



cub3D

MiniLibX-ով իմ առաջին RayCaster-ը

Հակիրճ. այս նախագծի գաղափարը գալիս է 90-ականների աշխարհահոչակ մի խաղից, որն ամենաառաջին FPS `First person-shooter (խաղ՝ ցուցադրված կրակողի աչքերով) խաղն էր: Այս նախագծի շրջանակներում կուտումնասիրեք ճառագայթի հեռարձակումը (ray-casting): Զեր նպատակը լարիրինթոսում դինամիկ տեսարան ստեղծելն է, որտեղ պետք է ձեր ճանապարհը գտներ:



Յանկ

I	Ընդհանուր ցուցումներ	2
II	Նախաբան	4
III	Նպատակներ	5
IV	Պարտադիր մաս - cub3D	7
V	Բունուսային մաս	13
VI	Օրինակներ	15

Գլուխ I

Ընդհանուր ցուցումներ

- Զեր նախագիծը պետք է գրվի Norm-ին համապատասխան: Եթե ունեք բոնուսային Փայլեր/Փունկցիաներ, դրանք ևս ստուգվելու են Norm-ով, և Norm-ի սխալի դեպքում կստանաք օ միավոր:
- Բացառությամբ չսահմանված վարքագծերի (սեգմենտացիայի սխալ, bus սխալ, կրկնակի free և այլն)` ձեր Փունկցիաները չպետք է անսպասելի ավարտվեն: Եթե դա տեղի ունենա, նախագիծը կհամարվի ոչ Փունկցիոնալ և գնահատման ժամանակ կստանաք օ միավոր:
- Հատկացված դինամիկ բաշխվող հիշողությունն անհրաժեշտության դեպքում պետք է ամբողջությամբ պատշաճ կերպով ազատվի: Արտահոսքերն անընդունելի են:
- Եթե նյութը պահանջում է, ապա պետք է հանձնել Makefile, որը ձեր ելակետային Փայլը կկազմարկի պահանջվող ելքի՝ -Wall, -Wextra և -Werror դրոշակներով, իսկ ձեր Makefile-ը չպետք է վերակապի:
- Զեր Makefile-ն պետք է պարունակի առնվազն \$(NAME), all, clean, fclean և re կանոնները:
- Զեր նախագծի հետ բոնուսներ հանձնելու համար պետք է Makefile-ում ներառել bonus կանոն, որը կավելացնի բոլոր հնարավոր վերնագրերը, գրադարանները կամ Փունկցիաները, որոնք արգելված են նախագծի հիմնական մասում: Բոնուսները պետք է լինեն առանձին Փայլում՝ _bonus.{c/h}-ում: Պարտադիր և բոնուսային մասերի գնահատումը կատարվում է իրարից անկախ:
- Եթե նախագիծը թույլ է տալիս օգտագործել ձեր libft-ը, ապա պետք է պատճենեք դրա ելակետային Փայլերը և համապատասխան Makefile-ը libft պանակում՝ իր համապատասխան Makefile-ով: Զեր նախագծի Makefile-ը պետք է կազմարկի գրադարանը՝ օգտագործելվ դրա Makefile-ը, իսկ հետո՝ կազմարկի նախագիծը:
- Խրախուսելի է ձեր նախագծի համար թեստային ծրագրերի ստեղծումը, որոնք, սակայն, **հաշվի չեն առնվի և չեն գնահատովի**: Դա հնարավորություն կընձեռնի հեշտությամբ թեստավորել ձեր և ձեր ընկերների աշխատանքը: Այս թեստերը

հատկապես օգտակար կլինեն պաշտպանության ժամանակ: Ավելին, պաշտպանության ընթացքում ազատորեն կարող եք օգտագործել ինչպես ձեր, այնպես էլ այն ուսանողների թեստերը, ում աշխատանքը գնահատում եք:

- Ձեր աշխատանքը պետք է ներառել հանձնարարված ցու պահոցում: Կգնահատվի միայն ցու պահոցի աշխատանքը: Եթե նախատեսվում է, որ DeepThought-ն է գնահատելու ձեր աշխատանքը, ապա դա կարվի ձեր ընկերների գնահատումից հետո: Եթե DeepThought-ի գնահատման ընթացքում աշխատանքի ցանկացած մասում որևէ սխալ ի հայտ գա, գնահատումը կընդիմանվի:



Գլուխ II

Նախաբան

Id Software-ը մշակվել է աշխարհահռչակ Զոն Քարմակի ու Զոն Ուսմեռյի կողմից, հրապարակվել է 1992-ին Apogee Software-ի կողմից: Wolfenstein 3D-ն տեսախաղերի պատմության մեջ առաջին իրական խաղն է՝ «կրակողի աշբերով»:



Նկար II.1: Զոն Ուսմեռն (ձախից) ու Զոն Քարմակը (աջից) :

Wolfenstein 3D-ն այնպիսի խաղերի նախահայրն է ինչպիսիք են Doom-ը (Id Software, 1993), Doom II-ը, (Id Software, 1994), Duke Nukem 3D-ն (3D Realms-ը, 1996) ու Quake-ը (Id Software, 1996), որոնք տեսախաղերի աշխարհում խևական ականներ են:

Հիմա պատմությունը վերապրելու ձեր հերթն է...

Գլուխ III

Նպատակներ

Այս նախագծի նպատակը հիմնական ալգորիթմների օգտագործումը և տեղեկության ուսումնասիրությունն է:

Հայ 3D-ն՝ որպես գրաֆիկ դիզայնի նախագիծ, ձեզ հնարավորություն կտա բարելավել պատուհաններ, գույներ, դեպքեր ստեղծելու, պատկերներ ներկելու և այլ հմտություններ:

Այսիսով, սահմանական կիրառությունները ուսումնասիրելու համար՝ առանց մանրամասները հասկանալու:

Համացանցում հասանելի մի շարք նյութերի օգնությամբ դուք մաթեմատիկան կօգտագործեք որպես էլեզանտ ալգորիթմներ գրելու գործիք:



Խորիուրդ ենք տալիս մինչև այս նախագիծը սկսելը թեստավորել
խաղի բնօրինակը՝ <http://users.atw.hu/wolf3d/>:

:

Գլուխ IV

Պարտադիր մաս - cub3D

Ծրագրի անուն	cub3D
Հանձնվելիք Փայլեր	Չեր բոլոր ֆայլերը
Ստեղծել Փայլ	all, clean, fclean, re, bonus
Արգումենտներ	*.cub ձևաչափով մի քարտեզ
Արտաքին Փունկցիաներ	<ul style="list-style-type: none">բացել (open), փակել (close), կարդալ (read), գրել (write), malloc, դատարկել (free), perror (սխալ նկարագրող ստանդարտ հաղորդագրություն ստեղծող Փունկցիա է), strerror (տողին վերադարձնում է ցուցիչ, որ նկարագրում է կոդի մեջ առկա սխալը), դուրս գալ (exit)մաթեմատիկայի գրադարանի բոլոր Փունկցիաները (-lm ման 3 մաթ)MinilibX-ի բոլոր Փունկցիաները
Libft թույլատրված	Այն
Նկարագրություն	Պետք է ստեղծել լարիրինթոսի ներսի «ուելիստիկ» 3D գրաֆիկական ներկայացում՝ խաղացողի տեսանկյունից: Այս ներկայացումը պետք է ստեղծել՝ վերևում ներկայացված հիմունքները օգտագործելով:

Սահմանափակումները հետևյալն են՝

- *Պետք է* օգտագործել `miniLibX-ը`՝ կամ այն տարրերակը, որ հասանելի է օպերացիոն համակարգում, կամ այն, որ առկա է հիմնական կոդում: Եթե որոշեք հիմնական կոդից օգտվել, `libft`-ի համար նույն կանոնները պետք է կիրառել, ինչ վերևում նշված *Հնդիանուր ցուցումների* մասում:
- Զեր պատուհանի կառավարումը պետք է սահուն լինի ուրիշ պատուհան տեղափոխվելիս, պատուհանը փոքրացնելիս և այլն:
- *Պետք է* ցուցադրվեն պատի տարրեր կառուցվածքներ (ըստ ձեր ընտրության)` կախված նրանից, թե պատերը որ կողմ են նայում (հյուսիս, հարավ, արևելք, արևմուտք):

- Ծրագիրը պետք է կարողանա պատի փոխարեն անիմացված կետագծային պատկեր ներկայացնել (sprite):
- Ծրագիրը պետք է կարողանա հատակի ու առաստաղի գույները փոխել:
- Եթե Depththought-ը ցանկություն ունենա ձեր նախագիծը գնահատելու, ծրագիրը պետք է պահպանի առաջին մշակված նկարը եղբ ձևաչափով, եթե դրա երկրորդ արգումենտը "----save"-ն է:
- Եթե երկրորդ արգումենտ չի տրամադրվել, ծրագիրը պետք է նկարը արտածի պատուհանի մեջ ու հետևի հետևյալ կանոններին:

 - Ստեղնաշարի աջ և ձախ սլաքներով կոճակները պետք է թույլ տան տեսնել լաբիրինթոսի աջ ու ձախ կողմերը:
 - W, A, S և D կոճակները պետք է թույլ տան լաբիրինթոսի մեջ դիտակետը փոխել:
 - ESC սեղմելը պետք է փակի պատուհանը ու ծրագիրը ամբողջությամբ անջատի:
 - Պատուհանի շրջանակի կարմիր խաչը պետք է ծրագիրը ամբողջությամբ փակի:
 - Եթե քարտեզում նշված Էկրանի չափը ավելի մեծ է քան Էկրանի տարրուծում (resolution), պատուհանի չափը պետք է սահմանափակվի Էկրանի չափերով:
 - minilibX-ի պատկերների օգտագործումը խիստ խրախուսվում է:

- Չեր ծրագիրը պետք է որպես առաջին արգումենտ վերցնի տեսարանի նկարագրման ֆայլ՝ .cub ընդլայնմամբ:

 - Քարտեզը պետք է կազմված լինի միայն 4 նիշերից՝
 - **0**՝ քացատի համար, **1**՝ պատի համար, **2**՝ առարկայի համար ու **N,S,E** կամ **W**՝ խաղացողի սկզբնական դիրքի ու տեղորշման համար:

Ահա մի պարզ, վավեր քարտեզի օրինակ՝

```
% \item The map must be composed of only 4 possible characters:
%   \textbf{0} for an empty space, \textbf{1} for a wall, \textbf{2} for an
%   item and \%textbf{N},\textbf{S},\textbf{E} or \textbf{W} for the player's
%   start position and spawning orientation.\\
%
% This is a simple valid map: \begin{42console}
111111
100101
102001
1100N1
111111
```

- Քարտեզը պետք է փակված/շրջապատված լինի պատերով: Եթե այդպես չէ, ծրագիրը պետք է սխալ վերադարձնի:

- Քարտեզի պարունակությունից բացի՝ յուրաքանչյուր տեսակի տարր կարող է բաժանվել մեկ կամ մի քանի գծերով:
- Քարտեզի պարունակությունից բացի, ինչը միշտ պետք է վերջինը լինի, տարրերի տեսակները Փայլում կարող են դրվել ցանկացած հերթականությամբ:
- Քարտեզից բացի՝ յուրաքանչյուր տարրի մասին տեղեկություն կարող է բաժանված լինել մի կամ մի քանի բացատներով:
- Քարտեզը պետք է վերլուծված լինի այնպես, ինչպես ցույց է տրված Փայլում: Բացատները քարտեզի ընդունելի մաս են, դուք եք որոշում՝ ինչպես վարվել դրանց հետ: Պետք է կարողանաք վերլուծել կամայական տեսակի քարտեզ, քանի դեռ այն բավարարում է քարտեզի կանոններին:

- Յուրաքանչյուր տարրի (քարտեզից բացի) առաջին տեղեկությունը տեսակի նույնականացնելու համար է (կազմված մեկ կամ երկու նիշից), որին հաջորդում է հատուկ տեղեկություն յուրաքանչյուր օբյեկտի մասին՝ հստակ հերթականությամբ՝

* լուծողունակություն՝

R 1920 1080

- նույնականացնելու համար՝ **R**
- x-ի ներկայացման չափ
- y-ի ներկայացման չափ

* հյուսիսի կառուցվածք՝

NO ./path_to_the_north_texture

- նույնականացնելու համար՝ **NO**
- հյուսիսի կառուցվածքի ուղի

* հարավի կառուցվածք՝

SO ./path_to_the_south_texture

- նույնականացնելու համար՝ **SO**
- հարավի կառուցվածքի ուղի

* արևմուտքի կառուցվածք՝

WE ./path_to_the_west_texture

- նույնականացնելու համար՝ **WE**
- արևմուտքի կառուցվածքի ուղի

* արևելքի կառուցվածք՝

EA ./path_to_the_east_texture

- նույնականացնելու համար՝ **EA**
- արևելքի կառուցվածքի ուղի

* անիմացված կետագծային նկար

S ./path_to_the_sprite_texture

- նույնականացնելու համար՝ **S**
- անիմացված կետագծային նկարի կառուցվածքի ուղի

* հատակի գույն՝

F 220,100,0

- նույնականացնելու համար՝ **F**
- R,G,B գույները [0,255] տիրույթում՝ **0, 255, 255**

* Առաստաղի գույն՝

C 225,30,0

- նույնականացնեցի՛շ՝ **C**
- R,G,B գույները [0,255] տիրույթում՝ **0, 255, 255:**

- Պարտադիր մասի օրինակ .cub-ի մինիմալիստական տեսարանով՝

```
R 1920 1080
NO ./path_to_the_north_texture
SO ./path_to_the_south_texture
WE ./path_to_the_west_texture
EA ./path_to_the_east_texture

S ./path_to_the_sprite_texture
F 220,100,0
C 225,30,0

111111111111111111111111111111
10000000001100000000000001
1011000001110000002000001
10010000000000000000000001
1111111101100000111000000000001
1000000000110000011101111111111
1111011111111011100000010001
11110111111111011101010010001
11000000110101011100000010001
10002000000000001100000010001
100000000000000001101010010001
11000001110101011111011110N0111
11110111 1110101 101111010001
11111111 1111111111111111
```

- Եթե Փայլում որևէ սխալ կոնֆիգուրացիա հանդիպի, ծրագիրը պետք է ինչպես հարկն է դուրս գա ու վերադարձնի՝ "Error\n", որին հաջորդում է սխալի պարզ հաղորդագրություն՝ ըստ ձեր ցանկության:

Գլուխ V

Բունուսային մաս



Բունուսները կսուրգվեն միայն այն դեպքում, եթե ձեր պարտադիր մասը ԿԱՏԱՐՅԱԼ է: ԿԱՏԱՐՅԱԼ ասելով՝ բնականաբար նկատի ունենք, որ այն պետք է ավարտված լինի: Այն չի կարող ձախողվել նույնիսկ ամենավատ սխալների դեպքում՝ ինչպես օրինակ սխալ գործածությունների և այլն: Այսինքն՝ նշանակում է, որ եթե ձեր պարտադիր մասը գնահատման ընթացքում առավելագույն միավորները չստանա, բունուսային մասը ամբողջությամբ ԿԱՐՀԱՍՏԱՐՀՎԵ:

Բոնուսային ցուցակ՝

- պատի վրա բախումներ
- մի skybox (հետևի ֆոն ստեղծելու մեթոդ)
- հատակի և/կամ առաստաղի տարբերակներ
- մի HUD (heads-up display՝ խաղի միջավայրի՝ խաղացողին ներկայացված մեթոդի վիճակ)
- վերև ու ներքև նայելու հնարավորություն
- ցատկել կամ կոանալ
- տարածության հետ կապված ստվերի եֆեկտ
- կյանքի սանդղակ
- լարիրինթոսի առարկաներ
- առարկաների բախումներ
- կյանքի միավորներ վաստակել և/կամ կորցնել՝ առարկաների/ծուլակների հանդիպելիս
- դոներ, որ կարող են բացվել ու փակվել
- գաղտնի դոներ

- զենքի կրակոցի անհմացիա կամ անհմացված կետապատկերային նկար, որը ավելի մեծ տեսարանի մաս է
- մի քանի մակարդակներ
- ձայներ ու երաժշտություն
- մկնիկով տեսակետը փոխելու հնարավորություն
- զենքեր ու թշնամիներ:



Բոնուսային բոլոր միավորները ստանալու համար, պետք է վավերացնել նրանցից գոնե 14-ը, այնպես որ ուշադիր ընտրե՛ք, որ ձեր ժամանակը չվատներ:



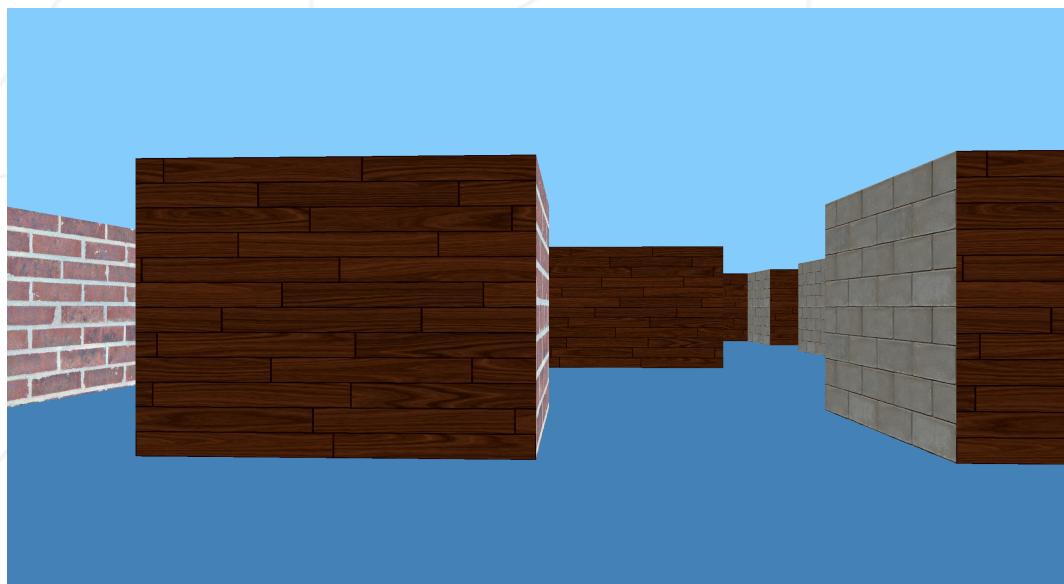
Զեզ թույլ է տրվում բոնուսային մասը կատարելու համար տարբեր ֆունկցիաներ օգտագործել քանի դեռ դրանց գործածումը արդարացված է ձեր գնահատման ժամանակ: Զեզ նաև թույլ է տրվում ձևափոխել սպասված տեսարանի Փայլի ձևաչափը, եթե ձեզ այդպես է հարմար: Խելացի եղե՛ք:

Գլուխ VI

Օրինակներ



Նկար VI.1: Wolfenstein3D խաղի բնօրինակը՝ RayCasting օգտագործելով:



Նկար VI.2: Օրինակ,թե ինչպիսին պետք է լինի ձեր նախագծի պարտադիր մասը:



Նկար VI.3: Բոնուսային մասի օրինակ փոքր քարտեզով, հատակով, առաստաղի տարբերակներով ու անիմացված հայտնի ողնիով:



Նկար VI.4: Բոնուսային մասի մի ուրիշ օրինակ HUD-ով, կյանքի սանդղակով ձայներով ու երաժշտությամբ, ստվերի էֆեկտով ու կրակող զենքով:



Նկար VI.5: Բոնուսային մեկ այլ մաս զենքի մի ուրիշ ընտրությամբ ու առաստաղին նայող խաղացողով: