;
}

private void FixedUpdate()
{
    float speed;
    GetInput(out speed);
    // always move along the camera forward as it is the direction that it being aimed at
    Vector3 desiredMove = Camera.main.transform.forward*m_Input.y + transform.right*m_Input.x;
}

```

Fig. 8. Setările Camerei

Următorul pas îl reprezintă introducerea obiectelor în scenă. Pentru acest lucru s-a folosit un program gratuit (Cad Exchanger) pentru conversia fișierelor din step. în obj. pentru a putea fi inserate în Uniy.

Odată introduse obiectele în scenă, sunt poziționate folosind tab-ul Transform.

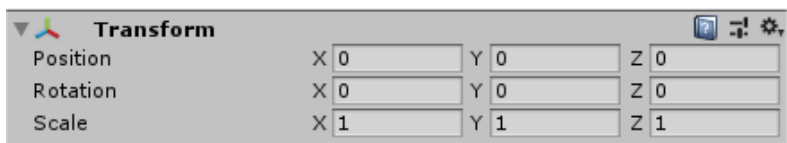


Fig. 9. Setările pentru modificarea poziției obiectelor

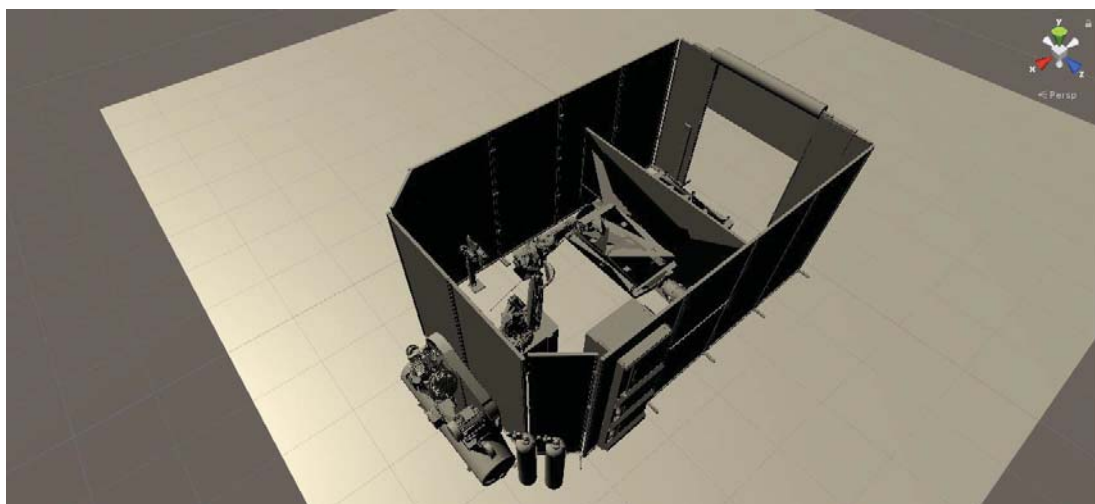


Fig. 10. Celula Completă

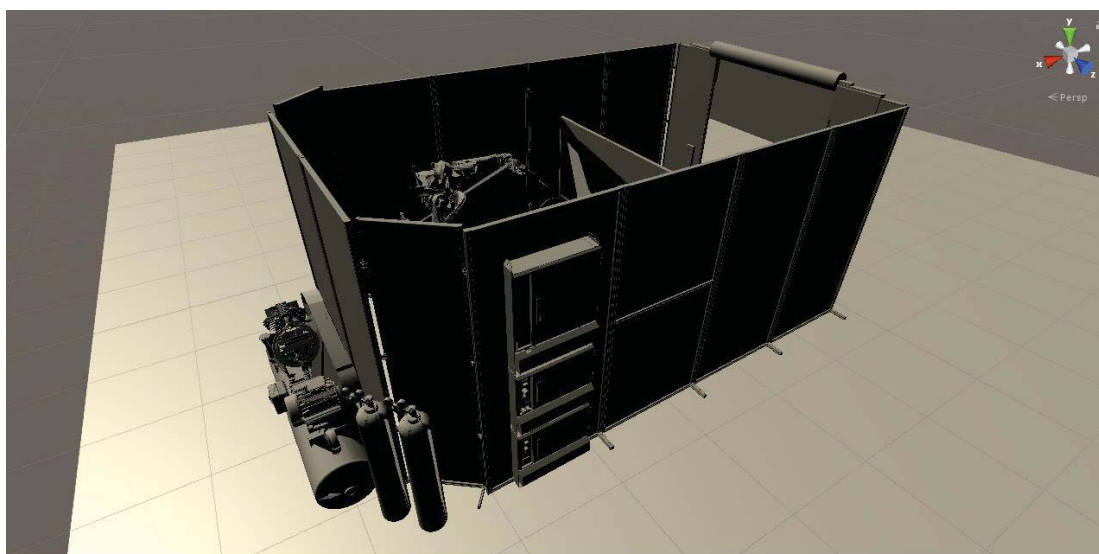


Fig. 11. Celula Completă

4. Concluzii

Aplicația a fost dezvoltată pentru sistemul de operare Android, fiind compatibilă cu orice telefon care rulează o versiune de Android 4.4 sau mai actuală.

Scopul studiului a fost dezvoltarea unei aplicații pentru vizualizarea unor astfel de celule și sisteme tehnice de către persoane ce nu au cunoștințe în utilizarea programelor de modelare asistată pe calculator, precum și de a oferi celor ce lucrează cu astfel de sisteme o nouă perspectivă asupra vizualizării obiectelor 3D, la 360° în mediu RV.

În continuarea cercetărilor, pentru a putea oferi o experiență completă, precum și pentru a putea renunța complet la prezentarea produsului în programul 3D CAD, se are în vedere introducerea animațiilor în proiectul realizat în Unity. Astfel, utilizatorul poate asista la un ciclu complet al fluxului de lucru din celulă. În plus, datorită faptului că se află într-un mediu virtual, se poate apropia și poate inspecta diferite detalii și elemente în timp ce celula robotică funcționează, lucru care este interzis în realitate.

5. Bibliografie

- [1]. <https://new.abb.com/products/robotics/industrial-robots/irb-2600id>
- [2]. <https://new.abb.com/products/robotics/controllers/irc5>
- [3]. <https://new.abb.com/products/robotics/application-equipment-and-accessories/arc-welding-equipment>
- [4]. <https://new.abb.com/products/robotics/application-equipment-and-accessories/workpiece-positioners>
- [5]. <https://www.lincolnelectric.com/en-us/Pages/default.aspx>
- [6]. <https://unity.com/>
- [7]. <https://developer.android.com/studio>