Izrada video igre koristeći GameMaker Studio 2

Ljutak, Ivan

Sveučilište Sjever

Završni rad br. 745/MM/2021

Izrada video igre koristeći GameMaker Studio 2

Ivan Ljutak, 2896/336

Varaždin, rujan 2021. godine

Sveučilište Sjever

Odjel za Multimediju, oblikovanje i primjenu

Završni rad br. 745/MM/2021

Izrada video igre koristeći GameMaker Studio 2

Student

Ivan Ljutak, 2896/336

Mentor

doc.dr.sc. Andrija Bernik

Varaždin, rujan 2021. godine

Sveučilište Sjever
Sveučilišni centar Varaždin
104. brigade 3, HR–42000 Varaždin

UNIVERSITY NORTH

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL: Odjel za multimediju

STUDIJ: preddiplomski stručni studij Multimedija, oblikovanje i primjena

PRISTUPNIK: Ivan Ljutak | MATIČNI BROJ: 2896/336

DATUM: 21.07.2021. | KOLEGIJ: Računalna animacija

NASLOV RADA: Izrada video igre koristeći GameMaker Studio 2

NASLOV RADA NA ENGL. JEZIKU: Creating a video game using GameMaker Studio 2

MENTOR: doc.dr.sc. Andrija Bernik | ZVANJE: Docent

ČLANOVI POVJERENSTVA:

1. mr.sc. Dragan Matković, v. pred. - predsjednik
2. pred. Nikolina Bolčević Horvatić, dipl.ing. - član
3. doc.dr.sc.Andrija Bernik - mentor
4. doc.art. Robert Geček - zamjenski član
5.

VŽ KC

MMI

Zadatak završnog rada

BROJ: 745/MM/2021

OPIS:

Cilj završnog rada je izrada video igre uz pomoć GameMaker Studio 2 engine-a.

GameMaker Studio 2 popularan je engine za izradu 2d igara. Omogućava izradu igara za razne popularne računalne i mobilne operacijske sustave, te i za HTML5 i najpopularnije igraće konzole. Njegove glavne prednosti su pristupačnost za početnike te vrlo brzi workflow u usporedbi sa drugim sličnim proizvodima. Tu brzinu postiže sa vrlo intuitivnim sučeljem i jednostavnim procesom uvoženja i uređivanja asseta, a najviše sa svojim programskim jezikom - GameMaker Language - čija je sintaksa dizajnirana za brzo i lagano programiranje igara.

U završnom radu će na primjeru izrade jedne igre koja će biti priložena uz pisani dio rada biti objašnjen proces programiranja raznih aspekata igre u GameMaker-u kao što su lik kojeg kontrolira igrač, neprijatelji, UI elementi, i slično. Također će se objasniti izrada i implementacija animiranih 2d sprite-ova u programu Aseprite, te razni aspekti stvaranja video igre kao što su dizajniranje korisničkog sučelja i gameplay-a.

ZADATAK URUČEN | POTPIS MENTORA

SVEUČILIŠTE SJEVER

Predgovor

Temu ovog rada odabrao sam jer već duže vrijeme pokušavam izraditi video igru kakva bi se meni svidjela. Primarni cilj mi je bio poboljšavanje sposobnosti dizajniranja video igara proučavajući njihove razne elemente i što ih čini efektivnim.

Zahvaljujem mentoru doc.dr.sc. Andriji Berniku koji mi je pomogao pri izradi završnog rada.

Također se zahvaljujem kolegici Sari Lauri Rokić koja mi je pomogla sa nekim dijelovima rada.

Sažetak

U ovom je radu prikazana izrada video igre u GameMaker Studio 2 engine-u, te izrada 2d sprite-ova u alatu Aseprite. Objašnjen je workflow u GameMaker Studio 2 i Aseprite-u, te se zatim detaljnije objašnjava proces izrade same igre. Prvo se objašnjava zamišljeni koncept igre, pa se onda prolazi kroz proces stvaranja lika kojeg igrač kontrolira. To uključuje i vizualni dizajn i objašnjenja svih akcija koje je lik sposoban izvesti. Prilaže se i kod koji omogućuje izvođenje tih akcija, napisan u GameMaker Language-u. Nakon svega toga se na isti način pokriva i neprijatelj, od vizualnog dizajna do akcija te kako je sve to implementirano u igru. Poslije toga, prolazi se kroz proces dizajniranja i implementiranja korisničkog sučelja u igri, ovdje se pokriva većina elemenata korisničkog sučelja i njihova svrha. Onda se prelazi na objašnjavanje training stage-ova, čija je svrha da nauče igrača osnovnim mehanikama igre, kako bi bio spreman za battle stage-ove koji dolaze nakon. U radu se zatim objašnjava i kako su dizajnirani te kako funkcioniraju battle stage-ovi.

Ključne riječi: Aseprite, Dizajniranje video igara, Game engine, GameMaker Studio, Pixel art, Sprite, Video igra

Summary

This paper showcases the process of creating a video game using the GameMaker Studio 2 engine, and the making of 2d sprites within Aseprite. A more detailed process of the creation of the game is explained after a summary of the workflow within GameMaker Studio 2 and Aseprite. The imagined game concept is explained first, after which the paper describes the way the playable character was designed. This includes both visual design and an explanation of the gameplay mechanics relating to the character. The code for these mechanics, written in GameMaker Language, is also shown. After all that the process of creating the game's enemy character is covered in the same way, from visual design to how it was implemented into the game. Then the process of designing and implementing the UI is covered, this includes most UI elements in the game and their purpose. The training stages are then explained, the purpose of which is to teach the players about the mechanics present in the game. Finally, the way the battle stages were designed and how they work is shown.

Key words: Aseprite, Video game design, Game engine, GameMaker Studio, Pixel art, Sprite, Video game

Sadržaj

1. Uvod

Cilj ovog rada izrada je računalne igre unutar GameMaker Studio 2 alata. Sve grafičke komponente igre osim jednog fonta izrađene su specifično za ovu igru, i njihova je izrada (korištenjem alata Aseprite) također objašnjena u ovom radu.

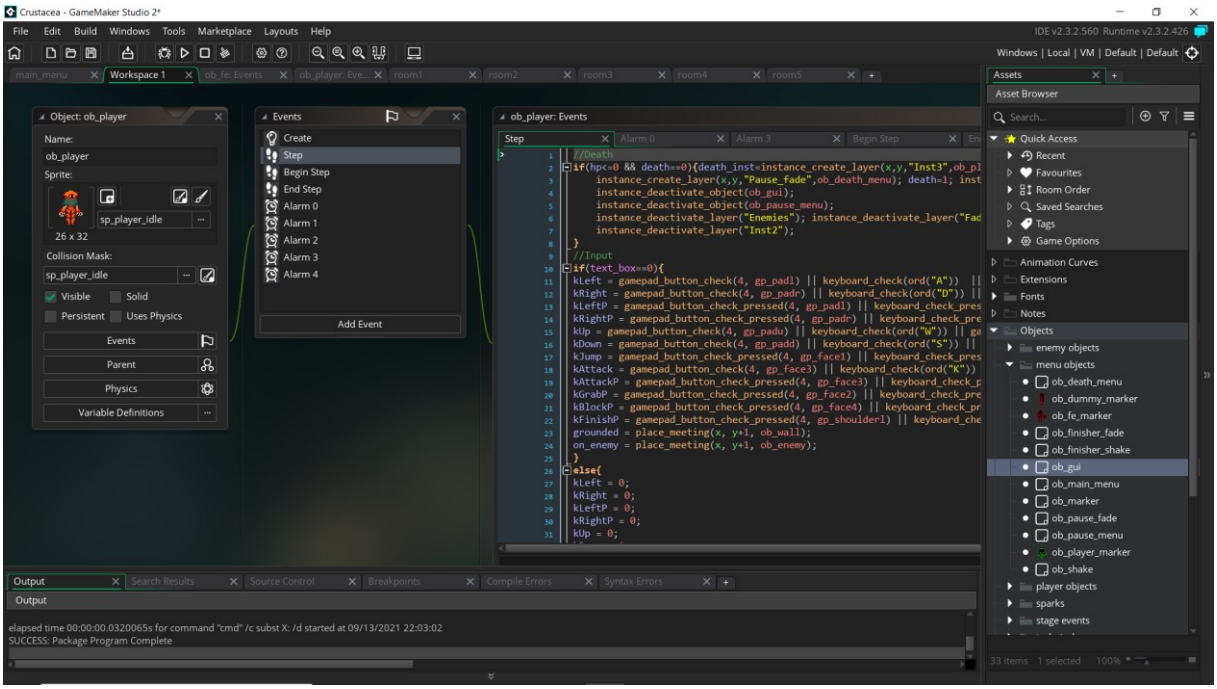
Izrada video igre odabrana je iz osobnih interesa autora. Poanta rada nije bila bolje se naučiti koristiti navedene alate, već poboljšati sposobnost dizajniranja video igara. U odabiru tipa 2d igre razmatralo se koji elementi igara su autoru najzabavniji. Donesen je zaključak da su to sistemi borbe u akcijskim igrama kao što su Bayonetta, Devil May Cry ili God Hand. Tako je za pristup dizajniranja igre fokus pretežno bio na sistemu borbe i interakcijama igrača i neprijateljskog AI-a u igri.

Proces dizajniranja sistema borbe sastojao se od nekoliko koraka. Prvo bi se smislila jedna mehanika, to jest akcija koju bi igrač mogao izvesti. Zatim bi se u Aseprite-u napravili sprite-ovi potrebni za ostvarenje te mehanike u igri. Nakon toga bi se sprite-ovi uvezli u projekt igre u GameMaker-u te bi se tamo isprogramirao potreban kod. Na kraju bi se podesili razni detalji kao brzina izvođenja animacija, veličina hitbox-a i slično, dok rezultati nisu bili zadovoljavajući. Onda se taj proces ponavlja imajući na umu kako će nova mehanika djelovati u kombinaciji sa ostalima.

2. GamerMaker Studio 2

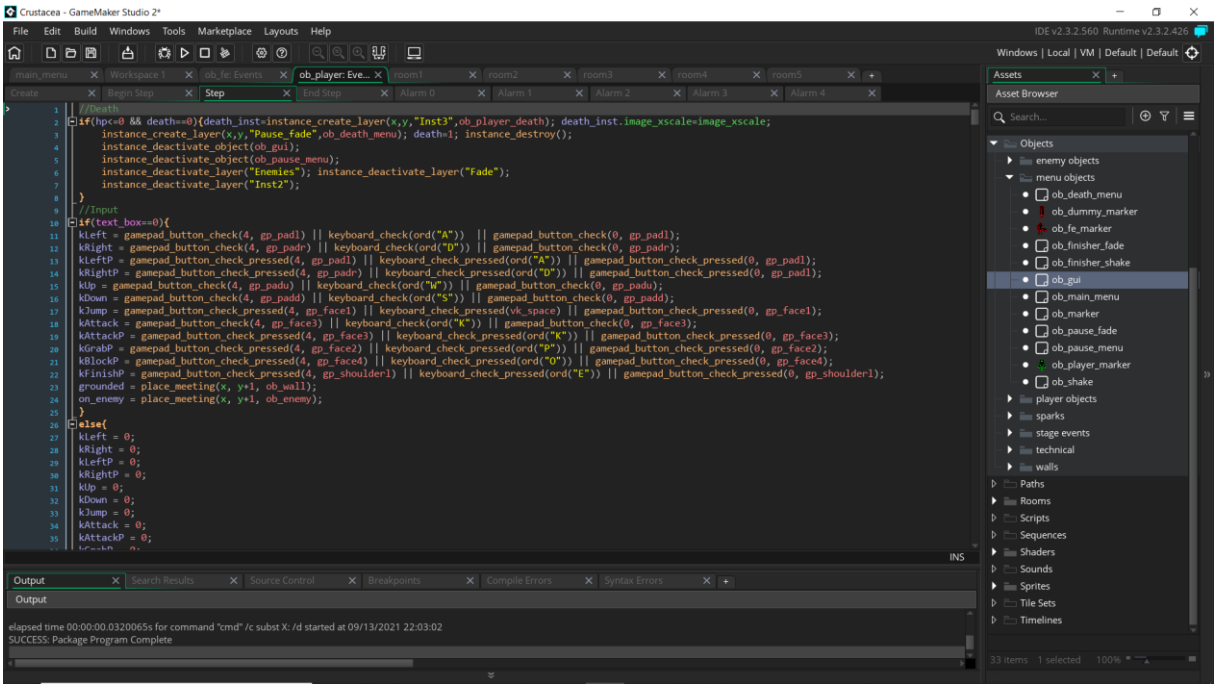
GameMaker Studio 2 popularan je engine za izradu 2d igara koji već dugi niz godina razvija YoYo Games. Podržava razvoj igra za brojne platforme, kao što su Windows, macOS, Ubuntu, Android, iOS, tvOS, fireTV, Android TV, Microsoft UWP, HTML5, PS4, PS5, Xbox One i Xbox Series X|S. [1]

Iako podržava izradu samo 2d igara, GameMaker ima prednost nad sličnim proizvodima u svom brzom workflow-u kojeg mu omogućava jednostavno i intuitivno sučelje kombinirano s originalnim programskim jezikom GameMaker Language.



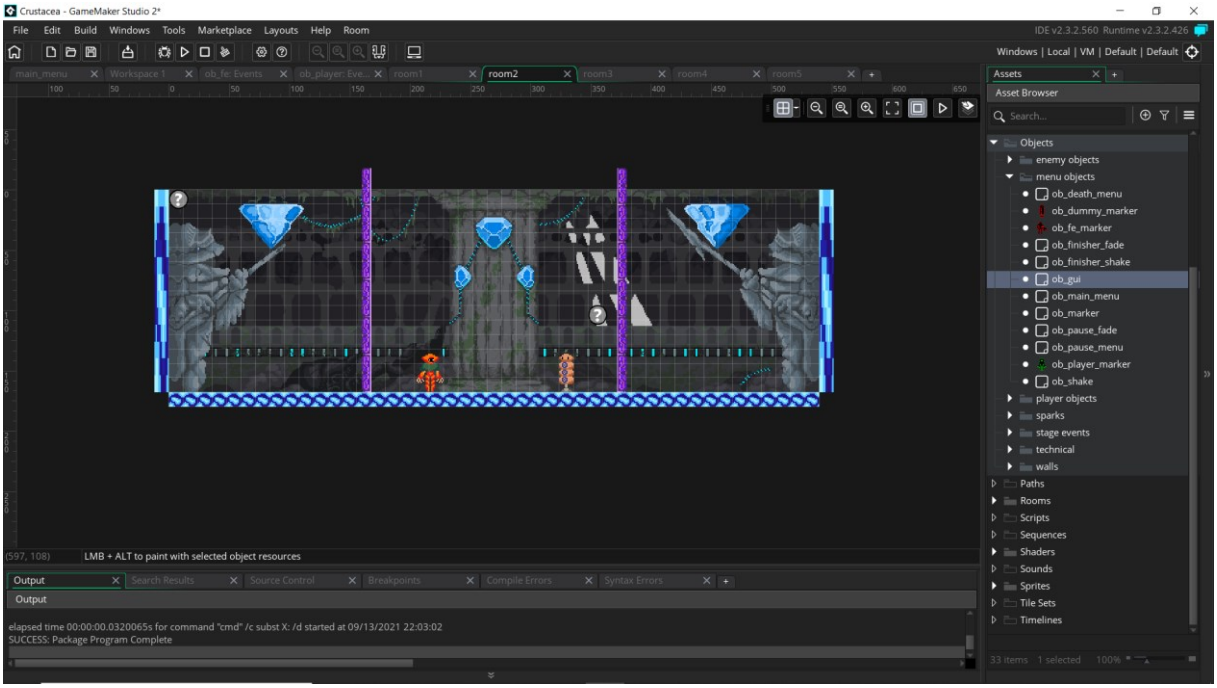
Slika 2.1. GameMaker Studio 2 workspace

Osnovni elementi prostora za izradu video igara u GameMaker Studio 2 su asset browser, workspace, events i room editor. Asset browser prozor je u kojeg se slažu svi resursi za igru, kao što su objekti, sprite-ovi, zvukovi, fontovi, skripte, sobe, shader-i i druge stvari. Asset browser ima unaprijed određene mape za te resurse, ali korisnik može brisati i preimenovati mape kako god želi, te može bilo koji tip resursa stavljati u bilo koju mapu po želji i potrebi. Također postoji i mogućnost izrade podmapa. Workspace je prostor u kojem se resursi iz asset browser-a otvaraju te se tu mogu uređivati svi njihovi aspekti. Na primjer, tu se sprite-ovima može mijenjati brzina izvođenja animacije te broj i redoslijed frame-ova, ili se može uređivati kod objekta.



Slika 2.2. GameMaker Studio 2 events

Events je prozor u kojem se mogu odabirati u uređivati kodovi objekata. Ovisno o tipu event-a, kod se izvodi u drugačijem trenutku igre. Prednost uređivanja koda u ovom prozoru u odnosu na uređivanje koda u workspace-u je što je ovdje kod prikazan preko cijelog ekrana i manje je stvari oko prozora koda koje bi mogle smetati korisniku.



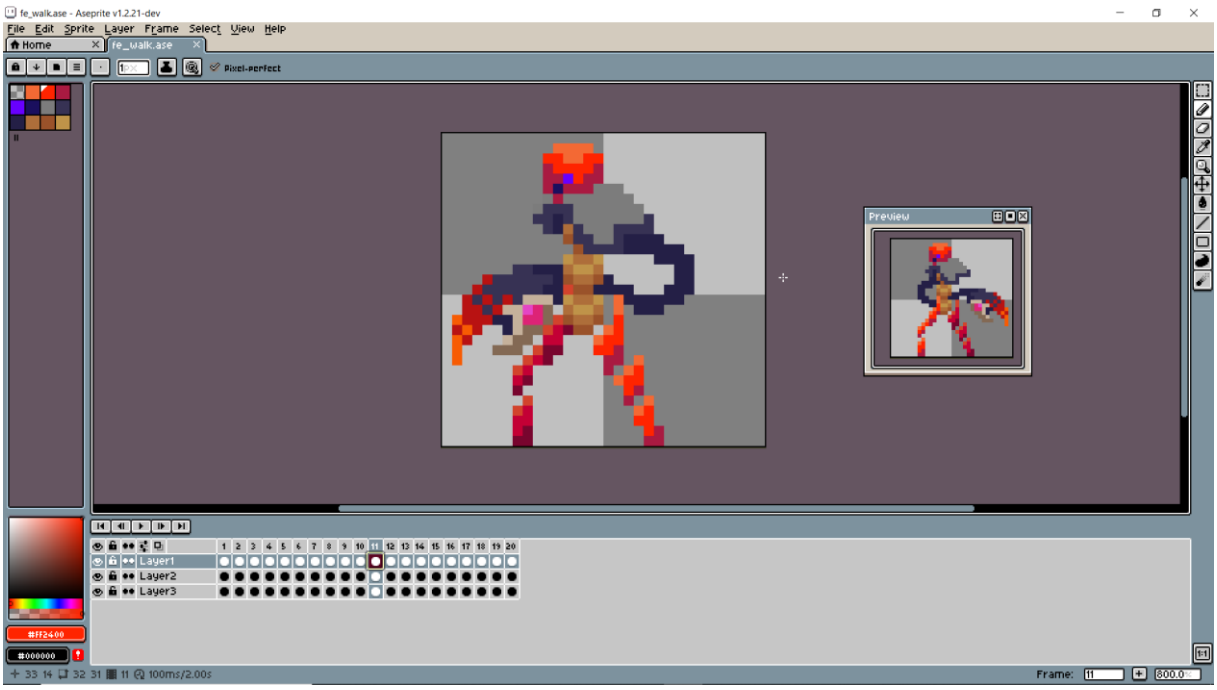
Slika 2.3. GameMaker Studio 2 room editor

Room editor u GameMaker Studio 2 je fantastičan alat za uređivanje prostora u kojem se gameplay igara odvija i jedno je od najboljih poboljšanja u odnosu na starije verzije GameMaker-a čiji je room editor bio dosta nepraktičan za upotrebu. Ovaj room editor ima odličan performans, intuitivan način postavljanja objekata i kretanja po prostoru, layer sistem, i još mnogo korisnih opcija koje sve rade besprijekorno.

3. Aseprite

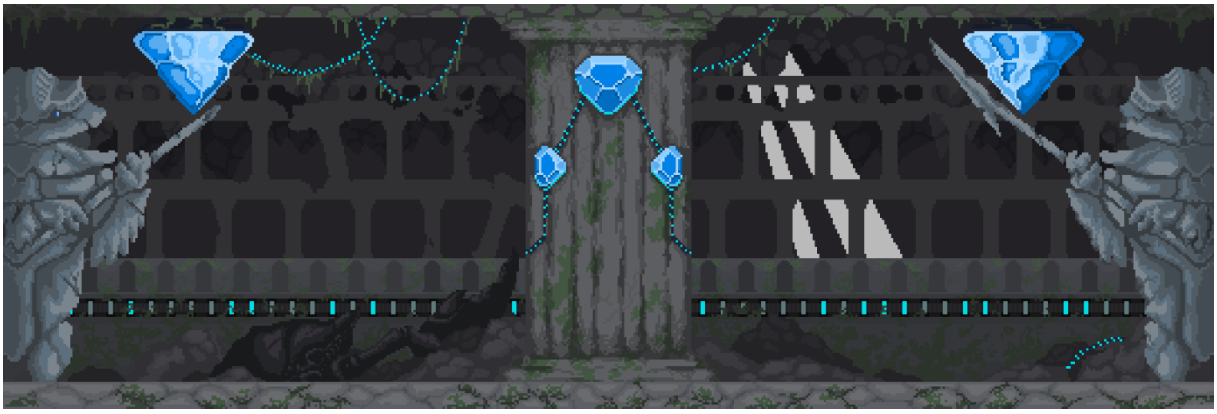
Aseprite je program za izradu 2d grafika niskih rezolucija. Namijenjen je za stvaranje sprite-ova, pixel art-a i grafika u stilu retro video igara. Već godinama ga razvija David Capello sa svojom kompanijom Igara Studio. [2, 3, 4]

Veliki broj sprite-ova i ostalih grafika u ovoj igri napravljeni su u Aseprite-u, iako su neki modificirani u programu Piskel, a nekoliko ih je u potpunosti izrađeno u Piskel-u. To je zato što, iako Aseprite ima više mogućnosti, za jednostavnije je stvari Piskel brži.



Slika 3.1. Aseprite workspace

Iako ima mnogo opcija i shortcut-ova, Aseprite ima dosta standardan i intuitivan workflow. Najbitniji elementi su sprite editor, timeline, color bar i preview. Sprite editor je prostor u kojem se uređuje sprite. Na njemu se može sa raznim alatima crtati klikom miša ili stylus-om ako imate crtaći tablet, te se vrlo lako i brzo može zumirati kotačićem miša. Klikom na kotačić se može i pomicati prostor. Timeline je sličan kao i većina timeline-ova u ovakvim programima, prikazuje sve frame-ove i layer-e koji se nalaze u animaciji/slici. Na timeline-u se također mogu pomicati, brisati, kopirati ili duplicirati layer-i i frame-ovi. Postoji i još mnogo mogućnosti prikaza i zaključavanja layer-a. Na color baru su prikazane sve boje prisutne na sprite-u, te se tu odabiru boje piksela koji se postavljaju lijevim klikom i boje koje se postavljaju sa desnim klikom. Također se mogu birati boje klikom na spektar boja ili unošenjem heksadecimalnih ili rgba vrijednosti. Preview je mali prozorčić koji se može lako prikazati ili sakriti pritiskom tipke F7. Veličina i položaj prozorčića te zoom na prostor u njemu se mogu lako podesiti. Pritiskom play tipke na prozorčiću ili kombinacijom tipki shift i enter prikazat će se kako animacija izgleda kad se kreće.



Slika 3.2. Pozadina stage-a u igri

Također se mora spomenuti da je u izradi nekih grafika pomogla kolegica Sara Laura Rokić. Jedna od tih grafika je pozadina koja se vidi na slici iznad.

4. Izrada igre

4.1. Koncept

Originalna ideja ove igre bila je dizajnirati combat sistem i napraviti nekoliko stage-ova u kojim bi se taj sistem mogao prikazati. U autorovim prijašnjim pokušajima dizajniranja video igara bile su osmišljene jednostavne mehanike lika kojeg se kontrolira, te su se nakon toga pokušali graditi stage-ovi i druge komponente igre oko tih mehanika. Međutim, te igre nikad nisu ispale dovoljno zabavnim za ukus autora. Zato je ovaj put odabran pristup u kojem bi se prvobitno fokusiralo isključivo na combat sistem – interakcije između lika kojeg igrač kontrolira i neprijatelja.

4.2. Player

Prvobitna zamisao za dizajn glavnog lika bila je humanoidno stvorenje s osobinama raka koje bi koristilo svoja kliješta za napadanje neprijatelja. Međutim, s vremenom se odlučilo za malo apstraktniji dizajn. Iako i dalje ima neke manje prepoznatljive karakteristike rakova, finalna verzija lika nema kliješta već fleksibilne prste čiji se oblik može mijenjati prilikom napadanja, iako se najčešće promjeni u oblik kliješta. Odabrao se ovakav dizajn kako bi se omogućilo više kreativnosti pri smišljanjem animacija napada.

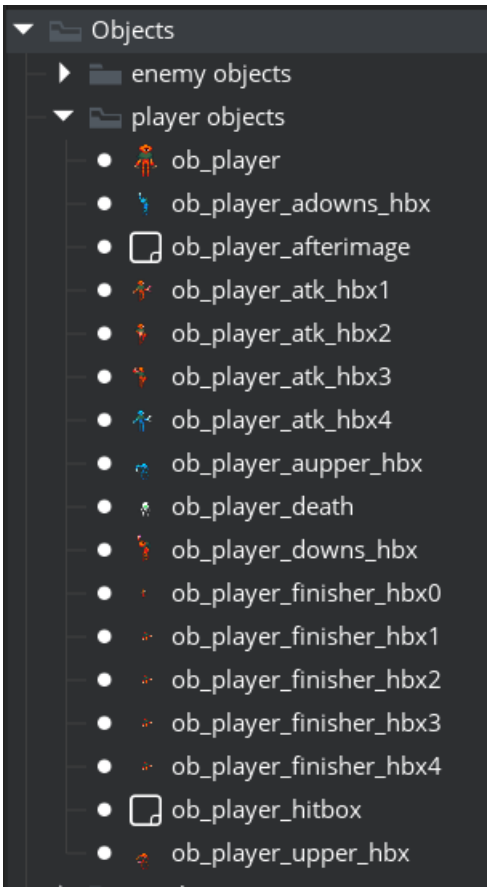
Slika 4.2.1. Dizajn lika u igri

```
//Input
if(text_box==0){
kLeft = gamepad_button_check(4, gp_padl) || keyboard_check(ord("A"))  || gamepad_button_check(0, gp_padl);
kRight = gamepad_button_check(4, gp_padr) || keyboard_check(ord("D")) || gamepad_button_check(0, gp_padr);
kLeftP = gamepad_button_check_pressed(4, gp_padl) || keyboard_check_pressed(ord("A")) || gamepad_button_check_pressed(0, gp_padl);
kRightP = gamepad_button_check_pressed(4, gp_padr) || keyboard_check_pressed(ord("D")) || gamepad_button_check_pressed(0, gp_padl);
kUp = gamepad_button_check(4, gp_padu) || keyboard_check(ord("W")) || gamepad_button_check(0, gp_padu);
kDown = gamepad_button_check(4, gp_padd) || keyboard_check(ord("S")) || gamepad_button_check(0, gp_padd);
kJump = gamepad_button_check_pressed(4, gp_face1) || keyboard_check_pressed(vk_space) || gamepad_button_check_pressed(0, gp_face1);
kAttack = gamepad_button_check(4, gp_face3) || keyboard_check(ord("K")) || gamepad_button_check(0, gp_face3);
kAttackP = gamepad_button_check_pressed(4, gp_face3) || keyboard_check_pressed(ord("K")) || gamepad_button_check_pressed(0, gp_face3);
kGrabP = gamepad_button_check_pressed(4, gp_face2) || keyboard_check_pressed(ord("P")) || gamepad_button_check_pressed(0, gp_face2);
kBlockP = gamepad_button_check_pressed(4, gp_face4) || keyboard_check_pressed(ord("O")) || gamepad_button_check_pressed(0, gp_face4);
kFinishP = gamepad_button_check_pressed(4, gp_shoulderl) || keyboard_check_pressed(ord("E")) || gamepad_button_check_pressed(0, gp_shoulderl);
grounded = place_meeting(x, y+1, ob_wall);
on_enemy = place_meeting(x, y+1, ob_enemy);
}
else{
kLeft = 0;
kRight = 0;
kLeftP = 0;
kRightP = 0;
kUp = 0;
kDown = 0;
kJump = 0;
kAttack = 0;
kAttackP = 0;
kGrabP = 0;
kBlockP = 0;
kFinishP = 0;
grounded = 0;
on_enemy = 0;
```

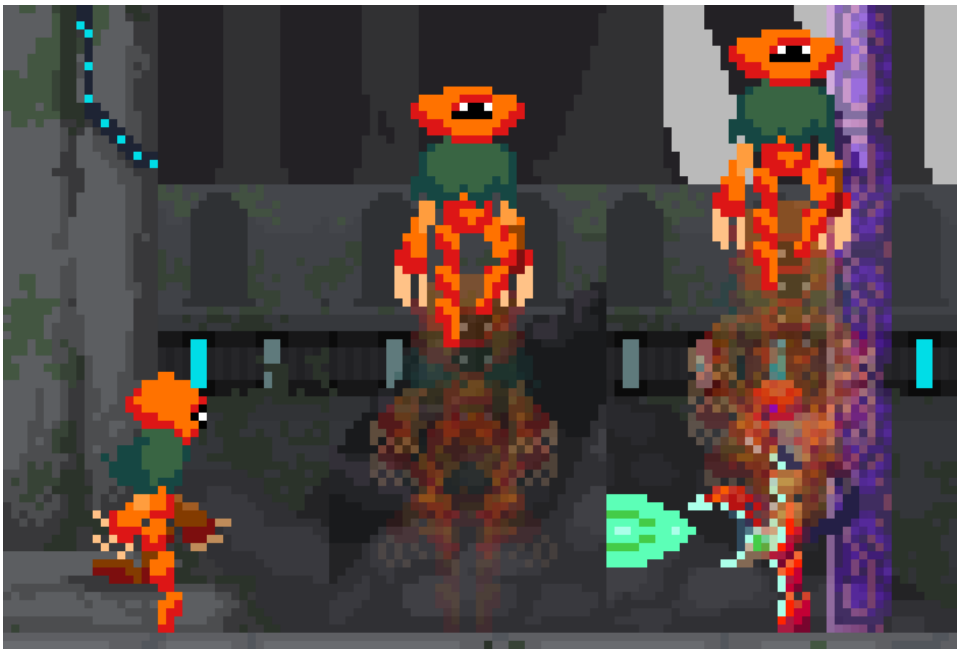
Slika 4.2.2 Kod za input u igri

Lik može trčati u lijevom i desnom smjeru, skakati, izbjegavati neprijateljske napade sa brzim dash potezom, blokirati neprijateljske napade, te ima 4 drukčija tipa napada. Prvi je normalni tip napada – to su samo 3 obična napada koji se ponavljaju. Slabo su efektivni što se tiče koliko štete rade neprijateljima. Drugi tip je guard break – to su dva napada sa kojima se može prekinuti neprijateljsko blokiranje. Ako neprijatelj blokira visoko, efektivan je visoki guard break. Ako blokira nisko, efektivan je niski guard break. U slučaju uspješnog guard break napada neprijatelj će preći u

posebno stun stanje, te će se žute zvijezde stvoriti iznad njega. Treći tip napada je boosted napad – snažniji napad koji je efektivan u više situacija nego normalni, radi veću štetu neprijateljima i stavlja neprijatelje u kraći stun stanje poput guard break-a. U ovom stanju će se iznad neprijatelja pojaviti plave zvijezde. Zadnji tip napada je finisher napad. Ovaj napad djeluje samo u specifičnim situacijama na neprijateljima čiji je health spušten na nulu.



Slika 4.2.3. Player objekti u igri

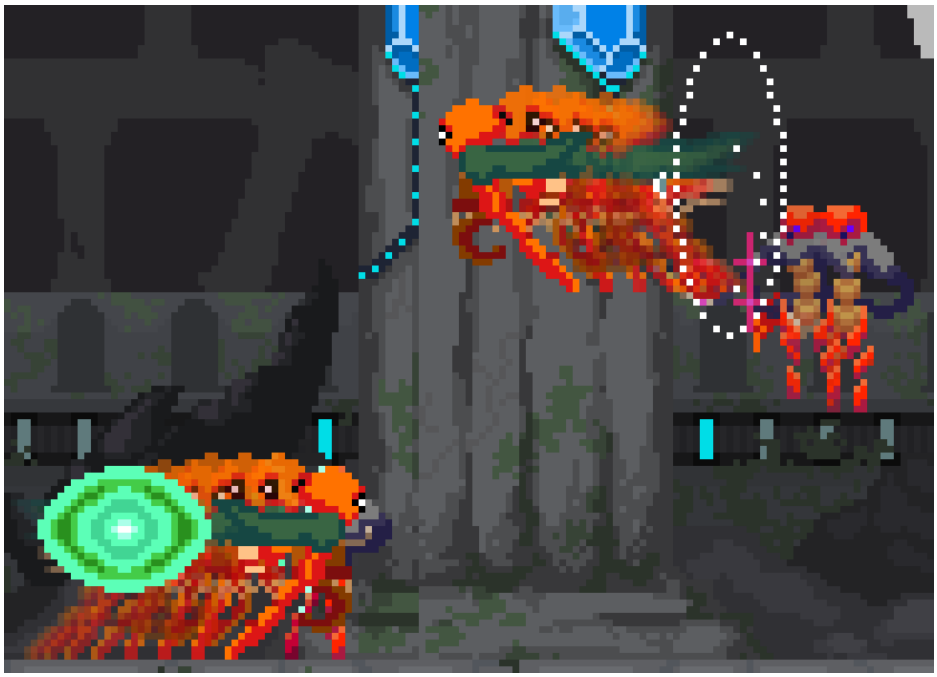


Slika 4.2.4. Kretanje lika

Trčanje počinje pritiskom na lijevu ili desnu tipku dok je lik na tlu. Brzina je isprva malena, ali za trenutak se popne na veću konstantnu brzinu. Ovo je napravljeno kako igrač ne bi mogao naglo izmijeniti smjer trčanja bez promjene brzine, te kako bi mu bilo lakše promijeniti položaj lika ako ga želi pomaknuti za malen broj piksela. Trčanje i skakanje normalno se mogu izvesti samo na tlu, ali lik može i odskočiti od neprijatelja ako se tipka za skok pritisne u trenutku u kojem su lik u neprijatelj u kontaktu a lik je u zraku.

```
//Movement
if(atk<=0 && dashr!=0 && dashl!=0 && hurt==0){
var move = kRight - kLeft;

if(next_frame_hsp<6){
    hsp = move;
    next_frame_hsp++;
}
else if(next_frame_hsp<12){
    hsp = move*2;
    next_frame_hsp++;
}
else{
    hsp = move * walksp;
}
if(kLeftP || kRightP){next_frame_hsp=0;hsp=0;}
if(kLeft && kRight){next_frame_hsp=0;hsp=0;}
if(!(kLeft||kRight)){next_frame_hsp=0;}

if(next_frame==4){
    vsp = vsp + grv;
    next_frame--;
}
else if(next_frame==1){
    next_frame=4;
}
else {
    next_frame--;
}

if((grounded || on_enemy) && kJump){
    next_frame = 4;
    vsp = -5;
    image_index = 0;
    shadow = instance_create_layer(x,y,"Inst2",ob_player_afterimage);
    shadow.sprite_index = sprite_index;
    shadow.image_index = image_index;
    shadow.image_xscale = image_xscale;
}
}
```

Slika 4.2.5. Kod za kretanje lika

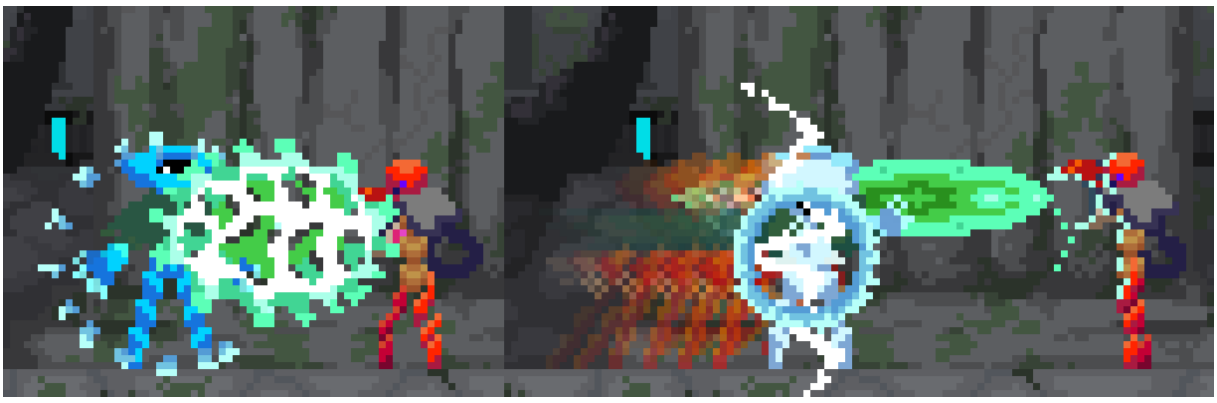


Slika 4.2.6. Dash potez lika

Dash je potez koji se može izvesti brzim dvostrukim pritiskom tipke lijevo ili desno. Tijekom njega je lik u potpunosti nepobjediv. Uobičajeno se može izvesti samo na tlu, ali ako neprijatelj stavi lika u posebno stun stanje blokiranjem (više o tom poslije) igrač može izvesti dash kako bi izašao iz tog stanja rano, a to se može napraviti i u zraku.

```
//Dashing
if(hurt==0 || hurt==56){

if(kRightP && dashr == 1 && (grounded || hurt==56) && vsp==0 && atk<=0){
    next_frame = 4;
    next_frame_hsp = 0;
    hsp=0;
    dashr--;
    atk=-1;
    hurt = 0;
    image_index=0;
    sprite_index=sp_player_dash;
    image_xscale=1;
}
if(kRightP && dashr == 2 && (grounded || hurt==56) && vsp==0 && atk<=0 && dashl==2){dashr--; alarm[1] = 16;}

if(dashr==0){
    if(image_index>5){dashr=2; atk=0;}
    if(hsp!=5){hsp++;}
    shadow = instance_create_layer(x,y,"Inst2",ob_player_afterimage);
    shadow.sprite_index = sprite_index;
    shadow.image_index = image_index;
    shadow.image_xscale = image_xscale;
}


if(kLeftP && dashl == 1 && (grounded || hurt==56) && vsp==0 && atk<=0){
    next_frame = 4;
    next_frame_hsp = 0;
    hsp=0;
    dashl--;
    atk=-1;
    hurt = 0;
    image_index=0;
    sprite_index=sp_player_dash;
    image_xscale=-1;
}
if(kLeftP && dashl == 2 && (grounded || hurt==56) && vsp==0 && atk<=0 && dashr==2){dashl--; alarm[2] = 16;}

if(dashl==0){
    if(image_index>5){dashl=2; atk=0;}
    if(hsp!=-5){hsp--;}
    shadow = instance_create_layer(x,y,"Inst2",ob_player_afterimage);
    shadow.sprite_index = sprite_index;
```

Slika 4.2.7. Kod za dash lika



Slika 4.2.8. Blokiranje neprijateljskih napada

Pritiskom tipke za blokiranje, lik staje na mjestu i mijenja boju u plavu, te se nakon sekunde vraća u normalno stanje. U ovo stanje blokiranja lik može preći u gotovo bilo kojem trenutku, osim ako je u stun stanju. Ako se normalno blokira neprijateljski napad, lik će primiti jako malu štetu. Međutim, ako blokira taman prije nego što dođe u kontakt sa neprijateljskim napadom, neće primiti nikakvu štetu. To se zove savršeni blok, i to je bitna tehnika vezana uz boosted napade. Iako bi u normalnim uvjetima bilo teško izvesti savršeni blok, kombinacijom dasha i blokiranja dosta je lagano.

```
//Combat
if(hurt == 0){
if(kBlockP && atk!=99 && atk<=998 && atk!=-2 && atk!=13 && atk!=15 && atk!=18 && atk!=19){
    next_frame = 4;
    next_frame_hsp = 0;
    hsp=0;
    vsp=0;
    image_index=0;
    sprite_index=sp_player_block;
    atk=99;
    dashr=2;
    dashl=2;
    combo=0;
    charge=0;
    hitbox_aper=0;
    with(ob_player_hitbox){instance_destroy();}
    hb_active=0;
}
if(block==1){
    if(image_index<1.2){block=3; sprite_index=sp_player_ampd_block; image_index=0; ap+=0.45
        bspark = instance_create_layer(x,y+16,"Inst3",ob_ampd_block_spark);
        bspark.image_angle = random(360);
        instance_create_layer(x,y,"Fade",ob_shake);
        if(s_val==1){skill+=8;}
        else if(s_val==2){skill+=6;}
        else if(s_val==3){skill+=4;}
        else if(s_val==4){skill+=2;}}
    else{
        image_index=0;
        sprite_index=sp_player_blockd;
        block=2;
        ap+=0.075;
        hp--;
        bspark = instance_create_layer(x,y+16,"Inst3",ob_block_spark);
        bspark.image_angle = random(360);
        if(s_val==1){skill+=3;}
        else if(s_val==2){skill+=2;}
        else if(s_val==3){skill+=1;}
        else if(s_val==4){skill+=0.5;}
    }
    recovery = 1;
```

Slika 4.2.9. Kod za blokiranje

Slika 4.2.10. Normalni napadi

Obični napad ne radi veliku štetu neprijateljima, tako da ima druge primarne svrhe. Prva je tjeranje neprijatelja da počne blokirati kako bi se napravio guard break. Druge svrhe su udaranje neprijatelja tijekom guard break stun stanja i produljivanje napadnih kombinacija (više o tome poslije). Može se izvesti i na tlu i u zraku.

```
//Attacking
if(atk==1){
    if(image_index>0.8){atk=4; image_index=0; sprite_index=sp_player_attack1;charge=0;
            instance_create_layer(x, y, "Player", ob_player_atk_hbx1);}
}

if(atk==2){
    if(image_index>0.8){atk=5; image_index=0; sprite_index=sp_player_attack2;charge=0;
            instance_create_layer(x, y, "Player", ob_player_atk_hbx2);}
}

if(atk==3){
    if(image_index>0.8){atk=6; image_index=0; sprite_index=sp_player_attack3;charge=0;
            instance_create_layer(x, y, "Player", ob_player_atk_hbx3);}
```

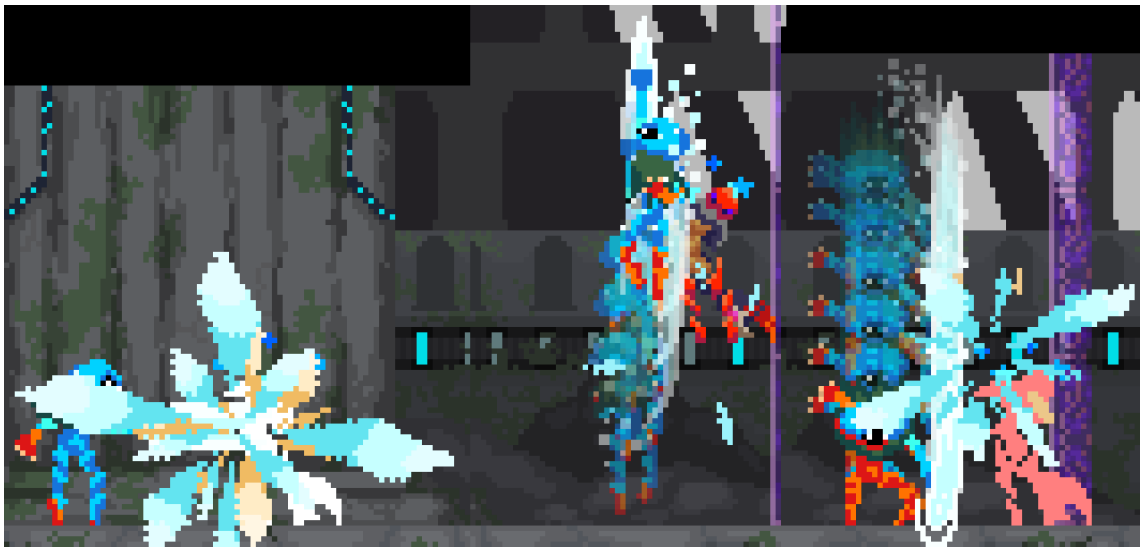
Slika 4.2.11. Kod normalnih napada

Slika 4.2.12. Guard break

Guard break ne služi samo za produljivanje napadnih kombinacija, već je i bitan faktor za bržu mogućnost izvođenja boosted napada. Može se izvesti i na tlu i u zraku.

```
if(atk==23){
    if(image_index>2.2 && kAttackP){combo = 1;}
    if(image_index>1 && hitbox_aper == 0){instance_create_layer(x, y, "Player", ob_player_upper_hbx); hitbox_aper=1;}
    if(image_index>4.6 && combo==1){
        image_index=0;
        sprite_index=sp_player_air_attack1;
        atk = 20;
        combo = 0;
        hitbox_aper=0;
    }
    if(image_index>5){atk=-1; alarm[0] = 16; hitbox_aper=0;}
    vsp=0;
}

if(atk==24){
    if(image_index>2.2 && kAttackP){combo = 1;}
    if(image_index>1 && hitbox_aper == 0){instance_create_layer(x, y, "Player", ob_player_downs_hbx); hitbox_aper=1;}
    if(image_index>4.6 && combo==1){
        image_index=0;
        sprite_index=sp_player_air_attack1;
        atk = 20;
        combo = 0;
        hitbox_aper=0;
    }
    if(image_index>5){atk=-1; alarm[0] = 16; hitbox_aper=0;}
    vsp=0;
}
```

Slika 4.2.13. Kod za guard break



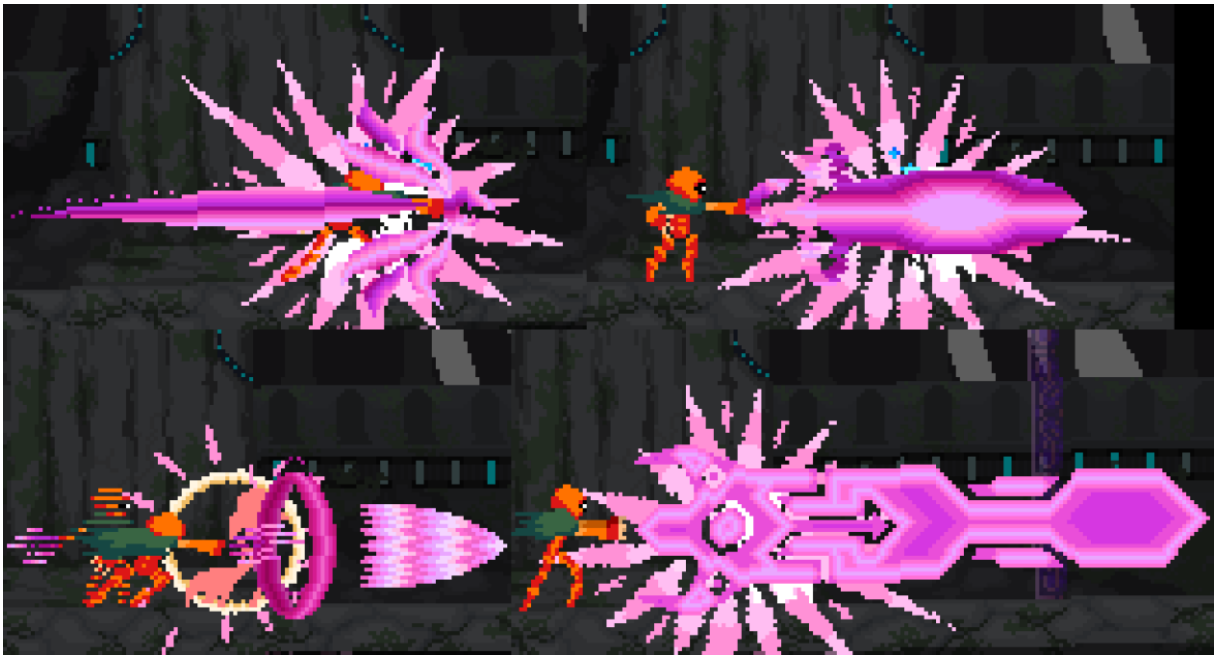
Slika 4.2.14. Boosted napadi

Boosted napadi mogu se izvesti samo pod uvjetom da je boost meter, koji se nalazi ispod health bar-a, napunjen barem jednu četvrtinu. Korištenje boost napada ispraznit će jednu četvrtinu boost meter-a. Boost meter se jako sporo puni kada se neprijatelj pogađa normalnim napadima, ili kada se blokiraju neprijateljski napadi. Međutim, može se brže puniti ako je neprijatelj u guard break stun-u dok ga se pogađa normalnim napadima, ili savršenim blokiranjem. Postoje 3 boost napada, prvi je obični boost napad kojeg igrač izvodi tako što pritisne tipku za napad dok je u stanju blokiranja. Ovaj boost napad pogađa ispred lika i ima vrlo dobar horizontalni domet. Može se izvesti i u zraku i na tlu. Drugi boost napad je boosted uppercut, koji

se unosi pritiskom tipke za napad u stanju blokiranja dok je pritsnuta tipka za gore. Ima manji horizontalni domet od običnog boost napada, ali može lansirati neprijatelje u zrak, te lika ne mogu pogoditi neprijateljski napadi tijekom kasnijeg dijela njegove animacije. Može se izvesti samo na tlu. Treći napad, silazni boosted napad, sličan je kao boosted uppercut, samo što spušta neprijatelja iz zraka na tlo te se može izvesti samo u zraku.

```
869 if(atk==99){
870     if(image_index>5){atk=-2; alarm[0] = 18;}
871     if(ap>=1){
872     if(kAttackP && grounded && !kUp){
873         image_index=0;
874         sprite_index=sp_player_ampd_attack;
875         atk = 13;
876         ap-=1;
877     }
878     if(kAttackP && grounded && kUp){
879         image_index=0;
880         sprite_index=sp_player_ampd_uppercut;
881         rising = 10;
882         atk = 15;
883         ap-=1;
884     }
885     if(kAttackP && !grounded && !kDown){
886         image_index=0;
887         sprite_index=sp_player_ampd_air_attack;
888         atk = 18;
889         ap-=1;
890     }
891     if(kAttackP && !grounded && kDown){
892         image_index=0;
893         sprite_index=sp_player_ampd_downstrike;
894         rising = 0;
895         atk = 19;
896         ap-=1;
897     }
898     }
899 }
```

Slika 4.2.15. Kod boosted napada



Slika 4.2.16. Finisher napadi

Finisher napadi mogu se izvesti samo u specifičnim uvjetima. Prvo, health neprijatelja mora biti 0. Zbog toga se finisher mora izvesti u zadnjoj napadnoj kombinaciji nad neprijateljem, što je drugi uvjet. To znači da se može izvesti dok god je neprijatelj u hit stun stanju, stun stanju, ili blokira. Treći uvjet je da finisher bar, koji se nalazi na lijevoj strani ekrana ispod ostala 2 meter-a, mora biti barem 25% napunjen, slično kao i boost meter. Međutim, kao što se vidi iz slike, postoje 4 drugačija finisher napada. Svaki dolazi zaredom nakon prijašnjeg, od LV1 do LV4. To znači da bi se izveo četvrti, neprijatelj se prije toga mora pogoditi sa prva tri napada. To može biti teško za izvesti zbog načina na koji se finisher bar puni (i prazni). Na nišim razinama ga relativno efektivno pune savršeni blok i normalni napadi. Međutim, kada je više napunjen zahtjeva boosted napade. No samo korištenje boosted napada nije dovoljno da dostigne maksimalnu razinu. Što je duža napadna kombinacija, to jest što je više normalnih napada iskorišteno tijekom nje, boosted napadi će više efektivno puniti finisher bar. Osim toga, korištenje dva ista boosted napada zaredom u jednog kombinaciji će smanjiti njihovu efektivnost punjenja. No, zbog kojeg bi razloga igrači uopće htjeli raditi finisher napade? Postoje 3 primarna razloga. Prvi je što izvođenje finisher napada puni health i boost meter. Svaki idući finisher u redu ih puni mnogo više nego prošli. Drugi razlog je činjenica da ako se ne izvede finisher,

neprijatelj će napasti lika još jednom prije nego što nestane. Treći je razlog to što finisher napadi donose ogroman broj bodova u battle stage-ovima, u kojima se igraču daje rank ovisno o gameplay-u. Bez korištenja finisher-a je gotovo nemoguće dobiti više od najnižeg rank-a, a najviši rank-ovi se ne mogu dobiti bez korištenja LV4 finisher napada.

```
if(kFinishP && skill>=20 && atk<999){
    instance_create_layer(x, y, "Player", ob_player_finisher_hbx0);
}

if(atk==998){
    next_frame = 4;
    next_frame_hsp = 0;
    hsp=0;
    vsp=0;
    image_index=0;
    sprite_index=sp_player_finisher1;
    atk=999;
    dashr=2;
    dashl=2;
    combo=0;
    charge=0;
    hitbox_aper=0;
    with(ob_player_hitbox){instance_destroy();}
    hb_active=0;
    skill-=20;
    instance_create_layer(x,y,"Fade",ob_finisher_fade);
}

if(atk==999){
    if(image_index>4 && kFinishP){
        combo=1;
    }
    if(image_index>3 && hitbox_aper==0){instance_create_layer(x, y, "Player", ob_player_finisher_hbx1); hitbox_aper=1;}
    if(image_index>8 && combo==1 && skill>=20){
        image_index=0;
        sprite_index=sp_player_finisher2;
        atk = 1000;
        combo = 0;
        skill-=20;
        hitbox_aper = 0;
    }
    if(image_index > 9){
        atk=0; hitbox_aper = 0; recovery=1; alarm[3]=10;
    }
}
if(atk==1000){
    if(image_index>4 && kFinishP){
```

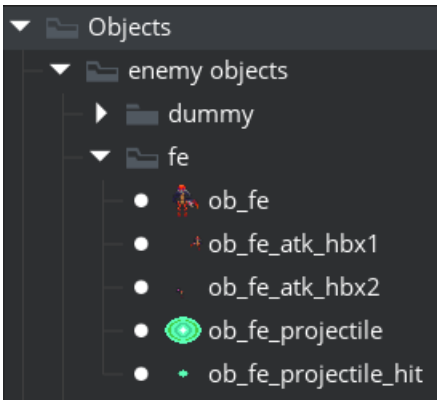
Slika 4.2.17. Kod finisher napada

4.3. Neprijatelji

Što se tiče izgleda, neprijatelje u igri planiralo se dizajnirati, poput lika kojeg kontrolira igrač, kao humanoidna stvorenja s osobinama raka. Razlika bi bila u tome što bi neprijatelji imali manje humanoidnih osobina i više osobina rakova. Kako se znalo da se želi da ovaj neprijatelj ima zračne napade i napade sa velikim dometom, jedna od prvih ideja koja se autoru svidjela bila je bazirati neprijatelje na rakovima iz

obitelji Ocypodidae, čiji mužjaci često imaju jedna kliješta mnogo veća od drugih. Tako je u finalnom dizajnu ostao taj aspekt korištenja samo jednog uda za napadanje. Međutim, dizajn je na kraju nekako ispao više vanzemaljski nego što je isprva planirano, ali se svidio autoru pa je bilo odlučeno zadržati ga.

Slika 4.3.1. Dizajn neprijatelja



Slika 4.3.2. Objekti neprijatelja

Neprijatelj može hodati i naprijed prema liku i unazad kako bi se udaljio od njega. To radi kako bi se pozicionirao za jedan od svoja dva primarna napada – projektil čije je vrijeme od početka animacije do stvaranja hitbox-a vrlo kratko, te laser koji je mnogo sporiji ali radi veću štetu i duže traje. Projektil teoretski ima veći domet, ali laser u trenutku stvaranja pokriva veći prostor. Zbog ove razlike u tempiranju napada igrač ne može lagano vizualno reagirati kako bi se naučio konzistentno savršeno blokirati svaki napad. Zato je bolja strategija dash-ati prema neprijateljskim napadima te pritisnuti blok kada se napad stvori ili malo prije. Osim toga neprijatelj ima i zračni napad koji koristi u primjerenim situacijama. Također može i blokirati napadne kombinacije igrača te dash-ati i odmah napasti lika.

Slika 4.3.3. Hodanje neprijatelja

Glavna razlika u hodanju neprijatelja i trčanju lika kojeg igrač kontrolira je, osim brzine, činjenica da se neprijatelj može kretati unazad bez mijenjanja smjera u kojem gleda. To je zato što neprijatelj uvijek pokušava ostati na udaljenosti od lika na kojoj ga može pogoditi sa svojim napadima. Iako se trudi uvijek gledati u smjeru lika, ako lik promjeni na kojoj je strani, neprijatelj se ne može odmah okrenuti već ima malo kašnjenje nasumične brzine. Iako je nasumična, brzina je uvijek približno ista. Ovo je tako dizajnirano ne samo kako bi bilo pravednije prema igraču, već kako bi se postigao bolji doživljaj neprijatelja kao stvarnog protivnika a ne samo programa.

```
//Movement
if(weak<9){
if(dodge==0 && counter==90 && block==0 && strike==0 && hit==0 && wlk_delay!=0){
    if(pos==0){
        if(wlk_delay!=1 && wlk_alarm==1){alarm[4]=irandom_range(15,25); wlk_alarm=0;}
        if(wlk_delay==1){
            if(atkdy>6 && ob_player.y > y){
                if(atkd>2){hsp++;}
            }
            else{
                if(atkd<30){hsp--;}
                else if(atkd>40){hsp++;}
            }
            if(image_xscale!=-1){image_xscale=-1;}
        }
    }
    else{
        if(wlk_delay!=2 && wlk_alarm==1){alarm[5]=irandom_range(15,25); wlk_alarm=0;}
        if(wlk_delay==2){
            if(atkdy>6 && ob_player.y > y){
                if(atkd>2){hsp--;}
            }
            else{
                if(atkd<30){hsp++;}
                else if(atkd>40){hsp--;}
            }
            if(image_xscale!=1){image_xscale=1;}
        }
    }
```

Slika 4.3.4. Kod hodanja neprijatelja

Slika 4.3.5. Blokiranje neprijatelja

Kao što je već spomenuto, neprijatelj može blokirati napade igrača visoko ili nisko. Uobičajeno će početi blokirati nakon četvrtog udarca, ali može i nakon trećeg ili petog. Tako je dizajniran kako bi player uvijek morao reagirati na blok, inače bi znao nakon kojeg će udarca neprijatelj blokirati. Ako igrač nastavi udarati neprijatelja bez da izvede guard break ili ga pogodi sa krivim guard break-om, neprijatelj će napraviti poseban potez kojim će staviti lika u posebno stun stanje, te će ga odmah nakon napasti.

```
//Block
if(hit != 0 && block_count == 0 && block == 0){
    block_count=round(random_range(3,5));
}
if(hit != 0 && block_count == 1 && block == 0 && ob_player.hurt == 0){
    if(stance==0){
        block=1;
    }
    else{
        block=2;
    }
    block_stop=1;
    block_count=0;
    if(pos==0){
        image_xscale = -1;
    }
    else{image_xscale = 1;}
    if(block == 1){
        if(image_xscale==1){b_inst=instance_create_layer(x-12,y+20,"Inst3",ob_enemy_block)}
        else{b_inst=instance_create_layer(x+12,y+20,"Inst3",ob_enemy_block)}
        b_inst.image_xscale=image_xscale
    }
    if(block == 2){
        if(image_xscale==1){b_inst=instance_create_layer(x-12,y+10,"Inst3",ob_enemy_block)}
        else{b_inst=instance_create_layer(x+12,y+10,"Inst3",ob_enemy_block)}
        b_inst.image_xscale=image_xscale
    }
}
```

Slika 4.3.6. Kod blokiranja neprijatelja

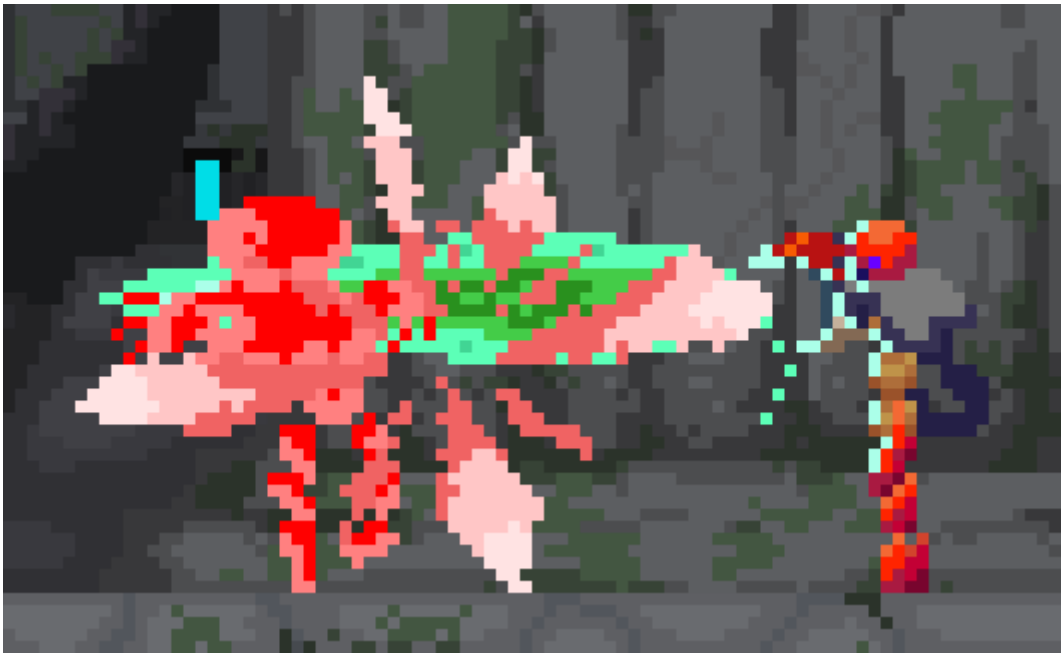
Slika 4.3.7. Dash neprijatelja

Kao i igrač, neprijatelj ima sposobnost izvođenja nepobjedivog dash poteza. Ipak, lik može udariti neprijatelja i ovom stanju ako ga pogodi sa boosted napadom.

Postoji dva tipa ovog dash-a. Prvi je dugački dash kojeg neprijatelj izvodi svaki put kada izađe iz napadne kombinacije lika. Ovaj dash će također izvesti u slučaju da ga lik pogodi sa guard break napadom kada ne blokira. Nakon dugačkog dash-a će napasti lika sa jednim od svoja dva primarna napada. Drugi tip je kratki dash koji izvodi odmah nakon što stavi lika u stun stanje. Nakon ovog dash-a će također napasti lika sa jednim od dva primarna napada, iako je veća šansa da će taj napad biti projektil pošto je on brži.

```
if(dodge==0 && hit>0){dodge = 1;}
if(strike>0){dodge = 0;}
if(dodge==1 && hit==0 && block!=5){if(pos==0){hsp=-5} else{hsp=5}; dodge=2; alarm[2]=dodge_dist;
    if(pos==0){image_xscale = -1;}
    else{image_xscale = 1;}
}
if(dodge==2){
    shadow = instance_create_layer(x,y,"Inst2",ob_player_afterimage);
    shadow.sprite_index = sprite_index;
    shadow.image_index = image_index;
    shadow.image_xscale = image_xscale;
    if((place_meeting(x+5,y,ob_wall) || place_meeting(x+5,y,ob_enemy)) && hsp==5){hsp=-5;}
    if((place_meeting(x-5,y,ob_wall) || place_meeting(x-5,y,ob_enemy)) && hsp==-5){hsp=5;}
}
```

Slika 4.3.8. Kod dash-a neprijatelja



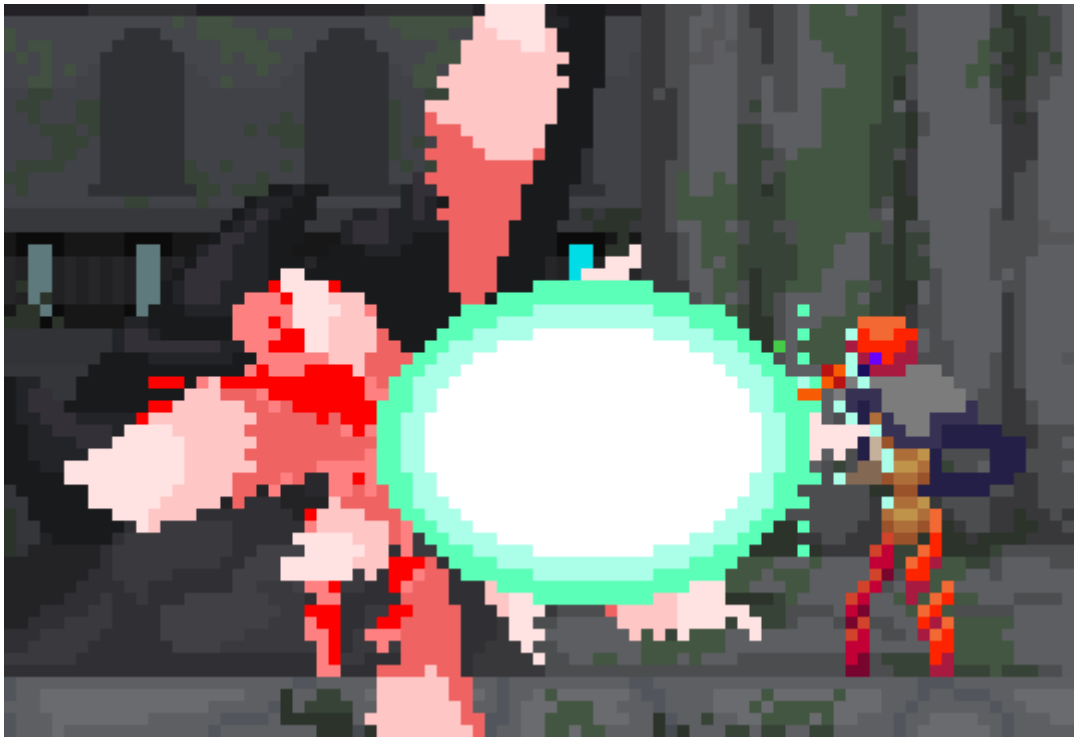
Slika 4.3.9. Prvi napad

Jedan od dva primarna napada neprijatelja je laser sa velikim dometom. Ovo je najsporiji napad, ali pokriva veliko područje, dugo traje i radi veću štetu od

projektila. Ovaj napad je i teže savršeno blokirati, iako se može dash-ati prema njemu za lagani savršeni blok zbog njegovog dugog trajanja i velikog hitbox-a. Dugo trajanje također je korisno pošto se neprijatelje ne može udariti dok napadaju, osim sa boosted napadima.

```
if(strike==1){sprite_index=sp_fe_laser_attack; image_index=0; strike=2;
    if(pos==0){image_xscale = -1;}
    else{image_xscale = 1;}
    vsp=0;
}
if(strike==2 && image_index>9 && striking == 0){striking=1;
    atk_inst = instance_create_layer(x,y,"Inst2",ob_fe_atk_hbx1);
    atk_inst.image_xscale = image_xscale;
    atk_inst.parnt = id;
    alarm[7]=irandom_range(60,110);
}
if(strike==2 && image_index>15){strike=0; hit=0; striking=0;}
```

Slika 4.3.10. Kod prvog napada



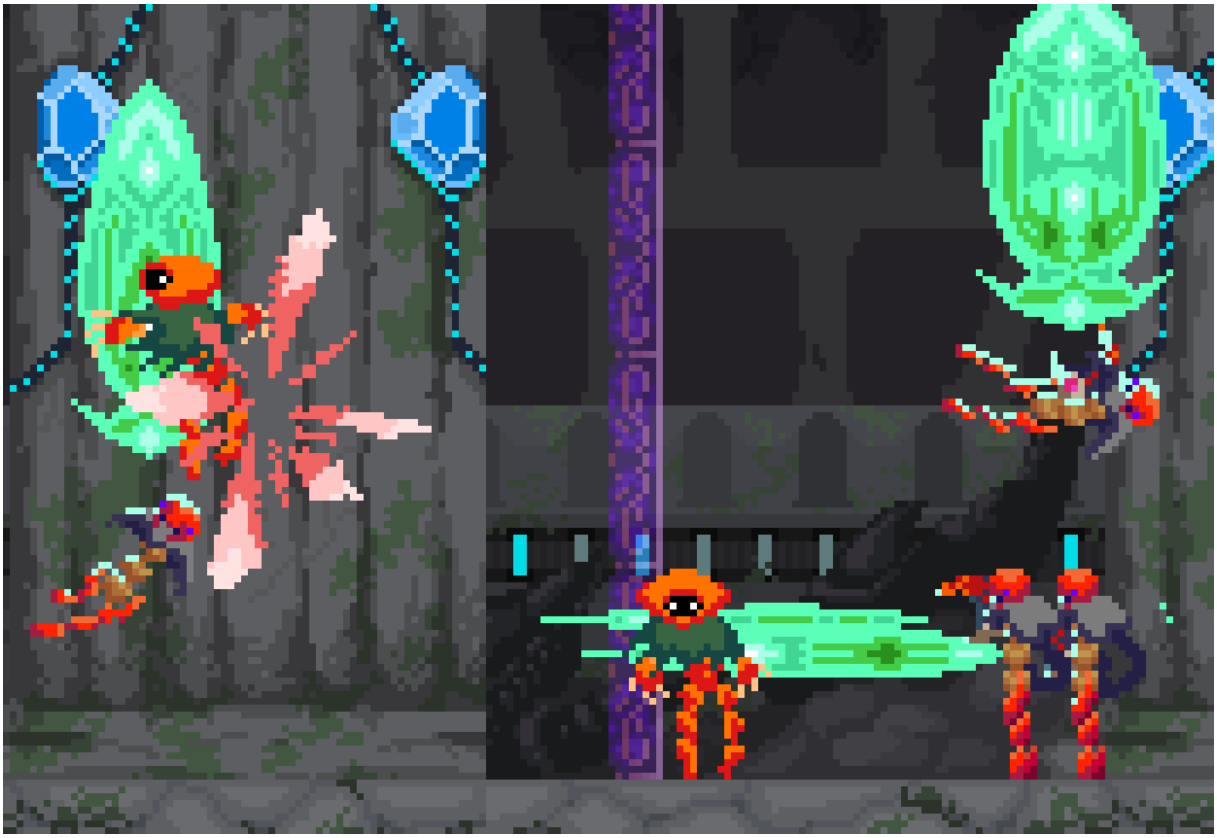
Slika 4.3.11. Drugi napad

Projektil radi nešto manju štetu od lasera, ali je iz blizine puno teži za izbjeći. To je zbog brzine ovog napada, i činjenice da je projektil, iako manji od lasera, i dalje dosta velik. Iz ovih razloga projektil ima veću šansu da bude izveden u slučaju da lik završi u stun stanju. Ako neprijatelj napravi laser, igrač ima vremena blokirati ga čak i

ako ne dash-a iz stun-a. Ali za projektil nema toliko vremena i ako igrač ne dash-a, garantirano je da će ga pogoditi.

```
if(strike==3){sprite_index=sp_fe_fireball_attack; image_index=0; strike=4;
    if(pos==0){image_xscale = -1;}
    else{image_xscale = 1;}
    vsp=0;
}
if(strike==4){
    if(image_index>5 && shot==0){
        pro_inst = instance_create_layer(x-(18*image_xscale),y+11,"Inst3",ob_fe_projectile);
        pro_inst.hsp = 3*-image_xscale;
        pro_inst.vsp = 0;
        pro_inst.image_xscale = image_xscale;
        shot=1;
        alarm[7]=irandom_range(60,110);
    }
    if(image_index>8){strike=0; shot=0; hit=0;}
}
```

Slika 4.3.12. Kod drugog napada



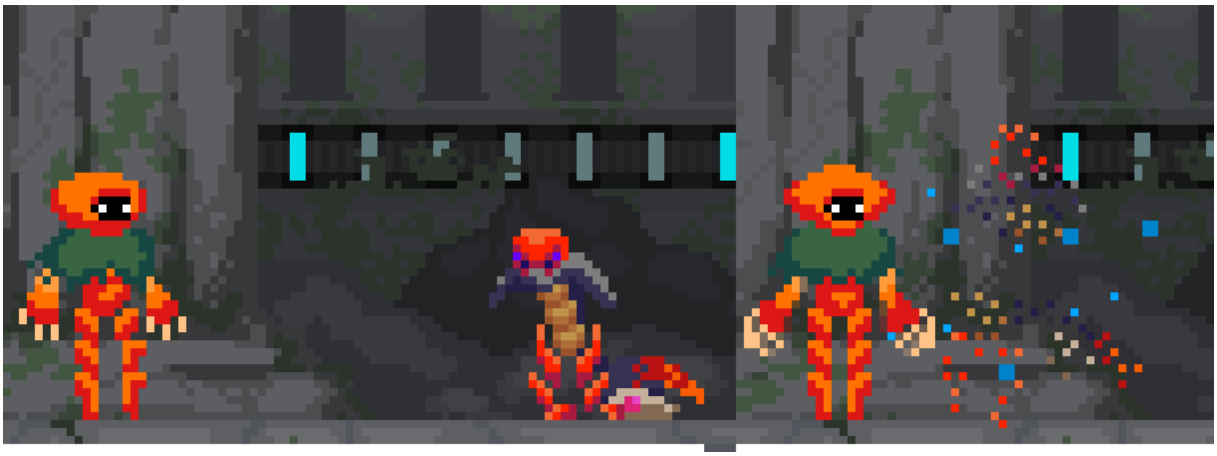
Slika 4.3.13. Treći napad

Zračni napad neprijatelj koristi samo u dvije situacije – ako je lik u zraku ili ako su dva ili više neprijatelja između njega i lika. Ovaj napad nije posebno opasan ako je pristuan samo jedan neprijatelj. Međutim, puno su veće šanse da pogodi lika dok

pokušava preskočiti napade drugih neprijatelja. Napad je također tu kako igrač ne bi mogao jednostavno lansirati jednog neprijatelja u zrak te ga držati u zraku napadnom kombinacijom bez brige za druge neprijatelje. Preskakanje drugih neprijatelja ovim napadom korisno je kako bi se brže približio liku, a djeluje i u slučaju da lik pokuša preskočiti neprijatelje kako bi lagano pobjegao iz nepoželjne situacije.

```
if(strike==5){sprite_index=sp_fe_jump_attack; image_index=0; strike=6;
    if(pos==0){image_xscale = -1;}
    else{image_xscale = 1;}
    dodge=0;
    vsp=0;
}
if(strike==6 && image_index>5 && striking == 0){striking=1;
    atk_inst = instance_create_layer(x,y,"Inst2",ob_fe_atk_hbx2);
    atk_inst.image_xscale = image_xscale;
    atk_inst.parnt = id;
    alarm[7]=irandom_range(60,110);
}
if(strike==6){
    if(image_index>2){if(image_xscale==-1){hsp=1;} else{hsp=-1;} vsp=-1;}
    if(image_index>3){if(image_xscale==-1){hsp=2;} else{hsp=-2;} vsp=-4;}
    if(image_index>4){vsp=-2;}
    if(image_index>5){if(image_xscale==-1){hsp=3;} else{hsp=-3;}vsp=4;}
    if(image_index>6){vsp=1;}
    if(image_index>7){vsp=0;}
    if(image_index>8){strike=0;hsp=0;vsp=0; hit=0; striking=0;}
}
```

Slika 4.3.14. Kod trećeg napada



Slika 4.3.15. Nestajanje neprijatelja

Ako health neprijatelj-a padne na 0, neće odmah nestati. Prvo će početi bljeskati narančastom bojom kako bi igrač znao da mu je health prazan, i tek kada

završi napadna kombinacija lika unutar koje je smanjen health na nulu će neprijatelj napraviti jedan zadnji dugački dash popraćen napadom, nakon čega će se izvesti animacija u kojoj padne na pod i nestane. Međutim, to se neće dogoditi u slučaju da igrač napravi finisher napad. U tom će se slučaju izvesti drugačija animacija neprijatelja kako nestaje odmah nakon što finisher napadi završe.

```
if(place_meeting(x,y,ob_player_finisher_hbx1) && weak==9){weak=10; hurt=1; sprite_index = hit1; layer = layer_get_id("Inst2");
    instance_create_layer(x,y,"Fade",ob_finisher_shake); ob_player.hp+=1; battle_score+=800;
    alarm[0] = 8;
    hspark_inst=instance_create_layer(x,ob_player.y+16,"Inst2",ob_finisher_hit_spark);
    if(pos==0){hspark_inst.x+=half_width}
    else{hspark_inst.x-=half_width}
    if(hspark_inst.y>(y+height)){hspark_inst.y-=(hspark_inst.y-(y+height));}
    if(hspark_inst.y<y){hspark_inst.y+=(y-hspark_inst.y)}
    hspark_inst.image_angle=random(360);
    }
if(place_meeting(x,y,ob_player_finisher_hbx2) && weak==10){weak=11; hurt=1; sprite_index = hit2;
    instance_create_layer(x,y,"Fade",ob_finisher_shake);  ob_player.hp+=2; battle_score+=1200; ob_player.ap+=0.1;
    alarm[0] = 8;
    hspark_inst=instance_create_layer(x,ob_player.y+16,"Inst2",ob_finisher_hit_spark);
    if(pos==0){hspark_inst.x+=half_width}
    else{hspark_inst.x-=half_width}
    if(hspark_inst.y>(y+height)){hspark_inst.y-=(hspark_inst.y-(y+height));}
    if(hspark_inst.y<y){hspark_inst.y+=(y-hspark_inst.y)}
    hspark_inst.image_angle=random(360);
    }
if(place_meeting(x,y,ob_player_finisher_hbx3) && weak==11){weak=12; hurt=1; sprite_index = hit1;
    instance_create_layer(x,y,"Fade",ob_finisher_shake);  ob_player.hp+=4; battle_score+=1800; ob_player.ap+=0.3;
    alarm[0] = 8;
    hspark_inst=instance_create_layer(x,ob_player.y+16,"Inst2",ob_finisher_hit_spark);
    if(pos==0){hspark_inst.x+=half_width}
    else{hspark_inst.x-=half_width}
    if(hspark_inst.y>(y+height)){hspark_inst.y-=(hspark_inst.y-(y+height));}
    if(hspark_inst.y<y){hspark_inst.y+=(y-hspark_inst.y)}
    hspark_inst.image_angle=random(360);
    }
if(place_meeting(x,y,ob_player_finisher_hbx4) && weak==12){weak=13; hurt=1; sprite_index = hit2;
    instance_create_layer(x,y,"Fade",ob_finisher_shake);  ob_player.hp+=8; battle_score+=2600; ob_player.ap+=0.6;
    alarm[0] = 8;
    hspark_inst=instance_create_layer(x,ob_player.y+16,"Inst2",ob_finisher_hit_spark);
    if(pos==0){hspark_inst.x+=half_width}
    else{hspark_inst.x-=half_width}
    if(hspark_inst.y>(y+height)){hspark_inst.y-=(hspark_inst.y-(y+height));}
    if(hspark_inst.y<y){hspark_inst.y+=(y-hspark_inst.y)}
    hspark_inst.image_angle=random(360);
    }
 if(hurt==0 && (weak>=10 && weak<15) && ob_player.atk==0){image_index=0;sprite_index=sp_fe_finished;weak=15;}
 if(weak==15 && image_index>4){
    with(ystars_inst){instance_destroy();}; with(atk_inst){instance_destroy();}; instance_destroy();}

if(hp<=0 && weak == 0){weak=1; alarm[9] = 3;}

if((weak == 1 || weak == 2) && hit == 0 && strike == 0 && dodge == 0){weak=3; hit=1000;}

if(hp<=0 && weak == 3 && strike == 0 && dodge == 0 && weak!=16){
    with(ystars_inst){instance_destroy();}; with(atk_inst){instance_destroy();};
    sprite_index=sp_fe_gone; image_index=0; mask_index=sp_fe_gone; hsp=0; vsp=0; weak=16;
}
if(weak==16){
    if(next_frame==4){
    vsp = vsp + grv;
    next_frame--;
    }
    else if(next_frame==1){
        next_frame=4;
    }
    else {
        next_frame--;
    }
}
if(weak==16 && image_index>11){
    instance_destroy();
}
```

Slika 4.3.16. Kod nestajanja neprijatelja

4.4. Korisničko sučelje

Htjelo se da korisničko sučelje ne izgleda previše dosadno, ali da ipak većina elemenata UI-a budu nekako tematski povezana. Pošto je autor u to vrijeme bio kreativno iscrpljen, sa velikim brojem grafika mu je pomogla kolegica Sara Laura Rokić.

Slika 4.4.1. Glavni izbornik

Glavni izbornik je dizajnirala i ilustrirala kolegica Rokić, dok ga je autor prilagodio i implementirao u igru. Ideja je bila imati pozadinu tematski vezanu uz rakove i drevne ruševine, što je općenito postala tematika većine grafičkih elemenata igre. Na izborniku se nalaze 3 opcije, stage select u kojem se odabira odlazak na sav drugi sadržaj u igri, options gdje igrač može podesiti razne postavke igre i pogledati kontrole, te quit game opcija koja jednostavno zatvara igru kada se odabere.

Slika 4.4.2. Stage select izbornik

Stage izbornik je također ilustrirala kolegica Rokić, na njemu se mogu odabrati 3 drugačija training stage-a i 2 battle stage-a, iako se prvo moraju otključati tako što ih se redom prođe. Također se tu zabilježavaju najviši rank i broj bodova koji je igrač ostvario na svakom battle stage-u.

Slika 4.4.3. Options izbornik

U options izborniku mogu se podesiti razne postavke igre. Ovaj options izbornik je dostupan i u izborniku pauze u igri. Njegove postavke su full screen, window size, vsync i fps display. Full screen mijenja igru iz pogleda iz prozora u pogled preko cijelog ekrana i obrnuto. Window size mijenja veličinu prozora. Vsync uključuje ili isključuje vertikalnu sinkronizaciju, što može utjecati na performanse. Fps display omogućuje prikaz broja slika koje igra prikazuje u sekundi. Također je tu opcija controls u kojoj se mogu pregledati kontrole za igru.

Slika 4.4.4. Kontrole - PC

Slika 4.4.5. Kontrole - Xbox

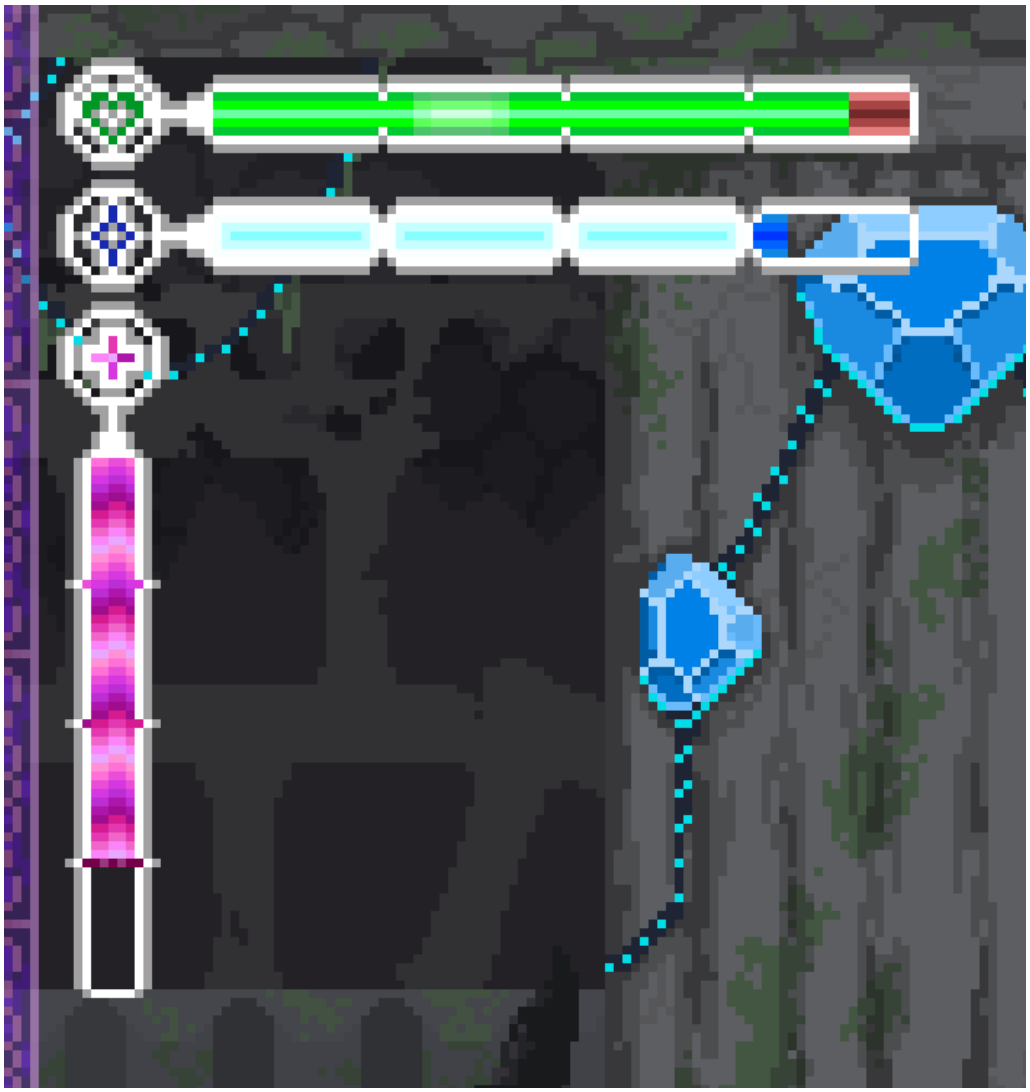
Slika 4.4.6. Kontrole - PlayStation

Kada se iz options izbornika odabere stavka controls, igrač može pregledati koje se kontrole koriste za igru. Ovdje se, kao i na svakom mjestu u igri na kojem su simboli za kontrole prikazani na ekranu, simboli za kontrole dinamički prilagođavaju ovisno koristi li igrač tipkovnicu, PlayStation tip kontrolera, ili Xbox tip kontrolera.

Step

```
if(keyboard_check_pressed(vk_anykey)){
    attack_key=sp_kb_k;
    back_key=sp_kb_p;
    jump_key=sp_kb_space;
    block_key=sp_kb_o;
    finish_key=sp_kb_e;
    training_combo1=sp_training_combo_1_kb;
    training_combo2=sp_training_combo_2_kb;
    controls=sp_controls_pc;
};
for(var i = gp_face1; i < gp_axisrv; i++){
if(gamepad_button_check(0, i)){
    attack_key=sp_xb_x;
    back_key=sp_xb_b;
    jump_key=sp_xb_a;
    block_key=sp_xb_y;
    finish_key=sp_xb_lb;
    training_combo1=sp_training_combo_1_xb;
    training_combo2=sp_training_combo_2_xb;
    controls=sp_controls_xb;
}
else if(gamepad_button_check(4, i)){
    attack_key=sp_ps_square;
    back_key=sp_ps_circle;
    jump_key=sp_ps_x;
    block_key=sp_ps_triangle;
    finish_key=sp_ps_l1;
    training_combo1=sp_training_combo_1_ps;
    training_combo2=sp_training_combo_2_ps;
    controls=sp_controls_ps;
}
}
```

Slika 4.4.7. Kod prepoznavanja input uređaja



Slika 4.4.8. Meter-i lika

Usred gameplay-a su u gornjem lijevom kutu prikazana 3 meter-a koji predstavljaju 3 drugačija aspekta lika kojeg se kontrolira. To su health, boost i finisher meter. Oni se dinamički pune i prazne u stvarnom vremenu kako se te varijable mijenjaju. Namjerno je svaki dizajniran sa malo drugačijim oblikom, bojom i ispunom i stilom animacije kako bi se lakše razlikovali.

ob_gui: Draw ...

Draw GUI

```
draw_sprite(sp_hp_meter,-1,6,6);
draw_sprite_part(sp_hp_meter_empty,0,0,0,hp_down+18,13,6,6);
draw_sprite_part(sp_hp_meter_full,-1,0,0,ob_player.hp+18,13,6,6);
draw_sprite(sp_amp_meter,-1,6,20);
draw_sprite_part(sp_amp_meter_full,0,0,0,round(ob_player.ap*20)+18,13,6,20);
draw_sprite_part(sp_amp_meter_bars,-1,0,0,ap_bars,13,6,20);
draw_sprite(sp_skill_meter,-1,6,34);
draw_sprite_part(sp_skill_meter_empty,-1,0,0,13,round(ob_player.skill*0.775)+18,6,34);
draw_sprite_part(sp_skill_meter_full,-1,0,0,13,skill_level,6,34);
```

Crustacea

File Edit Build Windows Tools Marketplace Layouts Help IDE

ob_gui: Step

Step

```
if(ob_player.ap==4){ap_bars=99;}
else if(ob_player.ap>=3){ap_bars=80;}
else if(ob_player.ap>=2){ap_bars=59;}
else if(ob_player.ap>=1){ap_bars=38;}
else if(ob_player.ap<1){ap_bars=0;}

if(ob_player.skill>=80){skill_level=80;}
else if(ob_player.skill>=60){skill_level=64;}
else if(ob_player.skill>=40){skill_level=48;}
else if(ob_player.skill>=20){skill_level=33;}
else{skill_level=0;}

if(ob_player.hurt==0 && (ob_player.recovery==0 && ob_player.block!=2) && hp_down>ob_player.hp){
    hp_down--;
}
image_speed=0.25*(fps/60);
```

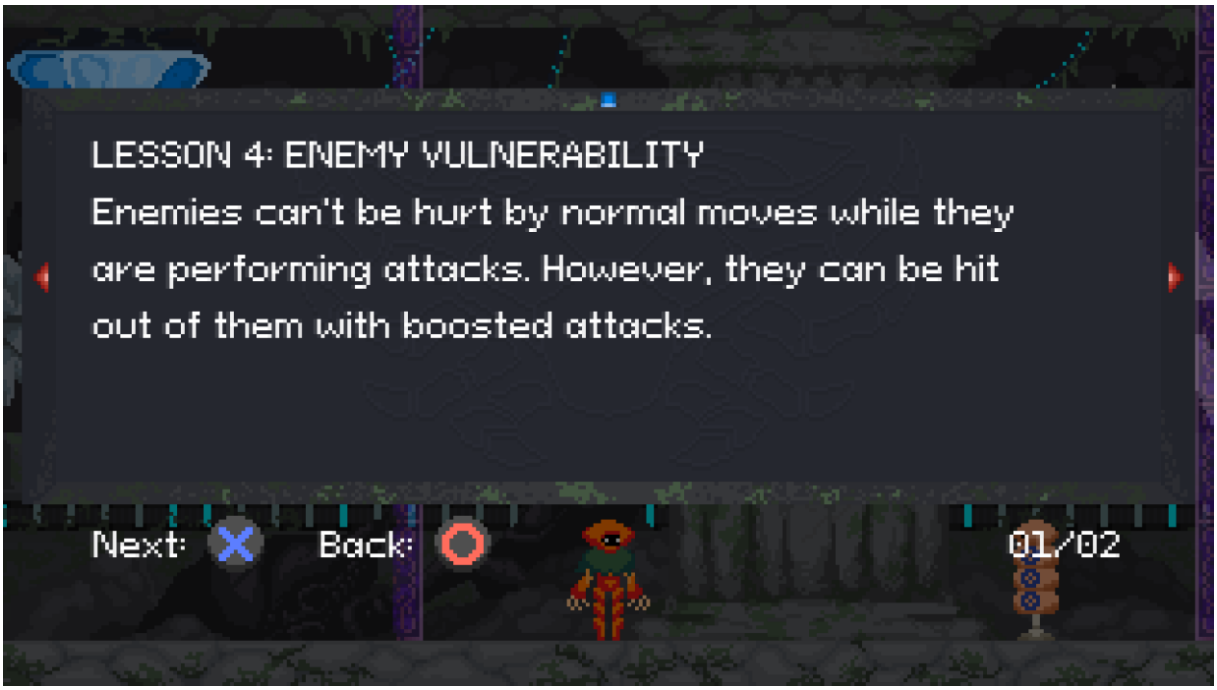
Slika 4.4.9. Kod meter-a lika

Slika 4.4.10. Izbornik pauze

Izbornik pauze još je jedan element koji je dizajnirala i ilustrirala kolegica Rokić, a ja autor ga je modificirao i implementirao u igru. Dizajniran je s motivom raka. Ovo je dosta jednostavan izbornik, ima opciju za gašenje pauze i vraćanje u igru, opciju za ponovno pokretanje sobe u kojoj se igrač trenutno nalazi, isti izbornik options kao onaj što se može pristupiti iz glavnog izbornika te opcija quit koja vraća igrača na stage select izbornik. Izbornik pauze pali se kada igrač pritisne tipku esc ili start, te se onda zamračuje ekran i stvara se izbornik. Također se na mjestu igrača i neprijatelja stvaraju marker-i prikladne boje koja označava njihove uloge.

Slika 4.4.11. Izbornik neuspjeha

Izbornik neuspjeha pojavi se nakon što se liku health smanji na 0. Prvo se ekran u potpunosti zamrači i samo se bijela silueta lika vidi. Na liku se vidi i zeleno srce koje kuca, te se onda raspršii pojavi se izbornik. Izbornik je modificirana verzija izbornika pauze sa samo dvije opcije, retry i quit, koje funkcioniraju identično onima iz izbornika pauze.



Slika 4.4.12. Training text box

Slika 4.4.13. Battle results text box

Na nekim mjestima u igri se pojavljuju text box-ovi kako bi se igraču nešto objasnilo ili kako bi mu se prikazala neka druga informacija. Prozor tih box-ova je dizajnirala kolegica Rokić.

4.5. Training stage

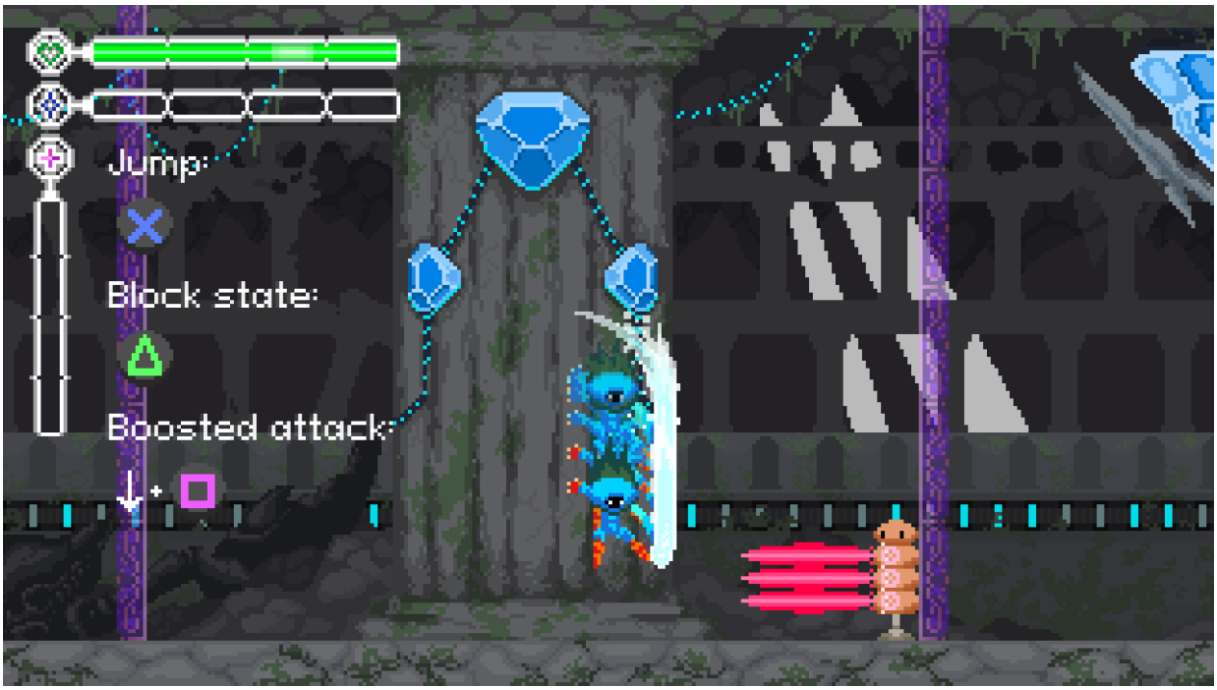
Cilj training stage-ova je da nauče igrača o raznim mehanikama igre. Sveukupno ih ima 3 i svaki je podijeljen na 3 dijela, te se u njima prođe kroz sve najbitnije mehanike, iako se ne objasni sve u najveće detalje kako bi igrač morao malo eksperimentirati u battle stage-ovima i naučiti kako se nositi sa stvarnim neprijateljima. U ovim stage-ovima neprijatelje predstavlja lutka koja je zapravo modificirana verzija neprijatelja u igri, iako su svi sprite-ovi i animacije lutke jedinstveni.

Slika 4.5.1. Prvi dio prvog training stage-a

Slika 4.5.2. Drugi dio prvog training stage-a

Slika 4.5.3. Treći dio prvog training stage-a

Prvi training stage je tu da objasni najosnovnije mehanike igre. To su normalni napadi, blokiranje i boosted napadi. Smatralo se da je najintuitivnije predstaviti ove mehanike prve i tim redom.



Slika 4.5.4. Prvi dio drugog training stage-a

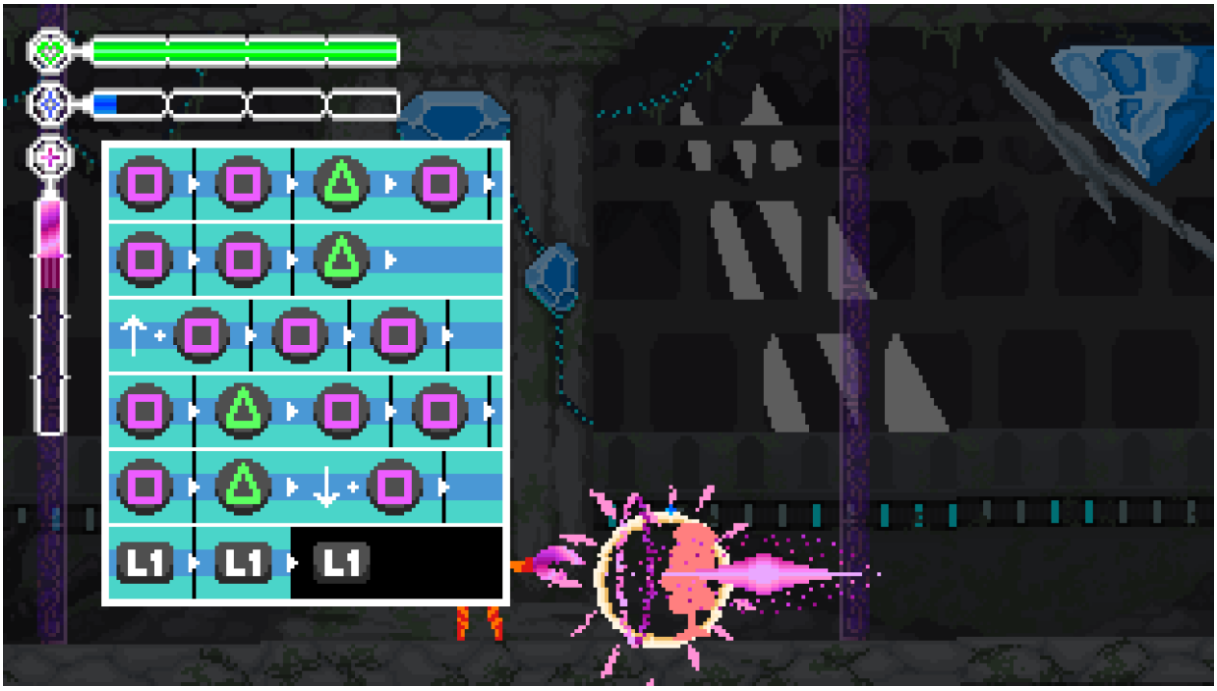
Slika 4.5.5. Drugi dio drugog training stage-a

Slika 4.5.6. Treći dio drugog training stage-a

Svrha drugog stage-a je naučiti igrača kako se neprijatelje ne može normalno udariti dok napadaju, kako koristiti boosted napada protiv neprijatelja koji napadaju, te objasniti guard break napade i dash.

Slika 4.5.7. Prvi dio trećeg training stage-a

Slika 4.5.8. Drugi dio trećeg training stage-a



Slika 4.5.9. Treći dio trećeg training stage-a

Zadnji training stage je tu kako bi istrenirao igrača za malo zahtjevnije radnje. Prvi dio ga nauči kako koristiti dash za lagani savršeni blok te brzo napadanje neprijatelja nakon blokiranja. Drugi dio je tu da objasni finisher napade i osnovnu strukturu napadnih kombinacija. Na kraju igrač mora unesti dugačku napadnu kombinaciju koja završava sa LV3 finisher-om. Svrha trećeg dijela je ne samo da objasni igraču kakve su kombinacije potrebne za jače finisher-e, već i da istrenira

igrača kako bi bio sposoban sam raditi dugačke kombinacije protiv neprijatelja u narednim stage-ovima.

4.6. Battle stage

Battle stage-ovi su glavni dio igre – dio zbog kojeg su sve mehanike i osmišljene. Dizajnirane su kako bi bile teške početnicima, ali moguće za preći. Međutim, za igrače koji žele postati majstori mehanika u igri, dizajniran je ranking sistem koji ocijenjuje igrače na osnovi njihovog performansa. Ranking sistem je baziran na bodovima. Bodovi se u dobivaju od napadanja neprijatelja te se nakon borbe dobiju i dodatni bodovi ovisno o tome koliko je health-a ostalo igraču te koliko je brzo prešao stage. Većina bodova se dobije od izvođenja finisher napada. Rank-ovi koji se mogu dobiti su, od najgoreg do najboljeg, E, D, C, B, A, S i SS.

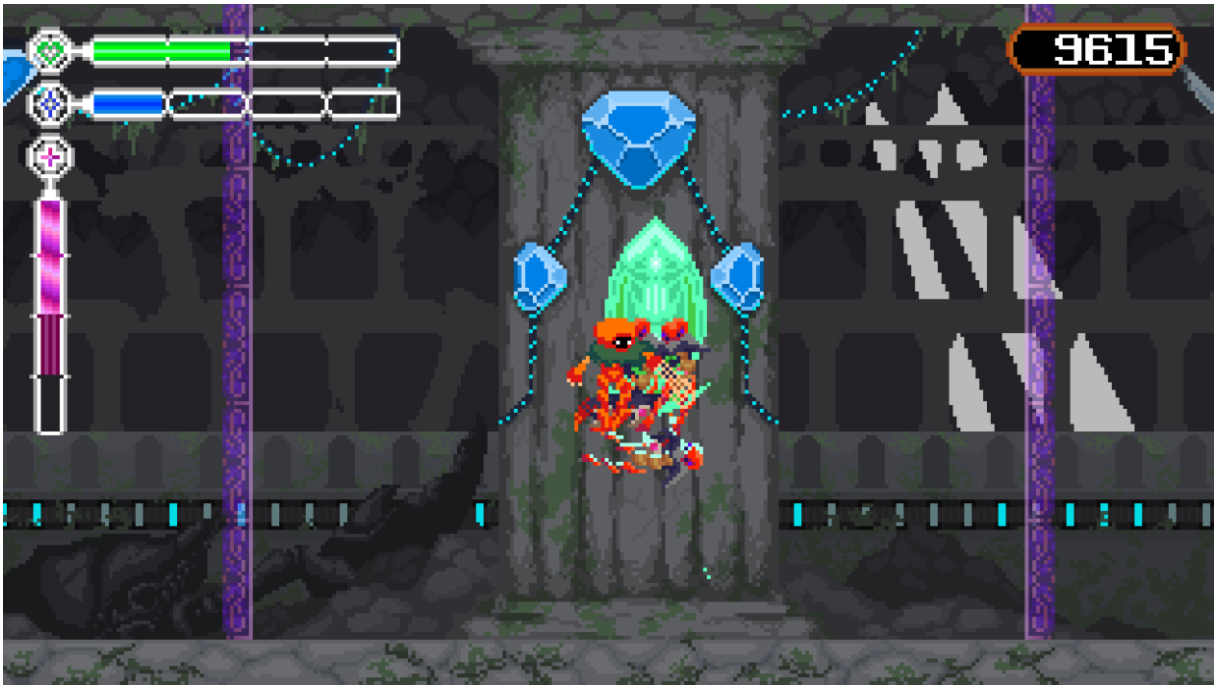
Slika 4.6.1. Prvi battle stage

Prvi stage je jako jednostavan – sastoji se samo od borbe između igrača i jednog neprijatelja. Čak i ovaj stage zna biti težak prvih nekoliko puta, tako da se igrači mogu vratiti ovom stage-u kako bi se uvježbali ako imaju problema sa drugim

stage-om koji je mnogo teži. Kako bi dobili SS rank na ovom stage-u, igrači moraju napraviti LV4 finisher nad neprijateljem.

Broj bodova potreban za rank:	Rank:
0	E
900	D
2000	C
3250	B
5250	A
7500	S
8100	SS

Tablica 4.6.1. Bodovni pragovi rank-ova u prvom battle stage-u



Slika 4.6.2. Drugi battle stage

Drugi je stage mnogo teži od prvog – ovaj stage kreće sa dva neprijatelja umjesto jednog, te se igrač mora naučiti kako se nositi sa dva neprijatelja odjednom. Nakon što ih pobjedi, stage ne završi već se stvore još tri neprijatelja. Tu već situacija postaje mnogo kaotičnijom. Nositi se sa 3 neprijatelja eksponencijalno je teže u odnosu na 1 ili 2. Ovaj stage je malo blaži što se tiče uvjeta za SS rank. Player i dalje mora izvesti LV4 finisher napad na većini neprijatelja, ali ne mora baš sve biti savršeno. To je zato što u ovom stage-u ima puno više varijabli i stvari koje bi mogle poći po zlu.

Broj bodova potreban za rank:	Rank:
0	E
2650	D
8800	C
12500	B
18000	A
25000	S
34000	SS

Tablica 4.6.2. Bodovni pragovi rank-ova u drugom battle stage-u

5. Zaključak

Kroz ovaj rad prikazani su workflow i sposobnosti GameMaker Studio 2 engine-a i Aseprite alata za izradu 2d pixel art grafika. Osim toga, prikazan je i proces osmišljavanja, dizajniranja i izrade video igre.

Iz povijesti video igara, prema reakcijama javnosti, može se zaključiti da su svi aspekti video igara vrlo bitni. Gameplay, glazba, grafički elementi, priča, zvuk i drugi aspekti gotovo uvijek dolaze u obzir u kritiziranju video igara, što naznačuje kako su svi oni bitni igračima. No, objektivna je činjenica kako je gameplay jedan aspekt po kojemu se razlikuju video igre od svih drugih medija zabavne industrije.

Iz tog je razloga fokus ovog rada bio na tom aspektu video igara, iako je počašćeno mnogo pažnje i drugim elementima pošto oni podižu gameplay na višu razinu.

U Varaždinu, 18.09.2021. Ivan Ljutak

Datum Potpis

Sveučilište Sjever

IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Ivan Ljutak *(ime i prezime)* pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom Izrada video igre koristeći GameMaker Studio 2 *(upisati naslov)* te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)
Ivan Ljutak
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnost i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, Ivan Ljutak *(ime i prezime)* neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom Izrada video igre koristeći GameMaker Studio 2 *(upisati naslov)* čiji sam autor/ica.

Student/ica:
(upisati ime i prezime)
Ivan Ljutak
(vlastoručni potpis)

6. Literatura

[1] – GameMaker Studio 2 Features And Tools:

https://www.yoyogames.com/en/gamemaker/features, rujan 2021.

[2] – Aseprite - Docs:

https://www.aseprite.org/docs/, rujan 2021.

[3] – David Capello:

https://davidcapello.com/, rujan 2021.

[4] – Igara Studio:

https://www.igarastudio.com/, rujan 2021.