

IZRADA 3D IGRE U UNITY PROGRAMU

Ančić, Maro

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:228:948626>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-13**



Repository / Repozitorij:

[Repository of University Department of Professional Studies](#)



UNIVERSITY OF SPLIT



SVEUČILIŠTE U SPLITU
SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Stručni prijediplomski studij Računarstvo

MARO ANČIĆ

ZAVRŠNI RAD

IZRADA 3D IGRE U UNITY PROGRAMU

Split, rujan 2024.

SVEUČILIŠTE U SPLITU
SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Stručni prijediplomski studij Računarstvo

Predmet: Sigurnost računala i podataka

ZAVRŠNI RAD

Kandidat: Maro Ančić

Naslov rada: Izrada 3D igre u Unity programu

Mentor: Lada Sartori, viši predavač

Split, rujan 2024.

Sadržaj

Sažetak	1
Summary.....	2
1. Uvod	3
2. Korištene tehnologije	4
3. Opis aplikacije.....	6
3.1. Korisničko sučelje.....	6
3.1.1 Glavni izbornik.....	6
3.1.2 Izbornik pauze	7
3.2 3D Grafičko Okruženje:.....	8
3.2.1 Kuća	8
3.2.2 Podrum	9
3.2.3 Šuma.....	10
3.3 Neprijateljski AI.....	11
3.3.1 Lutka	11
3.3.2 Duh.....	12
3.3.3 Kip.....	13
3.4 Sakupljanje predmeta i napredovanje	17
3.4.1 Šuma i otključavanje kuće.....	17
3.4.2 Istraživanje kuće i pristup podrumu	19
3.4.3 Podrum i škrinje s ključevima.....	21
3.4.4 Napredak i strategija.....	23
3.5 Mehanizam za bijeg	24
3.5.1 Podrum i vrata za bijeg.....	24
3.5.2 Izlazak u dvorište	27
3.5.3 Završetak igre.....	28

4. Zaključak.....	29
Literatura.....	30

Sažetak

Ovaj završni rad detaljno opisuje proces izrade 3D igre unutar Unity okruženja, s naglaskom na integraciju ključnih komponenti koje su neophodne za postizanje uspješnog i privlačnog iskustva igranja. Ključne komponente igre uključuju grafičko okruženje, neprijateljski AI, korisničko sučelje, sustav sakupljanja predmeta, te mehanizme za bijeg, a svaka od njih igra vitalnu ulogu u kreiranju sveobuhvatnog doživljaja za igrača. Korištenje Unity okruženja omogućilo je ne samo efikasno kreiranje kompleksnih vizualnih elemenata i interakcija, već i fleksibilnost u dizajniranju i implementaciji različitih funkcionalnosti koje čine igru izazovnom i zanimljivom. Rad pruža detaljan uvid u razvojne tehnike koje se mogu koristiti u izradi sličnih igara, ističući važnost usklađivanja svih elemenata igre kako bi se postigla koherentnost, balans između izazova i zabave, te ukupna funkcionalnost igre. Ova integracija omogućava kreiranje atmosfere koja potiče igrače na istraživanje i angažman, čineći igru uzbudljivom i privlačnom.

Ključne riječi: 3D igra, AI, korisničko sučelje, Unity,

Summary

Development of a 3D Game in Unity Software

This thesis provides a detailed description of the process of developing a 3D game within the Unity environment, emphasizing the integration of key components necessary for achieving a successful and engaging gaming experience. The main components of the game include the graphical environment, enemy AI, user interface, item collection system, and escape mechanisms, each playing a vital role in creating a comprehensive experience for the player. Utilizing the Unity environment has enabled not only the efficient creation of complex visual elements and interactions but also the flexibility in designing and implementing various functionalities that make the game challenging and engaging. The thesis offers a detailed insight into the development techniques that can be used in creating similar games, highlighting the importance of coordinating all game elements to achieve coherence, a balance between challenge and fun, and the overall functionality of the game. This integration facilitates the creation of an atmosphere that encourages players to explore and engage, making the game exciting and appealing.

Keywords: 3D game, AI, Unity, User Interface

1. Uvod

Ovaj završni rad opisuje proces izrade 3D igre u Unity okruženju, fokusirajući se na integraciju ključnih komponenti koje su bitne za uspješno igranje. Rad istražuje tehničke i dizajnerske aspekte razvoja igre, naglašavajući važnost usklađivanja različitih elemenata za postizanje funkcionalnog i privlačnog proizvoda.

Cilj ovog rada je pružiti pregled metoda i tehnika korištenih u razvoju igre unutar Unity okruženja, prikazujući kako se različite tehnologije i alati mogu koristiti za stvaranje koherentnog i uspješnog proizvoda.

Drugo poglavlje rada obrađuje tehnologije koje se koriste, uključujući Unity za razvoj igre, Blender za izradu 3D modela, i Unity Asset Store za preuzimanje resursa. Treće poglavlje detaljno opisuje aplikaciju, uključujući njene funkcionalnosti i dizajn. Četvrto poglavlje sadrži zaključak koji sažima glavne doprinose rada i predlaže smjernice za budući rad.

2. Korištene tehnologije

U izradi igre korištene su tri ključne tehnologije koje su bile od presudne važnosti za razvoj kompleksnog 3D horor iskustva. Svaka od ovih tehnologija doprinijela je specifičnim aspektima procesa izrade, omogućavajući kreiranje visokokvalitetne igre koja istovremeno izaziva strah i fascinira svojom izvedbom.

Unity je korišten kao glavna platforma za razvoj igre, zbog svoje sveobuhvatnosti i fleksibilnosti u radu s 3D okruženjima. Unity omogućava razvoj visokodetaljnih 3D scena, upravljanje složenim scenama i integraciju različitih elemenata igre kao što su grafički efekti, fizika, te umjetna inteligencija (AI) [1]. Ovaj razvojni alat podržava širok spektar mogućnosti, uključujući napredne alate za kodiranje, simulaciju fizičkih interakcija i skriptiranje događaja unutar igre. Korištenjem Unityja bilo je moguće brzo prototipirati i testirati različite aspekte igre, prilagoditi ih u skladu s povratnim informacijama i optimizirati performanse za različite platforme, uključujući računala, konzole i mobilne uređaje. Osim toga, Unity podržava visoku razinu prilagodljivosti u dizajnu korisničkog sučelja, što je bilo ključno za stvaranje intuitivnog i jednostavnog iskustva za igrače.

Unity Asset Store bio je ključan za učinkovito upravljanje vremenom i resursima tijekom razvoja igre. Ovaj online resurs omogućava pristup širokom spektru gotovih modela, tekstura, animacija, zvučnih efekata, skripti i drugih elemenata koji se mogu direktno integrirati u projekt. Upotreba Unity Asset Storea značajno je ubrzala razvoj igre, jer su se mogli koristiti kvalitetni resursi bez potrebe za kreiranjem svakog elementa ispočetka. To je omogućilo fokusiranje na dizajn specifičnih značajki igre koje čine iskustvo jedinstvenim, poput prilagodbe atmosfere i interakcija koje igrač doživljava tijekom igranja. Asset Store također pruža alat za pronalaženje resursa koji su kompatibilni s najnovijim verzijama Unityja, čime se osigurava da svi elementi igre rade besprijekorno zajedno i pružaju glatko korisničko iskustvo.

Blender, kao besplatan i alat otvorenog koda za 3D modeliranje i animaciju, bio je neizostavan dio alata korištenih u razvoju igre. Blender je omogućio izradu prilagođenih 3D modela likova, objekata i okruženja koji nisu bili dostupni na Unity Asset Storeu ili su zahtijevali specifične prilagodbe. Uz njegovu pomoć, dizajnirani su jedinstveni modeli koji su pridonijeli originalnosti i vizualnoj privlačnosti igre. Blender pruža napredne alate za teksturiranje, *rigging* i animaciju. Alat također podržava visoku razinu detalja, što je ključno

za stvaranje uvjerljivih modela i okruženja koji poboljšavaju igračevu uključenost u igru. Pored toga, Blenderova sposobnost izvoza modela u formate kompatibilne s Unityjem olakšala je integraciju prilagođenih elemenata u igru, čime se osigurava kontinuitet i kvaliteta između različitih alata za razvoj.

Kombinacija Unityja, Unity Asset Storea i Blendera omogućila je sveobuhvatno i efikasno kreiranje igre, osiguravajući visoku razinu kvalitete u svim aspektima razvoja. Korištenje Unityja kao okosnice za razvoj igre pružilo je stabilnu i fleksibilnu platformu za razvoj složenih interakcija i vizualnih elemenata. Unity Asset Store značajno je ubrzao proces razvoja pružanjem pristupa kvalitetnim resursima, dok je Blender omogućio stvaranje jedinstvenih i prilagođenih modela. Svaka od ovih tehnologija igrala je ključnu ulogu u stvaranju koherentnog, vizualno impresivnog i tehnički sofisticiranog horor iskustva koje angažira igrače i pruža im uzbudljivo i zastrašujuće okruženje u kojem mogu istraživati.

Rad na igri koristeći ove tehnologije također je omogućio individualni pristup razvoju, što je dalo slobodu u kreiranju specifičnih vizija i ideja bez potrebe za velikim timom. Ove tehnologije osigurale su sve potrebne alate i resurse kako bi jedna osoba mogla samostalno razviti kompleksnu i visoko kvalitetnu igru, fokusirajući se na svaki detalj od koncepta do konačne implementacije.

3. Opis aplikacije

U ovom poglavlju detaljno se razmatraju ključne komponente aplikacije igre, uključujući korisničko sučelje, 3D grafičko okruženje, neprijateljski AI, sustav sakupljanja predmeta i mehanizme za bijeg. Svaka od ovih komponenti igra važnu ulogu u oblikovanju iskustva igrača, omogućujući cjelovito i angažirano igranje. Ovdje se razmatra kako su ove komponente implementirane i kako doprinose funkcionalnosti i dinamici igre.

3.1. Korisničko sučelje

Dizajn korisničkog sučelja igra ključnu ulogu u iskustvu igrača, pa je posebna pažnja posvećena izradi jednostavnog i intuitivnog sučelja koje omogućava lako upravljanje igrom. Sučelje je kreirano koristeći razvojno okruženje Unity, pružajući igračima pristup glavnom izborniku s minimalnim brojem opcija radi jednostavnosti i učinkovitosti.

3.1.1 Glavni izbornik

Glavni izbornik (Slika 1) sadrži opcije za pokretanje nove igre i izlazak iz igre. Ova funkcionalnost osigurava da igrači mogu lako započeti svoju avanturu u igri bez potrebe za složenim postavkama ili dugotrajnim uputama. Kada igrač pokrene igru (opcija “Play”), korisničko sučelje postaje neupadljivo kako bi se omogućilo potpuno uranjanje u igru i fokus na atmosferu i igranje.



Slika 1: Izgled glavnog izbornika

Prilikom izlaska iz igre, korisnik može jednostavno odabrati opciju "Quit" iz glavnog izbornika, čime se igra zatvara. Dizajn sučelja stavlja naglasak na jednostavnost korištenja, omogućujući igračima brz i nesmetan pristup ključnoj funkcionalnosti bez potrebe za dodatnim navigacijskim elementima.

3.1.2 Izbornik pauze

Kako bi igrači mogli privremeno zaustaviti igru bez gubitka napretka, osmišljen je i jednostavan **izbornik pauze** (Slika 2). Aktiviranje izbornika pauze ostvaruje se pritiskom na tipku "Escape" tijekom igre, što automatski zaustavlja sve aktivnosti u igri i prikazuje osnovne opcije. Izbornik pauze sadrži opcije "Resume" za nastavak igre i "Quit" za povratak u glavni izbornik. Ova mogućnost omogućuje igračima da predahnu, promijene svoj fokus ili napuste igru kada je to potrebno.



Slika 2: Izgled izbornika pauze

Izbornik pauze dizajniran je tako da bude vizualno dosljedan glavnom izborniku, čime se zadržava konzistentnost sučelja. Opcije su jasno označene i lako dostupne, čime se osigurava jednostavno korištenje bez ometanja korisničkog iskustva. Povratkom u igru pritiskom na "Resume", igra se nastavlja od točke na kojoj je bila zaustavljena, osiguravajući neprekidno i ugodno igranje.

Takav pristup dizajnu korisničkog sučelja pruža korisnicima neometano iskustvo, minimizirajući potencijalne distrakcije i čineći igru pristupačnom širokom krugu igrača, bez obzira na njihovu razinu iskustva s igrama.

3.2 3D Grafičko Okruženje:

Dizajn 3D grafičkog okruženja ključan je za stvaranje uvjerljive i imerzivne atmosfere u horor igri. U ovom projektu korišten je Unityjev napredni grafički sustav kako bi se kreiralo realistično 3D okruženje koje igrače uvodi u svijet igre na način koji je istovremeno vizualno dojmljiv i emocionalno potresan [2]. Fokus je bio na detaljnom modeliranju i teksturiranju elemenata koji doprinose stvaranju zastrašujuće atmosfere, uključujući kuću, podrum i šumu.

3.2.1 Kuća

Kuća (Slika 3) je središnji dio igre, zamišljena kao napuštena građevina izgrađena od starih drvenih dasaka, koje daju osjećaj propadanja i zapuštenosti. Prozori su zabarikadirani debelim, grubo zakucanim daskama, što sugerira pokušaje zaštite ili zatočenja, dodajući dozu misterioznosti i opasnosti.

Interijer kuće modeliran je s velikom pažnjom prema detaljima. Hodnici su uski i mračni, osvijetljeni tek povremenim treperavim svjetlom koje stvara duge, prijeteće sjene. Zidovi su prekriveni starinskim tapetama koje su djelomično oštećene vlagom i prekriveni mrljama. Zvukovi škripanja podnih dasaka i udaljenih koraka dodatno pojačavaju osjećaj nesigurnosti i straha. Kroz prostorije su strateški postavljeni objekti kao što su stari namještaj, prašnjavi tepisi i napukla ogledala, svaki s namjerom da doprinese atmosferi napetosti i iščekivanja.

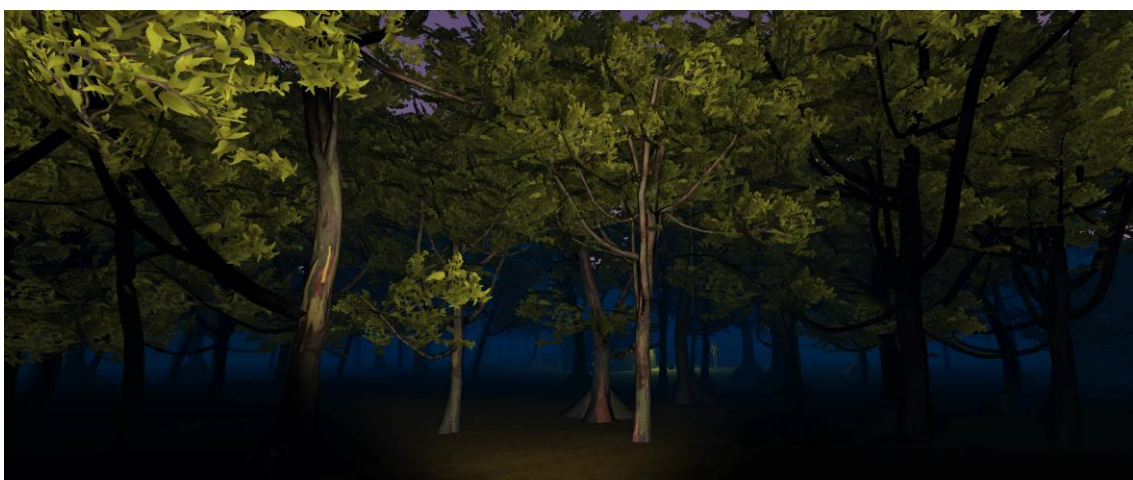


Slika 4: Izgled podruma unutar igre

3.2.3 Šuma

Okruženje igre također uključuje šumu koja okružuje kuću, dodatno naglašavajući osjećaj izoliranosti. Šuma (Slika 5) je gusta, s visokim stablima koja stvaraju tamu i osjećaj nelagode. Zrak je miran, a jedini zvuk koji se čuje dolazi od cvrčaka, stvarajući gotovo zlokobnu tišinu koja pojačava napetost.

Unutar šume postoji samo jedna staza koja vodi do kuće. Staza je stara i široka, pružajući igraču jedini siguran put kroz šumski teren. Ova jednostavnost dizajna pojačava osjećaj da je kuća jedino odredište, a povratak natrag nije opcija dok se ne obavi posao. Staza vodi direktno do zabarikadirane kuće, čime se dodatno pojačava osjećaj misterioznosti i straha.



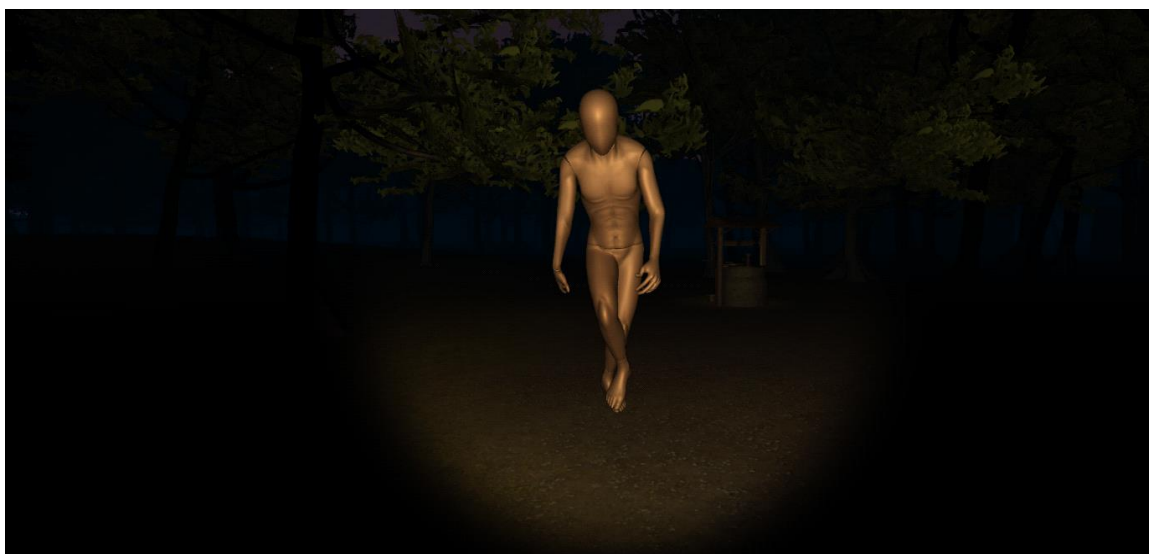
Slika 5: Prikaz šume koja okružuje kuću

3.3 Neprijateljski AI

U igri je razvijen sofisticirani AI sustav koji upravlja ponašanjem neprijatelja, stvarajući dinamične i izazovne situacije za igrača. Ovi AI algoritmi osiguravaju da neprijatelji nepredvidivo reagiraju na prisutnost igrača, svaki na svoj jedinstveni način, što doprinosi intenzivnom horor iskustvu igre [3]. Tri glavna neprijatelja u igri, svaki sa svojim specifičnim ponašanjem, čine jezgru ovog sustava: lutka, duh i kip.

3.3.1 Lutka

Prvi neprijatelj, lutka (Slika 6), nalazi se u šumi koja okružuje kuću. lutka lutajući šumom stvara osjećaj stalne prijetnje i neizvjesnosti. AI algoritam za lutku (Ispis 1) programiran je da reagira na blizinu igrača. Ako se igrač previše približi lutki, ona ga počinje aktivno loviti. U ovoj potjeri lutka se kreće brzo i neumoljivo, a ako uspije uloviti igrača, igra se resetira, vraćajući igrača na početak. Ovaj element igre tjera igrača da bude pažljiv unutar šume te da izbjegne lutku pod svaku cijenu.



Slika 6: Izgled lutke u šumi

```
if (angleToPlayer <= viewAngle / 2 f && distanceToPlayer <=
detectionRange && !obstacleBetween) {
    agent.destination = player.position; // Neprijatelj prati
igrača
    agent.speed = 5.5 f;
    Debug.Log("Distance to player: " + distanceToPlayer);
```



```

characterAnimation.SetBool("Seen", true); // ovo je animacija
characterAnimation.SetBool("Lost", false); // ovo je animacija
if (!heartbeatSound.isPlaying) {
    heartbeatSound.Play();
}
// Reproduciraj zvuk detekcije
if (!detectionSound.isPlaying && cnt == 0) // Provjeri je li
zvuk već u reprodukciji kako bi izbjegli neprekidno ponavljanje
{
    detectionSound.Play();
    cnt = 1;
}
} else {
    // Ako je agent stigao do trenutne destinacije, postavljamo
novu random destinaciju
    characterAnimation.SetBool("Seen", false);
    characterAnimation.SetBool("Lost", true);
    agent.speed = 2 f;
    if (agent.remainingDistance <= agent.stoppingDistance &&
!agent.pathPending) {
        SetNewRandomDestination();
        cnt = 0;
    }
}
}

```

Ispis 1: Kôd preko kojeg lutka primijeti te lovi igrača

3.3.2 Duh

Drugi neprijatelj, duh (Slika 7), boravi unutar kuće. Duh je slijep, ali izuzetno brz, što ga čini opasnim protivnikom. AI algoritam za duha (Ispis 2) osmišljen je tako da duh ne može ni vidjeti ni čuti igrača. Međutim, ako dođe do slučajnog kontakta, bilo da se duh zabije u igrača ili obrnuto, igra se resetira. Ova mehanika stvara napetost, jer igrač mora biti svjestan duhovog položaja i kretanja u kući kako bi izbjegao sudar, te time održao igru.



Slika 7: Duh u kući

```
void SetNewRandomDestination() {
    int randomNumber = Random.Range(0, 4);
    currentDestination = idle_spots[randomNumber].position;
    agent.destination = currentDestination;
}

IEnumerator WaitAndSetNewDestination(float waitTime) {
    yield
    return new WaitForSeconds(waitTime);
    SetNewRandomDestination();
}
```

Ispis 2: Funkcije za postavljanje nove nasumične destinacije za duha

3.3.3 Kip

Treći neprijatelj, poznat kao kip (Slika 8), nalazi se u podrumu kuće. Kip se ponaša na jedinstven način: dok ga igrač direktno gleda, kip se ne kreće. No, čim igrač skrene pogled, kip se počinje kretati velikom brzinom prema njemu. AI kipa (Ispis 3) koristi vizualnu mehaniku za praćenje pogleda igrača, što stvara osjećaj intenzivne paranoje. Sigurni dijelovi podruma osvijetljeni su svjetlom na stropu, koje kip izbjegava, čineći te dijelove sigurnim za igrača. Ova dinamička interakcija između igrača i kipa tjera igrača da pametno koristi osvijetljene prostorije za predah i promišljanje o daljnjem kretanju.



Slika 8: Kip u podrumu

```
void Update() {
    bool obstacleBetween = Physics.Linecast(transform.position,
    player.position, obstacleLayer);
    // Provjera da li igrač gleda neprijatelja
    float distanceToPlayer = Vector3.Distance(transform.position,
    player.position);

    if (distanceToPlayer <= 1.7 f) {
        //Debug.Log("Game Over triggered"); // Dodaj log
        //gameOverScreen.ShowGameOver(); // Prikaz Game Over ekrana

        gameOverScreen.ShowGameOver();

    }

    if (distanceToPlayer <= 3 f && IsPlayerLookingAtEnemy() && cnt
    == 0) {
        //Debug.Log("Game Over triggered"); // Dodaj log
        //gameOverScreen.ShowGameOver(); // Prikaz Game Over ekrana

        //gameOverScreen.ShowGameOver();
        scareSound.Play();
        cnt++;
    }
}
```

```

}

if (!IsPlayerLookingAtEnemy() && !IsEnemyInView()) {
    // Ako igrač ne gleda neprijatelja, neprijatelj se može
    kretati
    agent.isStopped = false;
    if (obstacleBetween && distanceToPlayer <= chaseRange) {
        MoveTowardsPlayer();
    } else if (!obstacleBetween && distanceToPlayer <=
chaseRange) {
        MoveTowardsPlayer();
    } else if (agent.remainingDistance <= agent.stoppingDistance
&& !agent.pathPending) {
        SetNewRandomDestination();
    }
    } else if (!IsPlayerLookingAtEnemy() && IsEnemyInView() &&
obstacleBetween) {
    // Ako igrač ne gleda neprijatelja, neprijatelj se može
    kretati
    agent.isStopped = false;
    if (obstacleBetween && distanceToPlayer <= chaseRange) {
        MoveTowardsPlayer();
    } else if (!obstacleBetween && distanceToPlayer <=
chaseRange) {
        MoveTowardsPlayer();
    } else if (agent.remainingDistance <= agent.stoppingDistance
&& !agent.pathPending) {
        SetNewRandomDestination();
    }
    } else {
    // Ako igrač gleda neprijatelja ili je neprijatelj u vidnom
    polju kamere, neprijatelj stoji
    StopAgentImmediately();
    isMoving = false;
    }
}
}

```

Ispis 3: Kôd preko kojeg kip prati igrača ako ga igrač trenutno ne gleda

U kodu (Ispis 4) može se vidjeti da se koriste određene funkcije za postizanje željenog ponašanja kipa. To su funkcije za provjeru gleda li igrač u neprijatelja, je li neprijatelj u vidnom polju, funkcija za zaustavljanje neprijatelja i funkcija za gibanje neprijatelja prema trenutnoj lokaciji igrača..

```
// Provjera da li igrač gleda neprijatelja
bool IsPlayerLookingAtEnemy() {
    bool obstacleBetween = Physics.Linecast(transform.position,
player.position, obstacleLayer);
    if (!obstacleBetween) {
        Vector3 directionToEnemy = transform.position -
player.position;
        float angle = Vector3.Angle(player.forward,
directionToEnemy);
        return angle < playerViewAngle / 2 f;
    }
    return false;
}

// Provjera da li je neprijatelj unutar vidnog polja kamere
bool IsEnemyInView() {
    Vector3 viewportPoint =
Camera.main.WorldToViewportPoint(transform.position);
    return viewportPoint.x >= 0 && viewportPoint.x <= 1 &&
viewportPoint.y >= 0 && viewportPoint.y <= 1 && viewportPoint.z >
0;
}

// Funkcija za trenutno zaustavljanje agenta
void StopAgentImmediately() {
    agent.isStopped = true;
    agent.velocity = Vector3.zero;
}

// Pomicanje prema igraču s offsetom
void MoveTowardsPlayer() {
    Vector3 directionToPlayer = (player.position -
transform.position).normalized;
```

```
Vector3 destination = player.position - directionToPlayer * 1.5f;
agent.destination = destination;
}
```

Ispis 4: Funkcije za provjeru vidnog polja igrača u odnosu na kip i gibanja istog tog kipa

3.4 Sakupljanje predmeta i napredovanje

U igri je implementiran detaljan sustav sakupljanja predmeta koji služi kao ključni mehanizam za napredovanje igrača kroz različite dijelove okruženja. Ovaj sustav ne samo da doprinosi osjećaju istraživanja i otkrivanja, već i stvara dodatnu napetost i složenost, čineći iskustvo igranja još zanimljivijim i uzbudljivijim.

3.4.1 Šuma i otključavanje kuće

Igrač započinje svoju avanturu u mračnoj šumi, okružen visokim stablima. Prvi zadatak igrača je pronaći ključ koji otključava ulazna vrata kuće. Ključ se nalazi u sefu koji je skriven na jednoj od tri moguće lokacije unutar šume. Kako bi otkrio pravu lokaciju, igrač mora pažljivo istražiti okoliš, pazeći na neprijatelja koji luta šumom.

Prije nego što može otvoriti sef, igrač mora pronaći papirić s kodom sefa (Slika 9). Ovaj papirić je strateški skriven i sadrži nasumično generirani četveroznamenkasti broj, koji se mijenja preko algoritma (Ispis 5) svaki put kada se igra pokrene, što onemogućava predvidljivost i zahtijeva od igrača da uvijek bude na oprezu.



Slika 9: Izgled papirića sa šifrom

```

void GenerateRandomNumber() {
    string randomNumber = "";

    // Generiraj četiri random broja od 1 do 9
    for (int i = 0; i < 4; i++) {
        randomNumber += Random.Range(1, 10).ToString();
    }

    // Postavi generirani broj kao tekst u gameobjectu
    numberText.text = randomNumber;
}

```

Ispis 5: Funkcija za generiranje nasumičnog četveroznamenkastog broja

Kada igrač pronade papirić s kodom, koristi interaktivni korisnički sučelje (Slika 10) na sefu za unos brojeva. Ukoliko unese pogrešan kod, sef ostaje zatvoren i na ekranu se prikazuje poruka „*Wrong password*”, čime se igrača potiče na daljnje istraživanje. Algoritam u pozadini provjerava korisnički unos (Ispis 6). Ako igrač uspješno unese točan kôd, sef se otvara i igrač dobiva ključ od kuće, omogućujući mu napredovanje u igri.



Slika 10: Prikaz otvorenog sefa i ključa unutar njega

```

public void Execute() {
    if (Ans.text == "") {
        // Ako je prazan string, nemoj postaviti "WRONG CODE"
        return;
    }
    if (Ans.text == answer && isOpen == true) {
        Ans.text = "ALREADY OPEN";
        return;
    } else if (Ans.text == answer) {
        Ans.text = "SUCCESS";
        safe.SetBool("Open", true);
        houseKey.SetActive(true);
        openText.SetActive(false);
        isOpen = true; // otvorit ce se vrata sefa
    } else {
        Ans.text = "WRONG CODE";
        openText.SetActive(true);
    }
}
}

```

Ispis 6: Funkcija za provjeru unosa unutar klase tipkovnice za sef

3.4.2 Istraživanje kuće i pristup podrumu

Nakon što otključa ulazna vrata kuće, igrač ulazi u unutrašnjost koja je tamna. Kuća je zabarikadirana drvenim daskama, a zrak je ispunjen osjećajem napuštenosti i prijatnje. Unutar kuće, igrač mora pronaći pilu i ključ (Slika 11) od podruma, koji su nužni za daljnje napredovanje. Pila služi za uklanjanje drvenih dasaka koje blokiraju ulaz u podrum, dok ključ otključava vrata podruma (Slika 12). Ovi predmeti su skriveni u kući, potičući igrača na pažljivo istraživanje svake sobe i svakog kuta. Vrata podruma koriste algoritam (Ispis 7) koji provjerava je li igrač ranije skupio pilu i ključ.



Slika 11: Prikaz pile i ključa



Slika 12: Izgled vrata podruma

```

if (isOpen && inReach && Input.GetButtonDown("Interact") &&
invKey.activeInHierarchy) {
    openText.SetActive(false);
    return;
} else if (inReach && Input.GetButtonDown("Interact") &&
invKey.activeInHierarchy) {
    StartCoroutine(OpenDoorsDelay());
    openText.SetActive(false);
    isOpen = true;

} else if (inReach && Input.GetButtonDown("Interact") &&
!invKey.activeInHierarchy) {
    StartCoroutine(ShowMissingKeyText());

} else {
    DoorCloses(); // funkcija koja zatvara vrata
}
if (isOpen) {
    door.GetComponent < BoxCollider > ().enabled = false;
}

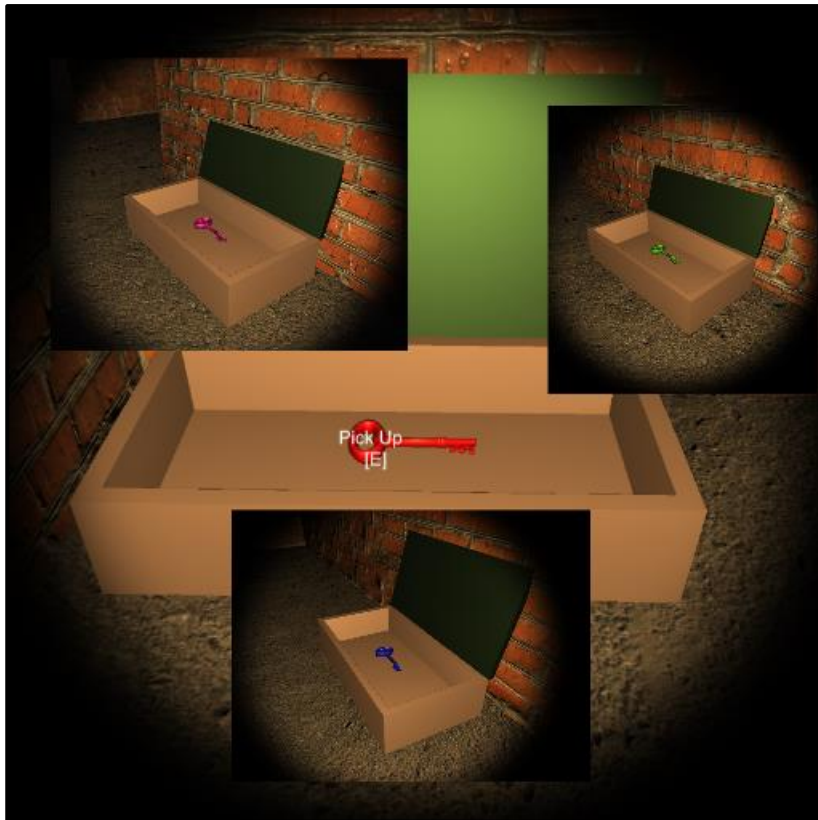
```

Ispis 7: Kôd koji provjerava ima li igrač alate potrebne za otvaranje vrata podruma

3.4.3 Podrum i škrinje s ključevima

Nakon što uspješno pronađe pilu i ključ, igrač sada može ući u podrum. Podrum je tamno, zastrašujuće mjesto ispunjeno sjenama i tišinom. Unutar podruma nalaze se četiri škrinje, svaka od njih sadrži ključ (Slika 13) određene boje: crveni, plavi, zeleni i ljubičasti. Svaki ključ otključava vrata (Slika 14) iste boje unutar podruma, a iza tih vrata nalaze se vrijedne vreće zlata. Koristi se kôd (Ispis 8) za provjeru pripada li određeni ključ određenim vratima.

Ovaj dio igre zahtijeva od igrača da pažljivo upravlja inventarom ključeva i taktizira koje će vrata prvo otvoriti, jer se četiri vreće zlata nalaze iza zaključanih vrata, dok su dvije vreće nasumično smještene po hodnicima podruma. Igrač mora prikupiti svih šest vreća zlata kako bi mu bio omogućen izlazak iz podruma i završetak igre.



Slika 13: Raznobarjni ključevi za vrata



Slika 14: Raznobarjna vrata

```

void Update() {
    if (inReach && Input.GetButtonDown("Interact") &&
keyOBNeeded.activeInHierarchy == true) // ako imamo
potreban(plavi) kljuc
    {
        DoorOpens(); // funkcija koja otvara vrata
        keyMissingText.SetActive(false);
        Debug.Log("It Opens");

    } else if (inReach && Input.GetButtonDown("Interact") &&
keyOBNeeded.activeInHierarchy == false) // ako imamo
potreban(plavi) kljuc
    {
        keyMissingText.SetActive(true);
        openText.SetActive(false);
        DoorCloses();
    } else {
        DoorCloses(); // funkcija koja zatvara vrata
    }
}

```

Ispis 8: Kôd za provjeru odgovara li određeni ključ određenim vratima (plava vrata)

3.4.4 Napredak i strategija

Ovaj sustav sakupljanja predmeta i napredovanja zahtijeva od igrača da razvije strategiju i planira svoje korake unaprijed. Slučajno generirane lokacije predmeta i nasumično generirani kôd sefa stvaraju dinamično iskustvo igranja gdje se igrači moraju prilagođavati novim izazovima pri svakom igranju. Napetost raste s svakim korakom, jer neprijateljski AI likovi stvaraju stalnu prijetnju, a svaki pogrešan potez može dovesti do ponovnog pokretanja igre.

Sustav sakupljanja predmeta i napredovanja ključan je za stvaranje intenzivne i napete atmosfere horor igre, pružajući igračima osjećaj postignuća i nagrade svaki put kada uspješno pronađu potrebne predmete i napreduju u svom zadatku.



Slika 15: Izgled zlatnih vreća

3.5 Mehanizam za bijeg

Kad igrač skupi svih šest zlatnih vreća (Slika 15) u podrumu, može pobjeći kroz posebna vrata (Slika 16) koja vode iz podruma u dvorište. Ako nije skupio sve vreće, vrata se neće otvoriti i pojavit će se poruka da treba skupiti svih šest vreća prije nego što pokuša pobjeći. Nakon što prođe kroz vrata, igrač mora doći do auta s kojim je stigao i pobjeći iz dvorišta, čime se igra završava. Važno je napomenuti da lutka i dalje luta dvorištem, pa igrač mora biti oprezan na putu do auta.

3.5.1 Podrum i vrata za bijeg

Nakon što igrač pronade i sakupi svih šest zlatnih vreća, mora se uputiti prema vratima koja se nalaze na suprotnom kraju podruma. Ta vrata vode izravno iz zemlje prema dvorištu kuće i predstavljaju jedini izlaz iz podruma. Međutim, ako igrač nije prikupio sve vreće zlata, vrata će ostati zaključana zbog implementiranog algoritma (Ispis 9). U tom slučaju, na ekranu će se pojaviti poruka koja ga obavještava da mora skupiti svih šest vreća prije nego što može pobjeći.



Slika 16: Izlazna vrata podruma

```
void Update() {
    if (inReach && Input.GetButtonDown("Interact") && canEscape ==
true)
    {
        DoorOpens(); // funkcija koja otvara vrata
        notEnoughBagsText.SetActive(false);
        Debug.Log("It Opens");
    } else if (inReach && Input.GetButtonDown("Interact") &&
canEscape == false)
    {
        notEnoughBagsText.SetActive(true);
        openText.SetActive(false);
        DoorCloses();
    } else {
        DoorCloses(); // funkcija koja drzi vrata zatvorenim
    }
}
```

Ispis 9: Kôd koji provjerava je li igrač skupio dovoljno zlatnih vreća

Tijekom igre, igrač može pratiti svoj napredak kroz brojač (Slika 17) u gornjem desnom kutu ekrana. Svaki put kada igrač skupi zlatnu vreću, algoritam (Ispis 10) će ažurirati brojač na način da će ga uvećati za jedan. Brojač prikazuje stanje u obliku "*Stolen Gold Bags*" i pomaže

igraču da zna koliko vreća još treba pronaći. Ova funkcionalnost osigurava da igrač bude svjestan svog napretka i motivira ga da temeljito istraži podrum te savlada sve prepreke prije nego što mu se omogući izlaz.



Slika 17: Brojač zlatnih vreća

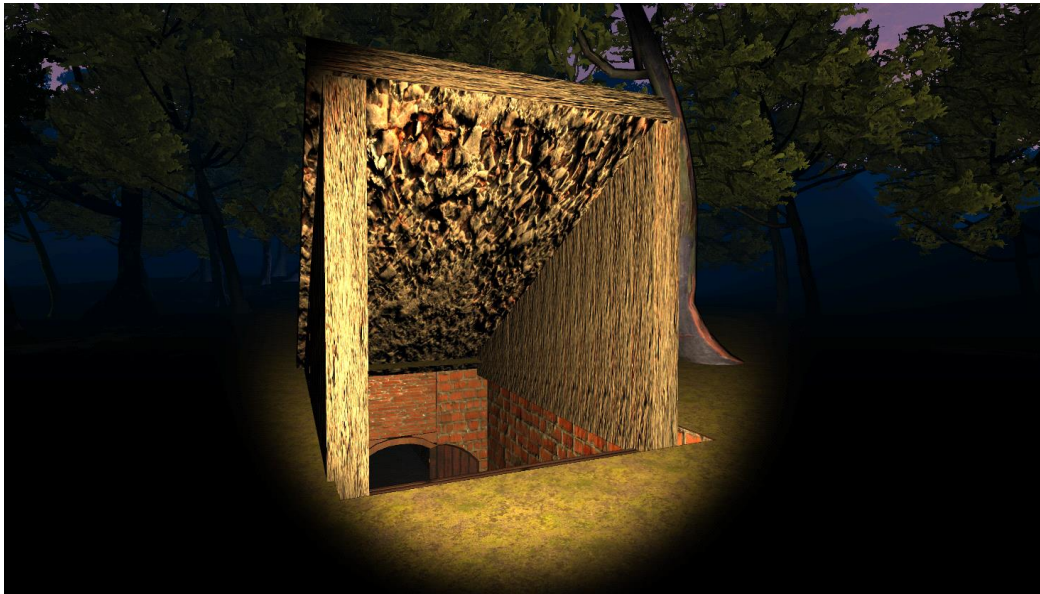
```
void Update() {
    if (inReach && Input.GetButtonDown("Interact")) {
        transform.position += new Vector3(-14 f, -10 f, -23 f); //
        Pomiče objekt ispod zemlje kako bi dao iluziju uzimanja
        pickupText.SetActive(false);
        Debug.Log("Interacted: Calling IncrementCount");
        bagsCollected.IncrementCount(); // Povećava count u
        BagsCollected
        cntGUI = bagsCollected.cnt;
        UpdateStolenGoldBagsText(); // Ažurirajte tekst kad ukradete
        zlatnu vrećicu
    }
    if (cntGUI > 5) {
        Debug.Log("You can escape now");
        exit.canEscape = true;
    }
}

void UpdateStolenGoldBagsText() {
    stolenGoldBagsText.text = "Stolen Gold Bags: " +
    cntGUI.ToString() + "/6"; // Ažuriranje teksta na UI Text
    elementu
}
}
```

Ispis 10: Kôd koji ažurira brojač zlatnih vreća

3.5.2 Izlazak u dvorište

Kada igrač konačno prikupi svih šest vreća zlata, vrata se otvaraju i vode ga u dvorište (Slika 18) kuće. Ovdje igra dobiva svoj posljednji preokret: iako je podrum napušten, dvorište je i dalje opasno jer lutka, neprijateljski AI lik, još uvijek luta ovim područjem. Igrač mora biti oprezan i izbjegavati lutku dok se kreće prema autu, s kojim je na početku igre stigao na lokaciju. Napetost je na vrhuncu dok igrač prolazi kroz dvorište, jer jedan pogrešan potez može dovesti do ponovnog početka igre, unatoč svemu što je do tada postigao.



Slika 18: Izlaz iz podruma



Slika 19: Auto za bijeg

3.5.3 Završetak igre

Nakon što igrač uspješno izbjegne lutku i stigne do auta (Slika 19), može napustiti dvorište i pobjeći iz te jezive situacije. Kada igrač sjedne u auto i odveze se, igra se završava. Na kraju se pojavljuje poruka "You Escaped" (Slika 20) s pozadinom šume, označavajući kraj igre. Ovaj trenutak pokazuje da je igrač prošao kroz sve izazove, od šume i kuće do podruma i dvorišta te pronašao način da pobjegne.



Slika 20: Poruka uspješno završene igre

4. Zaključak

Tijekom izrade ovog projekta, korištene su ključne tehnologije koje su omogućile kreiranje funkcionalne i vizualno privlačne igre. Unity je poslužio kao glavna razvojna platforma, nudeći sveobuhvatan alat za izradu kompleksnog 3D okruženja, upravljanje fizičkim sustavima, neprijateljskim AI-jem i grafičkim efektima. Korištenje Unity Asset Storea omogućilo je ubrzanje procesa, jer su gotovi resursi poput modela i tekstura korišteni kako bi se uštedjelo vrijeme na izradi određenih elemenata. Blender je korišten za modeliranje specifičnih dijelova igre, što je omogućilo kreiranje originalnih 3D objekata koji su savršeno odgovarali viziji igre.

Tijekom razvoja, najveći izazovi bili su povezani s pravilnom integracijom neprijateljskog AI sustava kako bi se postigao dobar balans između izazova i zabave. Neprijateljski AI zahtijevao je pažljivu prilagodbu kako bi reagirao u skladu s igračevim akcijama, a istovremeno bio dovoljno zahtjevan. Korištenje gotovih resursa iz Asset Storea značajno je ubrzalo proces razvoja vizualnih elemenata, omogućujući veću usmjerenost na detalje u mehanici igre.

Izrada korisničkog sučelja prošla je kroz nekoliko faza prilagodbe kako bi postalo jednostavno i intuitivno, bez ometanja imerzivnog iskustva igrača. Dobre strane ovog projekta uključuju uspješnu integraciju različitih tehnologija i alata, što je omogućilo izradu cjelovite i stabilne igre bez značajnijih tehničkih poteškoća.

Pproces izrade igre bio je iznimno koristan za razumijevanje tehnika razvoja i integracije različitih aspekata igre. Kombinacija Unityja, Blender modeliranja i resursa iz Unity Asset Storea omogućila je stvaranje horor igre koja pruža dinamično i izazovno iskustvo, ističući se svojom koherentnošću i visokom razinom tehničke izvedbe.

Literatura

[1] Wikipedia – „Unity (game engine)“ -

[https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)) - (Posjećeno 9.08.2024.)

[2] GameDeveloper - “Opinion: How to Make a Scary Game” -

<https://www.gamedeveloper.com/design/opinion-how-to-make-a-scary-game> - (Posjećeno 14.8.2024)

[3] Unity Learn – „AI and Navigation in Unity“ -

<https://discussions.unity.com/t/a-guide-on-using-the-new-ai-navigation-package-in-unity-2022-lts-and-above/371872> - (Posjećeno 16.08.2024.)