

ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ

Мэдээлэл, холбооны технологийн сургууль



БАГААР БҮТЭЭХ ПРОГРАММЧЛАЛ

Программчлалын дадлага (F.CSM360)
2023-2024 оны хичээлийн жилийн хавар

Шалгасан багш: Б.Алтантуяа
Гүйцэтгэсэн: З.Анхбаяр, B232270029
Д.Нурсолтан, B232270039
О.Мэнд-Амар, B232270001
С.Асилан, B232270115

Улаанбаатар хот
2024 он

Агуулга

1	Хураангуй	3
2	Онолын судалгаа	3
3	Алгоритм зохиомж	4
3.1	Үйл ажиллагааны диаграмм	4
3.2	Объектын холбоосын диаграмм	4
3.3	Класс диаграмм	5
3.4	Дэлгэцүүдийн зохиомж (Prototype)	9
3.5	Үндсэн функцуудын псевдо алгоритмууд	9
4	Үр дүн	10
5	Дүгнэлт	11

1 Хураангуй

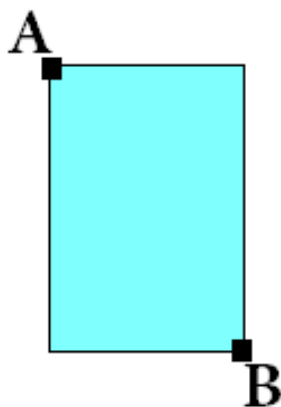
Манай баг платформ төрлийн тоглоом хийхээр ярилцсан. Платформ тоглоом (заримдаа үсэрч гүйдэг тоглоом гэж нэрлэдэг) нь тоглоомын гол зорилго нь тоглогчийн дүрийг хүрээлэн буй орчны цэгүүдийн хооронд шилжүүлэх явдал юм. Платформ тоглоомууд нь тэгш бус газар нутагтай, янз бүрийн өндөртэй дүүжин тавцантай тул үсрэх, авирах шаардлагатай байдаг. Зарим акробатын хөдөлгөөн модноос дүүжлэх эсвэл дэгээгээр цохих, хананаас үсрэх, агаарт гулсах, трамплин эсвэл трамплинаас үсрэх зэрэг тоглоомын явцтай холбоотой байж болно.

2 Онолын судалгаа

Платформ тоглоомыг хөгжүүлэх нь үр дүнтэй тоглоомын механик болон гүйцэтгэлийг хангахын тулд хэд хэдэн үндсэн өгөгдлийн бүтэц, сонгодог алгоритмуудыг ашиглах явдал юм.

Мөргөлдөөн мэдрэх

- Мөргөлдөөний хамгийн энгийн тест бол хязгаарлах хайрцагны шалгалт юм. Хязгаарлах хайрцгийг шалгах нь пикселийн мөргөлдөөнтэй тоглоомуудын ачааллыг бууруулах хамгийн хялбар бөгөөд хамгийн үр дүнтэй арга юм! Хоёр спрайт мөргөлдөж байгаа эсэхийг шалгахын өмнө мөргөлдөх магадлал хангалттай байгаа эсэхийг шалгана. Хэрэв тэдгээрийн хязгаарлах хайрцагнууд нь хүрэхгүй бол бүх пикселүүд нь тэдгээрийн хязгаарлах хайрцаг дотор байх ёстой тул пикселийн мөргөлдөөнийг шалгах ямар ч шалтгаан байхгүй. Хязгаарлах хайрцаг нөгөөд хүрч байгаа эсэхийг шалгах энгийн зүйл юм. Бидэнд OB1 ба OB2 гэсэн 2 объект байгаа бөгөөд тус бүр нь 4 координаттай, OB1-ийн хувьд энэ нь:



Зураг 1: Хязгаарлах хайрцаг

OB1 дээд, OB1 доод, OB1 Зүүн, OB1 Баруун, OB2-д адилхан
OB1 дээд нь A цэгийн у координат;

OB1 доод нь B цэгийн у координат;
OB1 зүүн нь A цэгийн х координат;
OB1 баруун нь B цэгийн х координат юм.[1]

Тоглогчийн хөдөлгөөн

- Тоглож байгаа хүн keyboard дээр товчлуур дарахад тоглоомын дүр баруун/зүүн, үсрэх, суух гэх мэт хөдөлгөөн хийнэ. Хөдөлгөөнийг нүдэнд эвтэйхэн харагдуулж тоглоход таатай мэдрэмж төрүүлэхийн тулд хурдатгал оруулж өгнө.

Физик симуляц

- Физикийн үндсэн тэгшитгэлийг ашиглан дүрийн босоо хурд, байрлалыг цаг хугацааны өөрчлөлтөд шинэчлэнэ.
- Үрэлтийг бодит хөдөлгөөн, зогсолтыг дуурайж, хурдыг аажмаар бууруулах зорилгоор ашигладаг.

Дайсны хүний хөдөлгөөн

- Дайсны хүн наашаа цаашаа хөдлөх ба харагдсан тохиолдолд тодорхой үйлдэл хийдэг байна.

Орчин

- Тоглоомын эргэн тойрон орчин нь жижиг хавтангуудаас бүрдэх ба эдгээрийг угсарснаар бага хугацаанд үеүүдийг бүтээх боломжтой.

Дүрс боловсруулах

- Parallax scroll нь тоглоомын арын фон нь наад талын дүрснээсээ өөр хурдаар хөдөлснөөр наана цаана байгаа гэдгийг ялгаж харагдуулна..

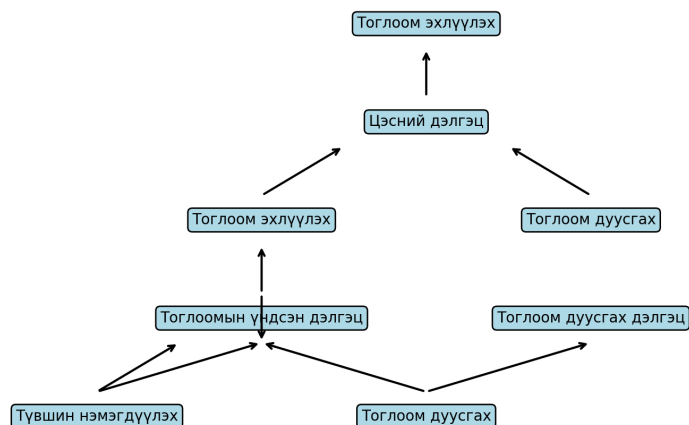
3 Алгоритм зохиомж

3.1 Үйл ажиллагааны диаграмм

Үйл ажиллагааны диаграмм нь тоглоомын үйл явцыг дүрслэн үзүүлдэг. Тоглогчийн хөдөлгөөн, мөргөлдөөн мэдрэх, бусад объектуудын үйлдлүүдийг харуулна. Энэхүү алгоритмын зохиомж нь 2D платформ тоглоомын элементүүдийг агуулсан бөгөөд Pygame модулийг ашиглан хийгдсэн. Тоглоомын гол логик, зураг, дуу хөгжим, объектуудын холбоос болон функцуудыг зохиомжлоход хэрэглэгдэнэ.

3.2 Объектын холбоосын диаграмм

Объектын холбоосын диаграмм нь тоглоомын объектуудын хоорондын холбоосуудыг харуулдаг. Үүнд тоглогч, дайсан, тавцан зэрэг объектуудын харилцан үйлчлэл орно.



Зураг 2: Үйл ажиллагааны диаграмм

Харилцаанууд

Тоглогч <|– Дайсан:

-Энэ харилцаа нь Дайсан классын объектууд нь Тоглогч классын объектуудтай харилцдаг бөгөөд Дайсан нь Тоглогч классын төлөвийг өөрчлөх боломжтойг илэрхийлнэ.

Орчин <|– Тоглогч:

-Орчин класс нь Тоглогч классын объектуудыг агуулдаг бөгөөд Тоглогч нь Орчин классын нэг хэсэг болохыг илэрхийлнэ.

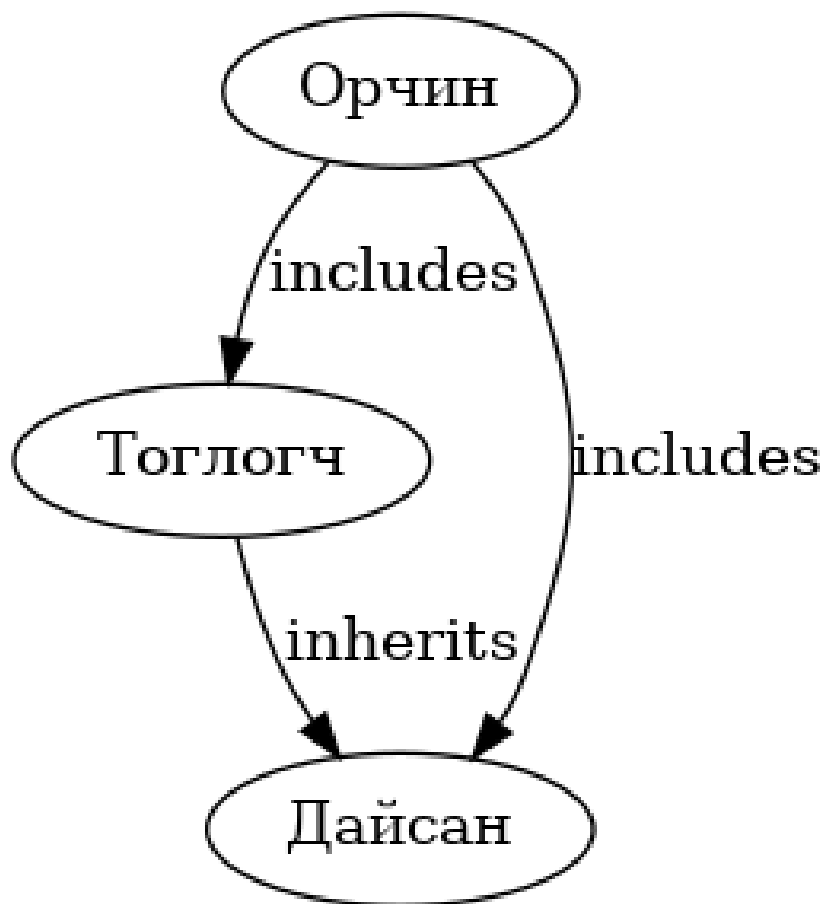
Орчин <|– Дайсан:

-Орчин класс нь Дайсан классын объектуудыг агуулдаг бөгөөд Дайсан нь Орчин классын нэг хэсэг болохыг илэрхийлнэ.

Энэ диаграмм нь тоглоомын ерөнхий бүтэц болон объектуудын хоорондын харилцааг тодорхойлж, ойлгоход хялбар болгодог.

3.3 Класс диаграмм

Класс диаграмм нь системийн үндсэн классууд болон тэдгээрийн хоорондын харилцааг дүрслэхэд ашиглагддаг. Доорх диаграммд тоглоомын гол классууд болон тэдгээрийн харилцааг харуулсан болно.



Зураг 3: Объектын холбоосын диаграмм

1. Bullet Класс:

Үйлдлүүд (Methods):

`update(screen_width: int, player: Player, bullet_group: Group, enemy_group: Group, obstacles: list, screen_scroll: int)` - Сумын байрлалыг шинэчлэх.
`draw(screen: Surface)` - Сумыг дэлгэцэн дээр зурна.

2. World Класс:

"World" класс нь тоглоомын ертөнцийг дүрслэхэд ашиглагддаг:

Үйлдлүүд (Methods):

`-process_data(data: list, img_list: list, tile_size: int, enemy_group: Group,`

item_box_group: Group, item_boxes: dict,
decs_group: Group, exit_group: Group,
spike_group: Group) - Дэлхийн өгөгдлийг боловсруулах.

draw(screen: Surface, screen_scroll: int)
- Дэлхий болон түүний элементүүдийг дэлгэц дээр зурах.

3.Player Класс:

Шинж чанарууд (Attributes):

-**health: int:** Тоглогчийн эрүүл мэндийн төлөвийг хадгалдаг атрибут.
-**ammo: int:** Тоглогчийн сумны тоог хадгалдаг атрибут.
-**jump: bool:** Тоглогч үсрэх төлөвийг хадгалдаг атрибут.
-**alive: bool:** Тоглогч амьд эсэхийг хадгалдаг атрибут.

Үйлдлүүд (Methods):

-**update()** - Тоглогчийн төлөвийг шинэчлэх.
-**draw(screen: Surface)** - Тоглогчийг дэлгэц дээр зурах.

-move(left: bool, right: bool, gravity: float, obstacles: list,
screen_width: int, screen_height: int, scroll_thresh: int,
screen_scroll: int, bg_scroll: int, level_len: int, tile_size: int,
spike_group: Group, exit_group: Group) - Тоглогчийн хөдөлгөөнийг удирдана.

shoot(bullet_group: Group, rock_img: Surface,
throw_fx: Sound) - Тоглогчийг буудах үйлдэл.

4.Button Класс:

Шинж чанарууд (Attributes):

-**image:** Surface - Товчны зураг.
-**rect:** Rect - Товчны дөрвөлжин хүрээ.
-**scale:** float - Товчны хэмжээ.

Үйлдлүүд (Methods):

-**draw(screen: Surface): bool** - Товчныг дэлгэцэн дээр зурна, товч дарсан эсэхийг буцаана.

5.HealthBar Класс:

Үйлдлүүд (Methods):

-draw(health: int, screen: Surface) - Эрүүл мэндийн түвшинг дэлгэцэн дээр зурна.

6.ItemBox Класс:

Үйлдлүүд (Methods):

-update(player: Player, screen_scroll: int)

- Эд зүйлсийн хайрцагийн төлөвийг шинэчлэх.

-draw(screen: Surface) - Эд зүйлсийн хайрцагийг дэлгэц дээр зурах.

7.ScreenFade Класс:

Шинж чанарууд (Attributes):

-direction: int - Бүдгэрэлтийн чиглэл.

-color: tuple - Бүдгэрэлтийн өнгө.

-speed: int - Бүдгэрэлтийн хурд.

-fade_counter: int - Бүдгэрэлтийн тоолуур.

Үйлдлүүд (Methods):

-fade(screen: Surface, screen_width: int, screen_height: -int): bool

- Дэлгэцийн бүдгэрэлтийг гүйцэтгэх.

Харилцаа:

-Player -> Bullet: Тоглогч сумыг удирддаг.

-Player -> Button: Тоглогч товчийг удирддаг.

-Player -> HealthBar: Тоглогчийн эрүүл мэндийн түвшинг харуулдаг.

-Player -> ItemBox: Тоглогч эд зүйлсийн хайрцагтай харьцдаг.

-Player -> ScreenFade: Тоглогчийн үйлдэл нь дэлгэцийн бүдгэрэлттэй холбогдоно.

-Player -> World: Тоглогч дэлхийн орчинд байрладаг.

-Bullet -> World: Сум дэлхийн орчинд байрладаг.

-Button -> World: Товч дэлхийн орчинд байрладаг.

-HealthBar -> World: Эрүүл мэндийн түвшин дэлхийн орчинд байрладаг.

-ItemBox -> World: Эд зүйлсийн хайрцаг дэлхийн орчинд байрладаг.

-ScreenFade -> World: Дэлгэцийн бүдгэрэлт дэлхийн орчинд байрладаг.



Зураг 4: Класс диаграмм

3.4 Дэлгэцүүдийн зохиомж (Prototype)

Дэлгэцүүдийн зохиомж нь тоглоомын интерфэйсийг хэрхэн харагдахыг дүрсэлдэг. Үүнд эхлэл дэлгэц, тоглоомын дэлгэц, тоглоом дууссан дэлгэц зэрэг орно.

1. Эхлэх дэлгэц
 - "Эхлэх"товчлуур
 - "Гарах"товчлуур
2. Тоглоомын үндсэн дэлгэц
 - Тоглогч, дайсан, объектууд
 - Тоглогчийн амь болон сумны тоо
 - Арын зураг
3. Тоглоом дуусгах дэлгэц
 - "Дахин эхлэх"товчлуур
 - "Гарах"товчлуур

3.5 Үндсэн функцуудын псевдо алгоритмууд

Доорх псевдо код нь тоглоомын үндсэн функцуудыг хэрхэн хэрэгжүүлэхийг харуулна.

1.Тоглоом эхлүүлэх

def main.py():

- тоглоомын тохиргоо эхлүүлэх

while run:

- FPS тохируулах

хэрэв тоглоом эхлээгүй бол:

- цэсний дэлгэц харуулах

хэрэв тоглоом эхэлсэн бол:

- үндсэн дэлгэц харуулах

- тоглогч болон дайснуудыг шинэчлэх

- үйл явдлыг шалгах

2.Тоглогчийн хөдөлгөөн

хэрэв зүүн товч дарагдсан бол:

- зүүн тийш хөдөлгөөн

хэрэв баруун товч дарагдсан бол:

- баруун тийш хөдөлгөөн
- хэрэв тоглогч үсрэх товч дарагдсан бол:
- үсрэх
- таталцал нөлөө үзүүлэх
- саад тотгортой мөргөлдөхийг шалгах
- дэлгэцийн хязгаарт хүрсэн эсэхийг шалгах
- түвшин дууссан эсэхийг шалгах

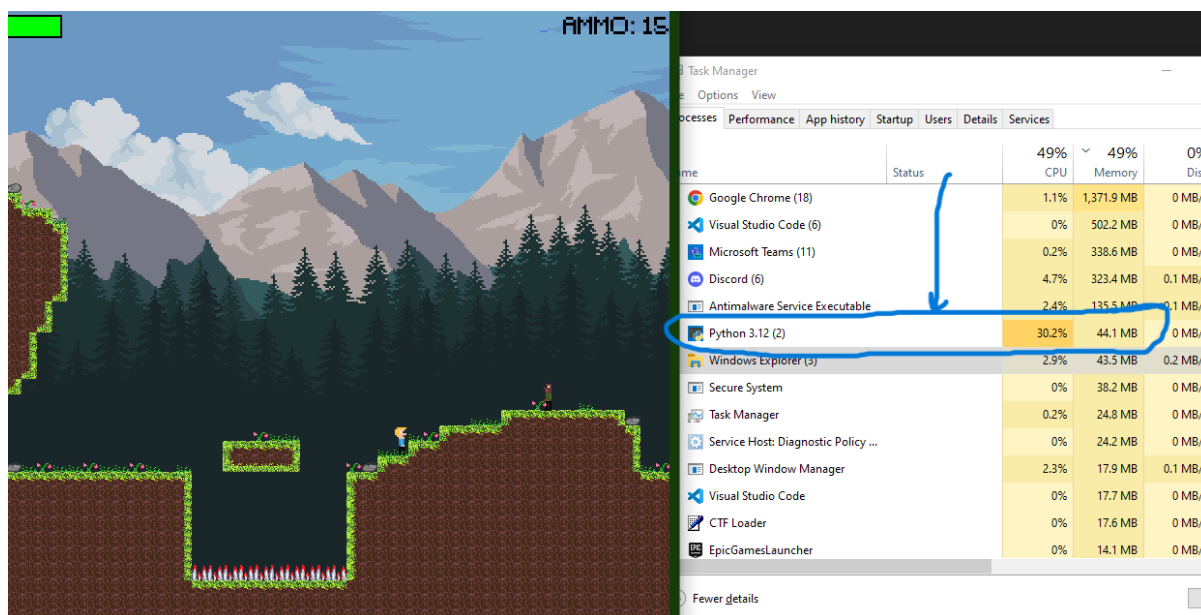
3.Дайсан AI

- тоглогчийн байрлалыг шалгах
- хэрэв тоглогч ойрхон бол:
- тоглогч руу довтлох
- хөдөлгөөн хийх
- саад тотгортой мөргөлдөхийг шалгах
- таталцал нөлөө үзүүлэх

Эдгээр псевдо алгоритмууд нь тоглогчийн үсрэлт, мөргөлдөөн мэдрэх, дайсны хөдөлгөөнийг хэрхэн зохицуулахыг харуулна.

4 Үр дүн

Тоглоомыг дуусаад ажиллуулж үзсэн байдал:



Зураг 5: Үр дүн

Тоглоомын ажиллуулж үзэхэд шуурхай санах ойн хэрэглээ хэвийн байсан ба CPU-ний хэрэглээ төсөөлснөөс өндөр байсан.

5 Дүгнэлт

- Тоглоомоо бүх юмыг өөрөө хийхээр шийдсэн тул дан код бичих бус хажуугаар нь pixel art хийдэг Aseprite гэх программ хэрэглэж сурсан.
- Мөргөлдөөн мэдрэх хийхдээ тоглогчийн дүр нь газартайгаа нийлж хөдлөхгүй болох асуудал гарсан ба үүнийг газраасаа бага зэрэг өндөрт мэдэрдэг болгосноор асуудлыг шийдсэн.
- Хийсэн asset нууд дэлгэцэн дээр маш жижиг хэмжээтэй гарч ирж байсан ба `pygame.transform.scale()` гэх method оор хэмжээг нь томруулж болдгийг олж мэдсэн.
- Parallax scroll алгоритм нь хамгийн энгийн харагдаж байсан ч хамгийн их асуудал дагуулж байсан. Тоглогчийг мар наас авч шидэх, удирдлагагүй нэг талруугаа урсах гэх мэт асуудлууд гарч ирсэн.
- Тоглоомыг хийж дууссаны дараа тоглоом нь аварга том координатын систем гэдгийг ойлгосон. Тиймээс тодорхой хэмжээний математикийн мэдлэгтэй байвал зохино. Мөн тоглоом хийх нь дан код бичих бус хажуугаар нь дүрээ бүтээх, ойр орчноо тохижуулах ямар нэгэн утга учир нэмэх гээд ард нь зөндөө асуудлууд байдгыг мэдсэн. Хажуугаар нь pixel art хийх, тоглогчийн хөдөлгөөнийг харуулсан animation хийх анхан шатны ур чадвар эзэмшсэн.

Ашигласан ном

- [1] D. A. de Oliveira Neto, “Intelligent 2d collision and pixel perfect precision.”