**COLEGIUL NATIONAL „ANDREI SAGUNA”**

**Cubkour**

**Atestat profesional**

**-informatica-**

Nume: Sebastian POMÎRLEANU – Mihai

Clasa: XII B

Profesori Îndrumatori: Adrian MODRIȘAN, Irina AUPICI

- Mai 2018 -

**Cuprins**

Scurta Introducere …………….………………….………………… 2

Argument asupra temei alese ………………….…………………… 2

Proceduri, functii si tehnici de programare folosite ……………….. 3

Prezentarea aplicatiei ………………….…………….………….….. 5

Cateva capturi de ecran………………….…………….…………… 8

Bibliografie ………………….…………….…………………….… 9

**Scurta introducere**

Unity este un software de creat jocuri atat 3D cat si 2D sau animatii pentru platforme multiple (MAC, Windows, Adndroid etc.). Prima oara lansat in 2005 doar pentru OS X, acum s-a extins la 27 de platforme. Pentru programare, Unity accepta limbajele C# si JavaScript. La inceput se putea programa si in BOO, dar acest lucru nu mai este posibil de la lansarea Unity 5.

Unity acepta inportare de modele 3D, sunete, imagini si diferite functii sau caracteristici pentru o grafica mai buna, crearea unui meniu sau diferite alte scopuri.

C# este un limbaj de programare orientat pe obiecte conceput de Microsoft la sfarsitul anilor 90. A fost conceput ca un concurent pentru limbajul Java. Ca si acesta, C# este un derivat al limbajului C++. C# simplifica mult scrierea de programe pentru sistemul de operare Windows.

**Argument asupra temei alese**

Tema aleasa este un joc 3D cu tematica de Parkour. Am ales aceasta tema din mai multe motive:

- am mai vrut sa fac un joc acum ceva timp in urma dar nu am reusit sa il termin;

- pentru ca imi place sa creez “*easter egg”*-uri, si unde loc mai bun de a face asta daca nu intr-un joc;

- deoarece imi place parkour-ul.

**Functii si tehnici de programare folosite**

Jocul este creat folosind mai multe script-uri scrise in limbajul C# si atribuite obiectelor.

Un script, odata creat, se poate atribui unuia sau mai multor obiecte (modele 3D/2D), depinde de scopul acestuia. El poate fi creat special pentru un obiect (exemplu: scriptul de miscare pentru player) sau poate fi general (exemplu: un script ce face obiectele sa se miste sus-jos, pentru a da o estetica mai buna).

Fiecare script creat vine cu doua functii predefinite: Start() si Update(). Pe langa acestea se pot crea altele pentru a face citirea codului si folosirea functiilor mai usoara. De asemenea, o functie sau o variabila dintru-un script, poate fi accesata dintr-un altul in functie de nevoie.

Functii predefinite (cele 2 amintite mai sus plus inca cateva functii utile) :

* *void Start()* : aceasta functie are rolul de intializare. Ea este apelata doar in momentul in care un obiect este setat ca *activ.* Acest lucru se intampla deobicei cand se da *run/play* la joc sau cand se trece la o scena noua
* *void Update():* esentiala pentru joc. Aceasta functie este apelata la fiecare *frame*. Ea este folosita pentru a verifica imputul de la tastatura. Aici se pune codul de miscare al playerului.
* void *OnTriggerEnter(Collider col)* / *OnCollisionEnter (Collision col)* : ca functia *Update()*, este apelata la fiecare *frame*. Aceste functii pot detecta coliziunile si apoi sa aplice orice cod necesar folosirii acestora (de exemplu apasarea unui buton). Obiectul cu care se colizioneaza se gaseste in variabila *col*, iar numele / tag-ul acestuia se poate accesa prin *col.gameObject.name* / *col.gameObject.tag*. Diferenta dintre cele 2 functii consta in faptul ca *pentru OnTriggerEnter()*, obiectul ce “poarta” functia trebuie sa fie de tip *trigger*, ceea ce il face sa poata trece prin restul obiectelor
* *void* *OnTriggerExit(Collider col) / OnCollisionExit(Collison col)* : se apeleaza la fiecare frame. Functiile detecteaza momnetul cand nu mai atingi obiectul.

Variabilele sau functiile, in C#, pot fi de mai multe tipuri. Cele folosite de mine sunt: *public,private* si *internal*.

* o variabila/functie de tip *public* este accesibila din orice script daca se face referirea corecta la ea, de asemenea ea poate fi modifica in orice moment atunci cand se da click pe obiectul caruia i-a fost atribuit scriptul in care se afla variabila de tip *public*.
* tipul *private* defineste o variabila/functie ca fiind accesibila doar in scriptul respectiv.
* tipul *internal* este o combinatie intre cele 2 de mai sus. Poate fi accesat si din alte scripturi, dar nu poate fi modificata dand click pe obiectul caruia i-a fost atribuit script-ul cu functia/variabila respectiva.

Tehnici de programare folosite:

Obiectul player trebuie atat sa se poata misca cat si sa detecteze coliziunea cu diferite obiecte. Uneori, folosirea unui singur obiect (cel pentru player), poate deveni destul de enervanta sau ar putea sa nu poata indeplini toate “task–urile” pe care le dorim. De exemplu, daca avem nevoie de mai multe script-uri, fiecare pentru un scop diferit, dar toate cu legatura cu player-ul, sa le atribuim pe toate unui singur obiect le va face greu de urmarit mai tarziu. Pentru acesta putem crea mai multe obiecte si sa le facem “copii” ale obiectului principal. Acesti “copii” vor detine script-urile necesare scopului lor. Player-ul meu, cu cub 3D cu numele “Cubvin”, trebuie sa poata detecta toate obiectele cu care colizioneaza. Pentru acest lucru am creat o copie, i-am dat atributul de *trigger* si l-am lasat de marimea player-ului, in exact acelasi loc. Dezactivandu-i gravitatia celui de-al 2-lea obiect, el va ramane la aceeasi pozitie reativa fata de „tatal” sau.

In joc se pot gasi mai multe „skill –uri”. Dintre acestea, „skill”-ul de „ninja” nu am reusit sa-l programez cum trebuie in nici-un mod. Ce trebuia sa faca era sa teleporteze *player-ul* cateva spatii in fata sa. Pentru a rezolva acest lucru, am creat un alt cub pe care l-am pus la distanta dorita in fata lui „Cubvin”, dezactivandu-i interactiunea gravitatiei, posibilitatea de a coliziona si facandu-l invizibil. Facand-ul copil al lui „Cubvin” , acesta isi pastreaza positia relativa fata de tatal sau. Astfel, pentru teleportare, tot ce trebuie sa fac e sa accesez cordonatele „copilului” si sa ii spun lui „Cubvin” ca vreau sa se teleporteze la acele coordonate. „Copilul” se va teleporta si el acelasi numar de spatii in fata, astfel ca „skill” –ul poate fi folosit din nou.

Tot in zona creearea obiectelor pentru player, am pus mai multe decoratiuni (modele 3D), carora le dau „disable” (nu apar in joc, dar exista) si pot sa le schimb intre ele depinde de cum trebuie/vreau sa arate „Cubvin” in momentul respectiv.

Un alt lucru la care aceasta tehnica este utila este pentru sunetul ce trebuie auzit cand *player-ul* sare. Pentru a fi redat doar o singura data la apasarea tastei ce face cubul sa sara, trebuie sa ii spunem sunetuli ca poate sa se redea doar atunci cand nu suntem in aer. Pentru acesta, am creat un cub mai mic ce poate detecta coliziunea si l-am pus la baza lui „Cubvin”. De fiecare data cand acesta atinge ceva, modifica o variabila ce comanda cand poate fi redat sunetul. Atlfel, daca tasta pentru sunet trebuie tinuta apsata, acesta s-ar reda in fiecare frame, facand-ul deranjant. Cubia lui „Cubvin” de care am spus mai sus nu era buna pentru acest scop, deoarece ea detecteaza orice coliziune. Nu vrem ca sunetul sa fie redat cand ne lovim de un perete in aer.

O alta tehnica folosita pentru a face mai usoara accesarea variabilelor generale este creearea unui obiect, pus undeva in fundal sau facut invizibil, caruia ii atribuim un script ce retine toate aceste variabile. Variabilele, fiind toate adunate intr-un singur loc, le va face mai usor de accesat. De asemenea, daca dorim sa trecem de la o scena la alta (trecerea la un nivel nou/accesarea meniului), toate obiectele scenei curente sunt sterse si sunt inlocuite cu cele ale scenei urmatoare. Daca dorim sa retinem scorul sau ”skill –ul” pe care il avem, acestea trebuie memorate in variabile si puse intr-un obiect caruia ii dam proprietatea de a nu se distruge la trecerea intre scene. Acesta este cel mai usor mod, mai ales daca functia de *Start()* de pe player trebuie apelata din nou la trecerea intre scene.

Pentru crearea meniului am folosit componentele de UI oferite de Unity, special facute pentru scopuri de genul acesta.

**Prezentarea aplicatiei**

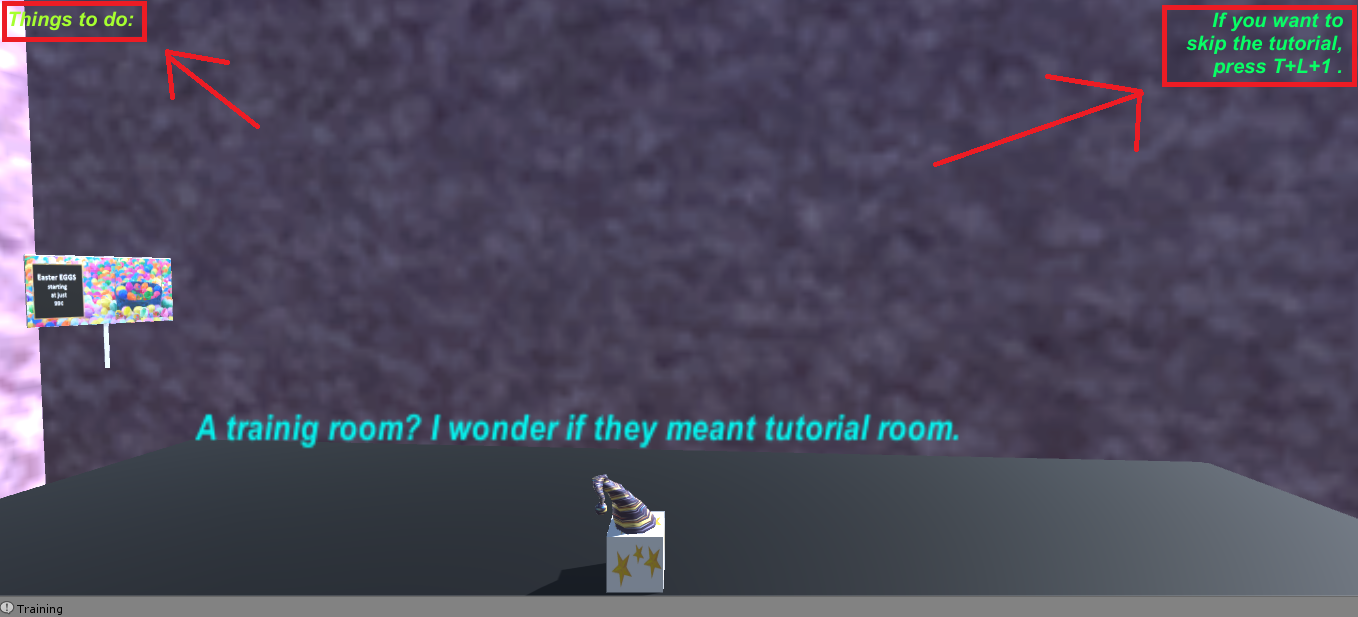
Esti un cub pe nume Cubvin si te trezesti intr-un loc necunoscut din care vrei sa scapi. Din fericire esti un practicant de „cubkour” (parkour, dar pentru cuburi), ceea ce iti faciliteaza scopul de evadare. Fiind practicant al acestui sport, poti sari si merge pe pereti. De fiecare data cand atingi un obiect cu tag de „wall”, acceleratia care te trage in jos este resetata la 0, astfel putand sari de pe un perete pe altul pentru a castiga inaltime.

Cand incepi jocul te afli in scena de tutorial. Propoziti vor aparea deasupra cubului care iti vor spune atat cum se controleaza player-ul cat si un inceput din povestea jocului. Tutorialul se incheie cu o animatie ce iti deschide tunelul catre nivelul 1, unde ai de colectat niste chei.

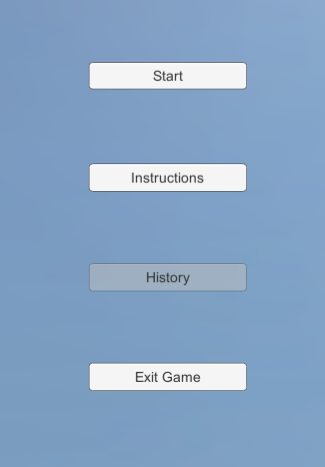
Cubul se controleaza cu tastele W,A,S,D (pentru miscare), tasta „space” pentru a sari si mouse-ul pentru a te uita imprejur. Daca deblochezi vreun „skill”, acesta se activeaza cu tasta F. Momentan exista un singur skill pus in joc, dintr-un total de 3 (toate „skill”-urile au fost programate, dar nu se pot afla toate in acceasi scena din joc, caci distruge distractia de a le cauta).

Jocul are si cateva „easter egg”-uri (zone secrete / mesaje ascunse). „Skill”-urile de care am amintit mai sus sunt „easter egg”, acesta fiind scopul pentru care au fost creeate. Cel care poate fi gasit in joc a fost inspirat din personajul Bill Cipher din desenul Gravity Falls, facut de Alex Hirsch. Cand „skill”-ul este activ, „Cubvin” devine temporar un triunghi galben.

In timpul jouclui, in colturile de sus, atat in partea dreapta cat si in partea stanga, se afla doua texte colorate diferit. Acestea au rolul de a ajuta player-ul sa se miste prin joc. Textul din coltul din dreapta-sus, are rolul de a afisa mesaje sau indici, pe cand celalalt este pentru a afisa scorul sau ce ai de facut.



La deschiderea aplicatiei va aparea un meniu:



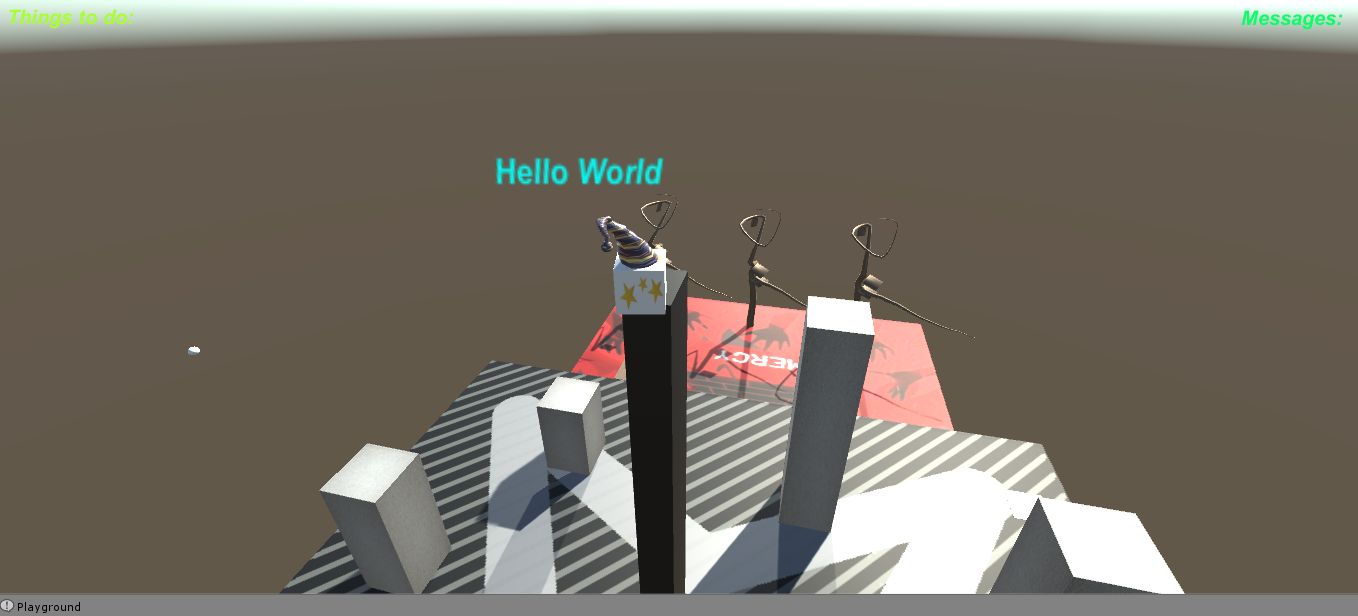
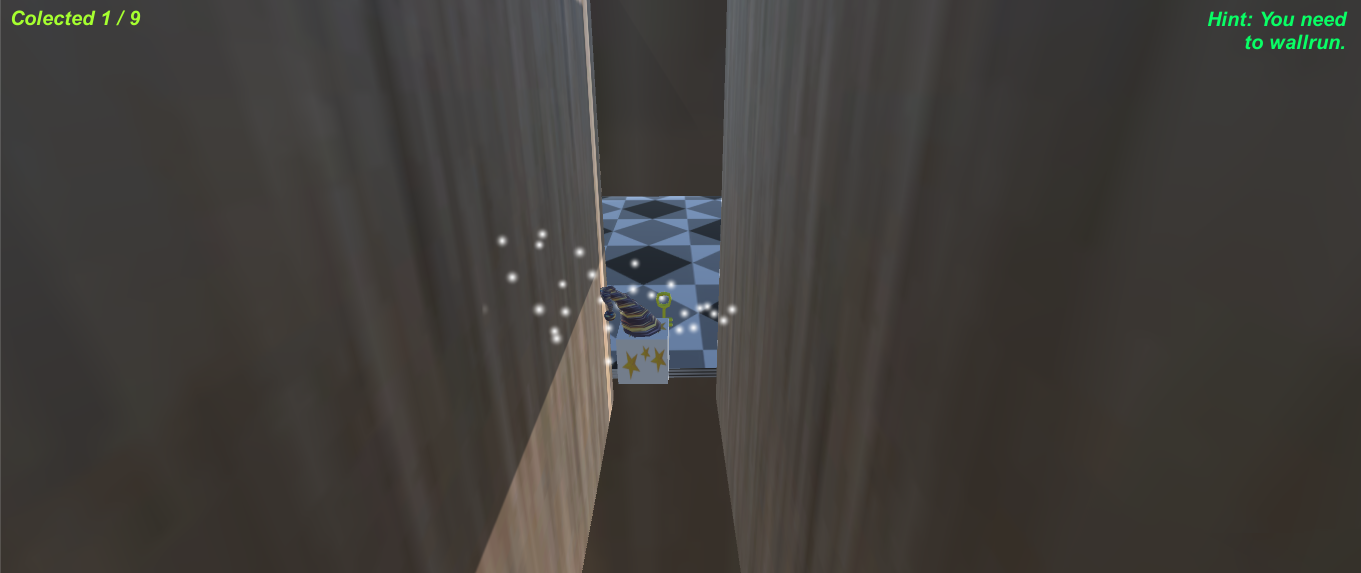
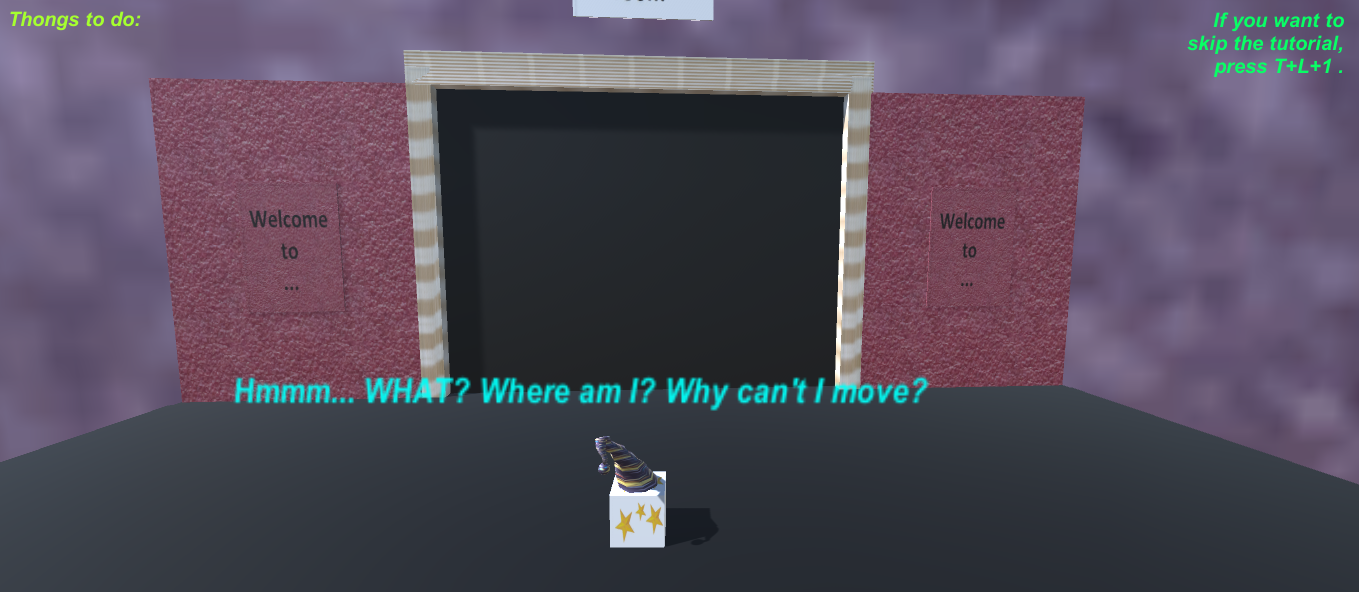
El are 4 butoane: „Start”, „Instructions”, „Hisory” si „Exit Game”. Butonul de „History” e pus mai mult de design, el fiind „disabled” momentan (va deveni util in varsiunile viitoare ale jocului).

Butonul de „Start” este cel care porneste jocul.

Butonul „Instructions” schimba panoul de meniu cu unul in care se povesteste pe scurt ce e cu jocul. De asemenea, apare un buton, „Back”, care odata apasat ne duce inapoi la meniu.

Butonul „Exit Game” face ca un nou panou sa apara, panou unde jucatorul este intrebat daca chiar doreste sa paraseasca jocul. Optiunile sale sunt „No” sau „Yes”. „No” are aceeasi functie ca butonul de „Back”, iar „Yes” inchide aplicatia.

**Cateva capturi de ecran**

****

**Bibliografie**

\*unity3d.com

\*https://www.dalecarnegie.ca/wp-content/uploads/2016/01/Reset-Button-300x300.jpg (imagine reset button)

\*https://vignette.wikia.nocookie.net/the-paper-puppets-wiki-object-show/images/a/ac/Shuriken.png/revision/latest?cb=20140727162741 (shuriken)

\*https://ih1.redbubble.net/image.243502115.9139/flat,800x800,075,f.u1.jpg (Magician Hat)

\*https://t2.rbxcdn.com/176671248530746c656f1bab33af8fcb (Tapet Pereti)

\*https://www.bcbrick.com/wp-content/uploads/2016/10/porcelain-stair-treads-granite.jpg (Tapet Piatra)

\*https://www.arroway-textures.ch/sites/default/files/styles/huge/public/texture/demo/crop\_concrete-052.jpg?itok=E9HjGadK (tapet piatra V2)

\*https://ih1.redbubble.net/image.394111990.2657/flat,800x800,075,f.u1.jpg (tapet\_de\_sah)

\*and a couple other things form youtube

\*https://1.bonami.cz/images/products/8c/b0/8cb0c66d73d6af8ee3ff87e18914c8cc3eda54e1-1000x1000.jpeg (Tapet covor mov)

\*https://s3.amazonaws.com/spoonflower/public/design\_thumbnails/0315/5389/rrrrrblack\_sketched\_arrows\_spoonflower-01\_shop\_preview.png (wall arrows)

\*http://open3dmodel.com/download/magic-hat-3d-model\_2411.html (Hat 3D model)

\*https://free3d.com/3d-model/katana-75134.html (Katana 3D model)

\*https://vignette.wikia.nocookie.net/clubpenguin/images/6/68/Magician\_Cape\_clothing\_icon\_ID\_305.png/revision/latest?cb=20130810161321(Tapet\_MagicianCape)

\*http://static.wpe.au.syrahost.com/var/m\_4/47/472/4233/38998-stars.jpg (Magician\_Stars)

\*http://www.mobiusengine.co.uk/invocation/site/invocation-source/circle-fullimage.gif (Summoning circle no 1)

\*https://www.youtube.com/watch?v=P-BqmegCUpA

\*plus anouther couple of other things from Google Image

\*https://www.youtube.com/watch?v=ew1TpesH-jw (OMGF-ok; background music)

\*https://free3d.com/3d-model/low-poly-key-680139.html (3D key)