

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA
DE
MATEMATICĂ ȘI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ
LUCRARE DE LICENȚĂ

HERO'S ARCHETYPE
Joc de aventură pentru platforma Unity

Coordonator științific

Lect.Dr. Ștefan Popescu

Absolventă

Băjănaru Carmen-Andreea

București
2021

Cuprins

| | |
|---|-----------|
| <i>Introducere</i> | 6 |
| <i>Preliminarii</i> | 10 |
| Capitolul 1 | 10 |
| 1.1 Jocuri populare | 13 |
| Seria de jocuri Metroid | 13 |
| Hollow Knight | 14 |
| Ori and the Blind Forest | 14 |
| Ori and the Will of Wisps | 16 |
| 1.2 Metroidvania | 16 |
| 1.3 Concluzii | 17 |
| Capitolul 2 | 18 |
| 2.1 Platforma Unity | 18 |
| Asset Packages | 23 |
| Cinemachine | 23 |
| 2.2 Limbajul C# | 24 |
| 2.3 Concluzii | 25 |
| <i>Contribuția propriu-zisă</i> | 26 |
| Capitolul 3 | 26 |
| 3.1 Prezentarea aplicației | 26 |
| 3.1.1 Povestea jocului | 26 |
| 3.1.2 Descriere aplicație | 26 |
| 3.1.3 Prezentarea personajului controlat de utilizator (Player) | 27 |
| 3.1.4 Tilemap | 28 |
| 3.1.5 Platformă care se mișcă | 28 |
| 3.1.6 Planurile și efectul de paralaxă | 28 |
| 3.1.7 Armament/ Atac/ Luptă | 29 |
| 3.1.8 Prezentarea unui inamic | 29 |
| 3.1.9 Dialogue System | 30 |
| Capitolul 4 | 31 |
| 4.1 Aprecierile personale privind relevanța rezultatelor obținute | 31 |
| 4.2 Obiective pentru viitor | 31 |

| | |
|---------------------------|----|
| <i>Concluzii</i> | 32 |
| <i>Bibliografie</i> | 33 |

Rezumat

Proiectul acesta reprezintă un joc de platformă destinat sistemului de operare Windows, dezvoltat folosind Unity 2D. Obiectivul jocului este parcurgerea unor etape transformatoare pentru personajul ce înaintează prin diferite niveluri de dificultate variabilă.

Platforma Unity a fost aleasă pentru dezvoltarea acestui joc deoarece din ce în ce mai multe jocuri impresionante din punct de vedere al mecanicilor, dar și din punct de vedere vizual și conceptual au fost produse folosind Unity. Totodată, jocurile create utilizând Unity sunt compatibile cu o gamă variată de platforme, precum Android, Linux, Xbox, PlayStation etc.

Jocurile 2D sunt destul de populare în prezent datorită numărului mare de oameni care dețin tablete și telefoane mobile. Un factor la fel de important îl reprezintă și creșterea numărului de jocuri indie¹ 2D, ceea ce a condus inevitabil și la creșterea interesului publicului larg și pentru alte jocuri de acest gen.

În lucrarea de față, voi prezenta caracteristicile principale pentru Unity 2D și voi explica etapele dezvoltării unui joc de platformă în Unity. Scopul este dobândirea de cunoștințe și abilități, descoperind noile caracteristici ale platformei. Această lucrare poate fi privită și ca o analiză a ceea ce înseamnă dezvoltarea unui joc video, oferind un model pentru crearea unui joc 2D.

Anumiți termeni din această lucrare vor fi utilizați întocmai ca în documentația Unity oficială.

¹ ro. joc independent

Abstract

The project presented is a platform game, compatible with the Windows operating system and developed using Unity 2D. The objective of the game is for the character to go through some transformative stages while advancing through various levels, each with a different degree of difficulty.

The Unity platform was chosen for the development of this game, because more and more impressive games produced using Unity impress through mechanics, but also from a visual and conceptual standpoint. At the same time, games created using Unity are compatible with a wide range of platforms, such as Android, Linux, Xbox, PlayStation, etc.

2D games are quite popular nowadays due to the large number of people who use tablets and mobile phones. An equally important factor is the increase in the number of 2D indie games, which has inevitably led to increased public interest in other games of this genre.

In this paper, I will present the main features of Unity 2D and explain the path of developing a platform game in Unity. The goal is to acquire new knowledge and skills, discovering the new features of the platform. This paper can also be seen as an analysis of what it means to develop a video game, as well as a tutorial for creating a 2D game.

Certain terms in this paper will be used in the same way as in the official Unity documentation.

Introducere

Pentru a vorbi despre jocurile video este necesar să privim atent către cele mai importante etape care au consolidat ceea ce avea să devină o industrie de succes. Trebuie să menționăm progresul tehnologic, apariția consolelor de jocuri și nu în ultimul rând trebuie luată în considerare contribuția fiecărui nou joc apărut pe piață. Fie că vorbim despre consolele Nintendo, Xbox, Sega sau PlayStation, fiecare dintre acestea a împins înainte mersul istoriei jocurilor. Competiția crescută dintre acestea a contribuit considerabil la dezvoltarea industriei. Nu putem vorbi totuși despre o industrie fără a vorbi despre consumatorii acestui gen de divertisment sau despre creatorii jocurilor video. Dacă la început jocurile erau poate doar un mijloc prin care oamenii se relaxau, treptat complexitatea acestora a crescut, având un mai mare impact asupra lumii contemporane prin îmbinarea mai multor domenii - muzică, grafică, literatură, psihologie - cu programarea. Când vorbim despre jocuri, deja putem vorbi despre emoție și despre artă, despre depășirea unor bariere și despre crearea unor legături. Cu o foarte interesantă perspectivă ne întâlnim la Roberto Dillon, autorul cărții *On the Way to Fun: An Emotion-Based Approach to Successful Game Design*. Dillon reușește să ilustreze o viziune plenară în jurul ideii că există o corespondență între emoțiile umane și jocurile pentru calculator de succes. Acestea din urmă sunt apreciate mai ales datorită capacității de a oferi o experiență bazată pe comportamente și calități ale omului. Ceea ce atrage oamenii, cum au devenit în doar câteva decenii jocurile pentru calculator o formă de artă sunt lucruri asupra cărora mulți cercetători se opresc pentru a reflecta; deși au o istorie recentă, caracteristici precum explorarea psihicului uman, înțelegerea emoțiilor, precum și faptul că au devenit o formă de exprimare atât pentru creatori, cât și pentru consumatori sunt încurajări subtile pentru societate să își centreze atenția mai mult asupra a ceea ce continuă să se construiască și să se reinventeze - industria jocurilor video. (Dillon, 2018)

Istoria jocurilor video începe mai devreme chiar cu două decenii de 1962. Totuși, acest an este unul de referință, fiind momentul când Steve Russell, un student la Massachusetts Institute of Technology, atras de genul SF, a creat un joc numit *Spacewar!*. A fost un joc extrem de simplu comparativ cu cele existente în zilele noastre, însă, cu toate acestea, pentru perioada anilor '60 reprezenta un progres considerabil și un proiect complex. Mai mult decât atât,

singurele computere capabile să îl ruleze erau de dimensiunea unei camere. La finalul anilor '60, "Spacewar!" a devenit popular printre studenții americani.

Un pilon al dezvoltării industriei jocurilor l-a reprezentat apariția computerului personal. Un moment important al istoriei îl constituie crearea computerului Apple I de către Steve Jobs și partenerul său, Steve Wozniak. Această invenție a pus bazele creșterii numărului de persoane care joacă jocuri video.

În continuare, apariția consolelor a marcat evoluția acestei industrii. În ceea ce privește jocurile video, anii '80 sunt unii dintre cei mai semnificativi pentru dezvoltarea a tot ceea ce înseamnă gaming² din mai multe motive. Au apărut jocuri din ce în ce mai elaborate: Pac-Man, devenind primul joc de tip arcade pentru consolă, *Donkey Kong*, Tetris și *The Legend of Zelda* (1987). Acestea sunt serii de jocuri devenite clasice care sunt jucate și în prezent. În 1989 a apărut și consola *Gameboy* de la Nintendo, prima consolă portabilă importantă. (Glancey, 1996)

De-a lungul timpului, din ce în ce mai mulți oameni au început să se familiarizeze cu dispozitivele inteligente. Nu doar rețelele de socializare au reușit să unească oamenii, s-au creat întregi comunități și prin intermediul platformelor în care fanii își împărtășeau sfaturi, recorduri sau creații legate de noile jocuri apărute sau prin intermediul jocurilor online. Acest curs al lucrurilor vine oarecum natural având în vedere efectul terapeutic și educativ al acestora. (Griffiths, 2002)

Existența telefoanelor, a consolelor de jocuri, a laptopurilor și a calculatoarelor a favorizat creșterea numărului de persoane care aleg să petreacă o parte din timpul liber jucându-se. Dacă ne concentrăm atenția asupra platformelor pentru dezvoltare a jocurilor disponibile pentru PC, vom constata că unele dintre acestea au lansat jocuri datorită cărora anual se organizează competiții la nivel mondial cu premii considerabile, comparabile cu premiile celor mai importante competiții sportive. Acesta este un factor care nu doar că semnalează succesul industriei jocurilor, dar este și un factor de predictibilitate al continuării dezvoltării acestei ramuri a industriei. (Geig, M.,2021).

² termen folosit pentru a descrie tot ceea ce presupune activitatea de a juca jocuri video

Jocurile de tip single-player³ sunt de asemenea o sursă de divertisment pentru un număr mare de utilizatori. Acestea se concentrează pe alte aspecte apreciate de consumatori, pun accentul pe experiența individuală, poveste, grafică, păstrând centrală ideea de atingere a obiectivelor. Deși nu se poate vorbi despre competitivitate în modul clasic în cazul acesta, multe dintre jocurile single-player oferă din ce în ce mai mult în ultima perioadă posibilitatea de a explora și de a finaliza jocul în moduri proprii.

Prezenta lucrare poate fi considerată o extensie a universului creat în jocul "Hollow Knight" și o reinterpretare, în același timp, aducând un omagiu acestui joc indie realizat utilizând platforma Unity, apreciat la nivel mondial, și care a reușit să demonstreze nu doar faptul că din ce în ce mai mulți oameni au posibilitatea de a crea un joc, dar și că se poate realiza o aplicație impresionantă optând pentru simplitate; jocul a fost creat utilizând elemente disponibile pe Asset Store și PlayMaker (pentru inamici), în timp ce elementele de grafică au fost realizate de către un membru al echipei în Photoshop.

Îmi propun să ilustrez în lucrarea de față implementarea unui joc de platformă (single-player) utilizând motorul de joc Unity și să construiesc o analiză a stării actuale a industriei jocurilor. Utilizatorii au posibilitatea de a urmări în cadrul jocului parcursul personajului pe care îl joacă, după ce acesta își pierde identitatea și urmează o călătorie prin inconștient, explorând arhetipul eroului; provocările pe care le înfruntă iau forma unor inamici controlați de calculator, în timp ce colectează fragmente din sinele său sub forma unor bucăți de text. Obiectivul principal este invitarea la autocunoaștere prin oferirea unei experiențe cât mai plăcute pentru utilizatori, atât din punct de vedere vizual și tematic, dar și din punct de vedere al provocărilor. (Smith et al)

În cele ce urmează, *Capitolul 1* prezintă o viziune generală asupra jocurilor (2D) de tip pe platformă (eng. *platformer*).

Capitolul 2 prezintă atât tehnologiile folosite și limbajul de programare ales, precum și platforma pe care a fost dezvoltat jocul.

Capitolul 3 oferă o fereastră către prezentarea mai detaliată a implementării jocului care face obiectul acestei lucrări.

³ ro. un singur jucător

Capitolul 4 prezintă aprecieri personale privind rezultatele obținute și obiective de viitor.

Preliminarii

Capitolul 1

Observăm cum de-a lungul timpului interesul față de jocurile (2D) de platformă a coincis cu apariția unora dintre cele mai apreciate jocuri Nintendo sau cu un remake⁴ al acestora. Putem, de altfel, să concluzionăm că interesul crescut sau interesul scăzut al oamenilor pentru jocurile de platformă a fluctuat în funcție de game designul preferat în industrie. Publicul larg s-a dovedit a fi mai receptiv la niște principii și caracteristici ce pot fi considerate vechi, demonstrând totodată și importanța laturii emoționale; mai exact, pentru mare parte din persoanele care aleg să joace jocuri video contează într-o măsură considerabilă ca cerințele și obiectivele din fiecare nivel să trezească interesul și să producă jucătorilor o gamă variată de trăiri. Desigur, vom stabili în acest capitol ce anume trezește interesul oamenilor, precum și elementele care influențează acest aspect.

De remarcat este faptul că succesul jocului *Hollow Knight* pare să fi dus la creșterea numărului de utilizatori interesați de acest gen de jocuri, fapt care întărește convingerea că există anumite mecanici și elemente care par să fie apreciate de publicul larg și asigură, în mare parte, realizarea unui joc de succes.

În cadrul jocurilor de platformă, accentul se pune pe evitarea obstacolelor, iar acestea nu sunt un gen de jocuri independent, ci în majoritatea timpului îmbină elemente specifice unui RPG⁵, unui puzzle sau elemente specifice unui shooter⁶, fapt care va fi ilustrat și în *Hero's Archetype*.

Printre cele mai cunoscute jocuri platformer⁷ se numără *Donkey Kong* și *Super Mario Brothers 3* de la Nintendo. Din multe puncte de vedere, jocurile Nintendo au influențat profund industria, reușind să ofere întotdeauna personaje memorabile și mecanici tot mai complexe, precum și elemente unice de game design⁸. Motivul pentru care jocurile pentru consola

⁴ joc refăcut la o distanță de câțiva ani după lansarea inițială

⁵ **RPG** - Role Playing Games (jocuri video de rol)

⁶ joc video de acțiune în care accentul se pune pe învingerea inamicilor folosind o armă

⁷ joc de platformă

⁸ ro. designul jocului

Nintendo rămân clasice, fiind apreciate chiar și după zeci de ani, se poate datora designului lor atent gândit care atrage tot mai mulți oameni.

În analiza jocurilor platformer⁹, luăm în calcul rolul pe care îl are conceptul de level design¹⁰. Nu doar că acest concept este vital pentru procesul de creare al jocului, acesta fiind spațiul în care jucătorul explorează mecanicile, povestea, obstacolele, explorează elementele lumii virtuale și pătrunde din ce în ce mai adânc în universul ficțional prin descoperirea regulilor, însă acesta asigură fluiditatea și coerența necesare unui spațiu menit să ofere oamenilor o formă de divertisment. (Smith et al)

Perspectiva din care este necesar să fie privite jocurile este rezumată de ideea că într-o anumită măsură un joc presupune posibilitatea de a explora, oferită de creatorii jocului, și implică, pe de altă parte, capacitatea de a explora a utilizatorilor. Mai mult decât atât, vorbim despre explorare în contextul regulilor jocului. De aceea, rezultă imediat că e necesară înțelegerea în profunzime a tot ceea ce presupune crearea unui joc pentru a putea construi nivele complexe care să poată servi punerii în scenă a unei idei generale, ba chiar a unui mesaj pentru publicul țintă. Însă o viziune mai amplă este necesară, putem vorbi chiar despre necesitatea unei analize a genului abordat. Nu este suficient să creezi noi provocări pentru jucător, evoluția recentă a platformelor de game development¹¹ și apariția jocurilor indie tot mai elaborate și originale, primind apreciere din partea unui număr mare de persoane, impune un nivel ridicat pentru dezvoltatorii jocurilor, mai ales pentru cei cu tradiție (Xbox, Ubisoft, EA, Nintendo). (Fanfarelli, 2014)

Interesul oamenilor pare să varieze în funcție de nivelul de interes pe care îl au în mod normal pentru acest gen de divertisment. Deși există anumite jocuri care sunt alese de către utilizatori ce consumă în mod normal jocuri din categorii complet diferite, în mare parte a timpului acest lucru nu este valabil. Concret, deși există și jocuri apreciate de utilizatori cu cerințe complet diferite, astfel de aplicații sunt lansate destul de rar. De altfel, și dezvoltatorii de jocuri par să se axeze pe satisfacerea așteptărilor publicului țintă. Din acest punct de vedere, aplicațiile lansate pe piață își ating obiectivul. Totuși, având în vedere potențialul pe care îl au jocurile pentru calculator, îmbinând atât de multe domenii, nu este de preferat limitarea la a dezvolta cu preponderență aplicațiile pentru care pare a fi o cerere crescută. Mai mult decât

⁹ joc de platformă

¹⁰ proiectarea nivelurilor unui joc

¹¹ ro. dezvoltare a jocurilor

atât, este evident că oamenii apreciază momentele în care pe piață apar jocuri nu doar complexe sau care folosesc cele mai noi tehnologii, ci sunt și o expresie creativă a propriei lor viziuni. Ca în cazul oricărui alt mijloc de transmitere a viziunii personale, originalitatea prin care sunt ilustrate comportamente și interese umane devin puncte centrale și indicatori ai succesului.

Un alt aspect către care merită să ne îndreptăm atenția în cadrul analizei industriei jocurilor este că inclusiv în cazul în care dezvoltatorii jocurilor sunt limitați măcar la nivel conceptual ori din punct de vedere al poveștii, expresia artistică și calitățile tehnice sunt puse în valoare. Tehnologia se dezvoltă constant, din punct de vedere vizual mereu vor apărea îmbunătățiri, însă există serii de jocuri precum *The Witcher*, *The Last of Us*, care și după trecerea anilor își păstrează valoarea, deși multe aspecte ale jocurilor nu mai sunt de actualitate. În vederea creării unui joc memorabil, este necesar să se construiască o legătură clară între sursa de inspirație, game design¹², orizontul de așteptare al publicului țintă, tehnologiile folosite și celelalte domenii care sunt tangențial implicate în procesul de creație.

Am ales dezvoltarea unui joc de platformă deoarece are mult potențial - regulile sunt relativ simple, ușor de înțeles, însă permit realizarea unor aplicații de complexitate mare. Există o varietate de tipuri de jocuri platformer. Trei jocuri populare exemplifică acest lucru: *Super Mario World*, *Donkey Kong Country 2: Diddy's Kong Quest* și *Sonic the Hedgehog*. *Super Mario World* are un sistem de reward¹³ similar cu cel din *Super Mario 64*. Jucătorii sunt atrași de provocarea pe care o reprezintă completarea unui nivel. Există o singură cale de a completa nivelul, cu doar câteva locații ascunse la care pot căpăta acces jucătorii. Sonic, pe de altă parte, este un joc extrem de simplu ce constă în colectarea unor inele. Obiectivul jocului este să termini un nivel cât mai repede posibil. *Donkey Kong*, pe de altă parte, este un joc ce pune accentul pe explorare, este de dorit ca utilizatorii să parcurgă nivelurile în moduri proprii. (Smith et al.)

Platforma Unity este utilizată de mulți dintre creatorii de jocuri indie, *Hollow Knight* sau *Ori and the Blind Forest* sunt jocuri care au fost dezvoltate prima dată pentru platforma Unity. Mai mult decât atât, Unity este un mediu de lucru care permite atât începătorilor să se

¹² designul jocului

¹³ ro. recompensă

familiarizeze cu ceea ce înseamnă game development¹⁴, cât și profesioniștilor în domeniu să creeze în continuare aplicații complexe.

1.1 Jocuri populare

Seria de jocuri Metroid

În plus față de valoarea de divertisment, jocurile video au mult potențial, bucurându-se și de succes acele jocuri create pentru a rezolva o anumită problemă sau pentru a contribui la



Fig. 1

dezvoltarea unei abilități. (Arnott, 2017)

Seria de jocuri *Metroid* care a pus bazele subgenului *Metroidvania*, oferită utilizatorilor de către Nintendo, este printre cele mai vechi și printre cele mai apreciate la nivel mondial. A apărut încă din 1986, fiind un shooter¹⁵ platformer de acțiune aventură. Printre inovațiile aduse de această serie se numără faptul că a devenit „primul platformer¹⁶

neliniar în lume deschisă”, punea la dispoziția utilizatorilor o hartă, fiind necesară familiarizarea acestora cu geografia lumii jocului (Bissell, 2010). Totodată, lansarea jocurilor *Metroid* reprezintă un moment important în ceea ce privește muzica jocurilor video datorită utilizării în loc de muzică armonică a sunetelor pentru atmosferă (Collins, 2008, p. 26). De asemenea, arată un exemplu timpuriu de semiotică spațială intenționată. Luând ca exemplu o scenă de început din primul joc, utilizatorul trebuie să decidă între a merge către dreapta sau către stânga, lucru nemaîntâlnit până atunci în alte jocuri. Instinctiv, jucătorul ar fi înaintat spre dreapta, după modelul învățat din alte jocuri, neștiind că există opțiunea de a merge spre stânga. La un moment dat, drumul îi este blocat și poate înainta doar dacă se întoarce și merge către stânga pentru a obține ceea ce îl poate ajuta să își continue drumul în partea cealaltă. (Arnott, 2017)

¹⁴ dezvoltarea jocurilor

¹⁵ joc video de acțiune în care accentul se pune pe învingerea inamicilor folosind o armă

¹⁶ joc pe platformă

Hollow Knight

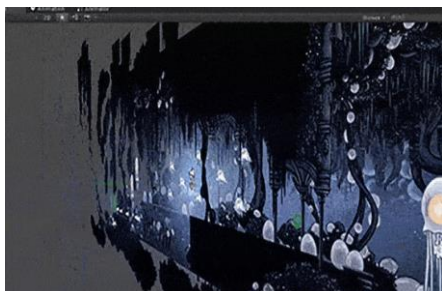


Fig. 2

Hollow Knight este un joc 2D de aventură și de acțiune. Pentru a rezolva un secret antic care a fost ascuns de mult timp, jucătorul călătorește într-un oraș subteran unde va lupta cu personaje care îl amenință, înfruntând în drumul său capcanele ce apar. Acesta explorează o lume imensă și interconectată, întâlnește aliați și oponenti care îl ajută să evolueze într-un personaj puternic, cu noi abilități și puncte forte. *Hollow Knight* impresionează și din punct de vedere tematic, dar și prin profunzimea ideilor care se împletesc în poveste. Amintind de „*Metamorfoza*” de Franz Kafka, personajul este modelat după un coleopter (eng. *beetle*), dar este și o trimitere subtilă către trupa *Beatles*. [8]

Jocul este creat într-un mediu 3D, așa cum se poate observa mai sus, în Fig. 2. Acesta impresionează și prin atenția la detalii; planul din față conține iarbă, aceasta fiind animată, se creează senzația de briză, personajul este desenat 2D, pentru atac au fost realizate mai multe cadre cu efecte vizuale pentru fiecare tip de atac (deblocat de către utilizator). Comparativ, un alt joc lansat în aceeași perioadă, și anume *Ori and the Blind Forest* (2015), folosește un model 3D pentru personaj, jocul fiind un hibrid 2D/3D. (Team Cherry, 2021)

Ori and the Blind Forest

Ori and the Blind Forest este un joc realizat utilizând platforma Unity, asemenea jocului *Hollow Knight*. Printre cele mai remarcabile caracteristici ale acestui joc sunt animația, povestea, modul subtil în care utilizatorul este familiarizat cu felul în care poate controla personajul pentru a se deplasa, dar și fluiditatea mișcării. Acesta îmbină elemente 3D cu elemente 2D, fiind un joc, din punct de vedere grafic, 2.5D; partea vizuală a jocului atrage

atenția din primele momente, elementele artistice fiind inspirate din animațiile realizate de Studio Ghibli.

Creatorii jocului, deși limitați de un cadru 2D pentru prezentarea elementelor de control pentru personaj, dar și pentru familiarizarea jucătorului cu universul din *Ori and the Blind*

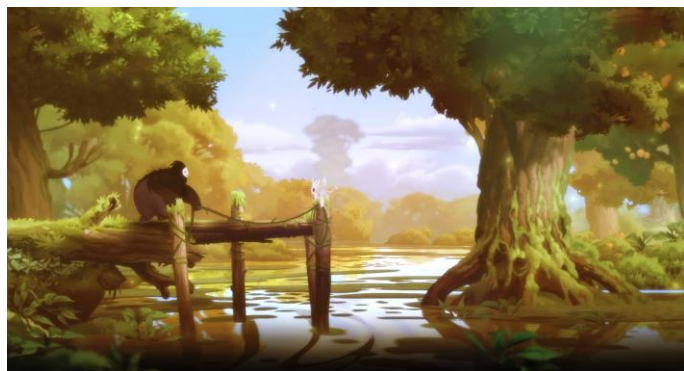


Fig. 3

Forest, și-au atins totuși obiectivele inițiale: au reușit să creeze iluzia de animație atunci când Ori se luptă cu inamici, scenele au fluiditate și totodată mediul în sine are anumite proprietăți și interacționează cu Ori (de exemplu, încăperea care e un obstacol pentru acesta; tavanul se prăbușește, lava este

îndreptată înspre Ori etc.). Totodată, aceștia au optat pentru crearea a 3 planuri - planul din spate, platforma și planul din față. Mișcând planurile cu viteze diferite, au reușit să creeze efectul de paralaxă. Deși această caracteristică este utilizată de mult timp și în alte jocuri, în cazul acestui joc planul din față este utilizat pentru a-l ghida pe jucător. Faptul că acest joc reușește să nu recurgă la metode evidente prin care ghidează utilizatorul este de asemenea apreciat. Multe jocuri recurgeau la instruirea jucătorului până dovedea că s-a familiarizat cu modul în care poate controla funcțiile jocului, fără a-i permite să înceapă direct, sau chiar afișau în mod explicit ceea ce trebuie făcut. Pentru utilizatori, faptul că li se acordă încrederea de a descoperi singuri jocul este, cu siguranță, unul dintre motivele pentru care au acordat recenzii pozitive. (Anthropy & Clark, 2014)

Pentru jocurile de tipul Metroidvania este specifică ascunderea unor obiecte care fie deblochează calea jucătorului, fie îi conferă noi abilități ș.a.m.d., încurajând astfel explorarea lumii din joc. În *Ori and the Blind Forest*, fiecare element (eng. *item*) are un scop, colectarea celor mai dificil de obținut elemente aduce cu sine o recompensă întotdeauna. De asemenea, s-a optat pentru evitarea, pe cât era posibil, a repetițiilor în ceea ce privește mediul, dorind a fi păstrate numai anumite abilități ale personajului pe tot parcursul jocului. (Gutierrez Rodriguez et al.)

Ori and the Will of Wisps

Când un zbor îndrăzneț al bufniței Ku o pune în pericol, curajul nu va fi de ajuns pentru a reuni o familie, pentru a reclădi ceea ce a fost distrus și pentru ca Ori să-și găsească menirea. Jocul de acțiune-aventură pe platformă, *Ori and Will of the Wisps* este aclamat de critici pentru efectele vizuale, mulți l-au considerat cel mai bun joc de tip Metroidvania, cel mai bun joc single-player și cel mai bun joc pe platformă al anului. A câștigat premii oferite de *Unity Awards*, *Windows Central*, *The Steam Awards* etc. Utilizatorii au posibilitatea de a călători printr-un univers mare, întâlnind aliați și adversari. Artistic, *Ori and Will of the Wisps* creează o impresie puternică, având o multitudine de imagini pictate manual. *Ori and Will of the Wisps* continuă tradiția Moon Studios de acțiune pe platformă. Este construit meticulos și are o poveste emoționantă. (Moon Studios, 2021)

1.2 Metroidvania

Termenul *Metroidvania* este atribuit jocurilor care folosesc mecanici asemănătoare, precum și elemente de design specifice seriilor de jocuri Metroid și Castlevania. Pe lângă numeroasele confruntări cu forțele inamice, jocurile de tip Metroidvania prezintă și lumi de dimensiuni considerabile care trebuie explorate, utilizând abilitățile învățate pe parcursul jocului. Acest gen de joc creează o experiență fascinantă bazată pe menținerea curiozității utilizatorului și pe depășirea obstacolelor, precum și pe ceea ce înglobează perspectiva evoluției, activând latura creativă și găsind noi utilizări pentru abilitățile deținute și întâlnirile inedite. Prin urmare, jocurile Metroidvania sunt creații foarte complexe care necesită mult timp pentru dezvoltare. (Oliveira et al., 2020)

Jocurile Metroidvania sunt de obicei jocuri pentru platformă side-scroller¹⁷ și pot fi realizate prin definirea corectă a următoarelor aspecte: mecanica jocului (toate acțiunile jucătorului sau ale personajului), regulile jocului (definesc niște consecințe pentru fiecare acțiune, i.e. în ce mod afectează o acțiune starea jocului), obiecte, personaje, nivelurile jocului, precum și elementele componente ale unui nivel. (Gutierrez Rodriguez et al.)

¹⁷ cu defilare laterală

1.3 Concluzii

Industria jocurilor a dovedit o deschidere din ce în ce mai mare față de jocurile 2D de-a lungul ultimului deceniu, mai ales după anul 2013 când dezvoltatorii Unity au introdus caracteristicile specifice modului 2D.

În ceea ce privește jocurile pe platformă, acestea, fiind inspirate din seriile clasice precum *Metroid*, impresionând prin grafică, muzică și sunete, dar și prin poveste, atrag din ce în ce mai mulți utilizatori.

Se observă niște aspecte pe care jocurile prezentate mai sus le au în comun: o poveste ingenioasă, elemente artistice care fac jocul memorabil, personaje carismatice, iar mecanica jocului este atent gândită, integrând idei deja consacrate care fac obiectul interesului utilizatorilor.

Capitolul 2

Tehnologii folosite

2.1 Platforma Unity

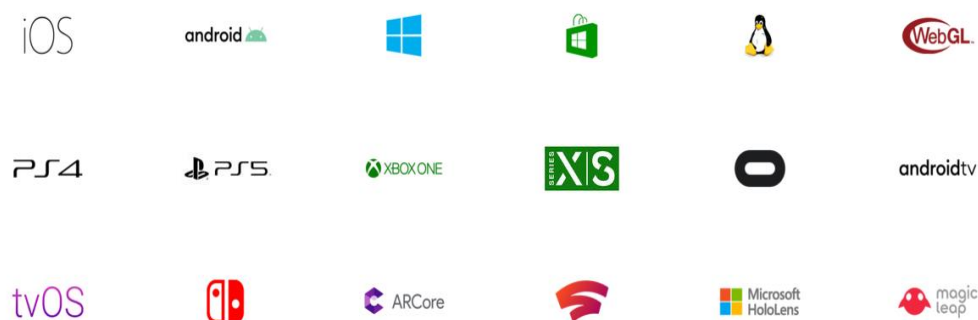


Fig.4

Platformele pentru dezvoltare a jocurilor precum Unity sunt uneltele esențiale din spatele jocurilor create pretutindeni. Unity este unul dintre cele mai frecvent utilizate și apreciate sisteme de creare a jocurilor. De la pasionați până la companii uriașe, un număr mare de oameni folosesc Unity pentru a crea jocuri pentru Windows, dispozitive mobile, pentru console și pentru web. (Unity Technologies, *Multiplatform*)

Unity este conceput în principal pentru jocuri 3D. Inclusiv instrumentele de proiectare a luminii, iluminare în timp real, sistemele de particule, spații de coordonate 3D etc. - toate acestea sunt legate de jocurile 3D. Prin urmare, crearea de jocuri 2D în Unity aduce unele probleme și obstacole care nu există în multe programe concurente. Acest lucru se datorează faptului că Unity nu a fost destinat încă de la început realizării unor astfel de aplicații. În orice caz, Unity este un instrument atractiv de producție de jocuri 2D, pentru că acceptă o gamă largă de platforme de joc populare și permite creatorilor să integreze multe dintre caracteristicile 3D ale Unity în cele 2D pentru a le îmbunătăți aspectul. Așadar, nu numai că este posibilă dezvoltarea jocurilor 2D în Unity, așa cum o demonstrează aplicațiile prezentate mai sus, dar Unity ne oferă și posibilitatea de a amesteca și integra funcționalitatea 3D existentă și în

jocurile 2D. Acestea toate pot fi utilizate în mod creativ pentru a îmbunătăți calitatea jocurilor noastre 2D. (Thorn, n.d.)

Principalele directoare pe care utilizatorii le vor utiliza constant în timpul lucrului cu platforma Unity sunt: Assets, Library și ProjectSettings. În Assets, sunt salvate toate elementele importate. Se pot crea subfoldere în interiorul acestui folder, se pot șterge și adăuga elemente, iar schimbările se încarcă automat în editorul Unity. Library este un folder ce conține toate fișierele temporare care se creează în momentul în care se apasă *play* în editorul Unity. ProjectSettings este folderul care conține setările personalizate pentru proiect; este vorba de setări precum o nouă valoare pentru gravitație, de exemplu.

GameObjects

Putem crea un GameObject apăsând click dreapta în Ierarhie, apoi *Create empty*. Acestuia îi pot fi atribuite o etichetă (eng. *tag*) și un *layer*. În general, eticheta este utilizată pentru a descrie o proprietate/caracteristică pentru obiectul creat, în timp ce categoria *Layers* este utilizată pentru *raycasting*¹⁸, *rendering*¹⁹ și *linecasting*²⁰.

De exemplu, am putea crea o etichetă cu denumirea *InamicFinal* simbolizând faptul că obiectul creat este ultimul oponent al jocului și am putea să îi asociem și un layer *Inamic*, acesta din urmă definind un grup mai mare de obiecte care împart acest layer și determinând o caracteristică pe care toate aceste obiecte definite o posedă.

Components²¹

Comportamentul obiectelor este definit prin elemente individuale numite Components. Aceste elemente reprezintă o bucată de cod care face un singur lucru. Unity are o serie de componente predefinite care se ocupă de controlarea mișcărilor personajelor, de fizica din joc ș.a.m.d.

Prin definirea unui script (sau a mai multora), utilizatorul poate crea componente cu scopul de a determina un anumit comportament al obiectului. De exemplu, toate obiectele au o

¹⁸ Termen în engleză utilizat în limbajul de specialitate în dezvoltarea jocurilor video

¹⁹ Termen în engleză utilizat în limbajul de specialitate în dezvoltarea jocurilor video

²⁰ Termen în engleză utilizat în limbajul de specialitate în dezvoltarea jocurilor video

²¹ ro. *componente*

componentă numită Transform²² prin modificarea căreia obiectele pot fi mutate. Aceasta determină poziția, rotația și dimensiunea în lumea jocului pentru GameObject.

Adăugarea unei noi componente se poate realiza prin selectarea *Add Component* din Inspector. Astfel se poate selecta componenta dorită din cele puse la dispoziție de către Unity sau se poate crea un script pentru a determina comportamentul dorit al obiectului.

Sprites²³

În Unity, un sistem numit *sprite rendering* face posibilă redarea succesivă a imaginilor 2D, ceea ce, întocmai ca în cazul desenelor animate, creează senzația de mișcare. Acest lucru se poate realiza urmând câțiva pași, după cum urmează. Se poate importa o imagine 2D în folderul Assets, urmând a fi adăugată în Hierarchy²⁴. Înainte de adăugarea imaginii, după metoda menționată anterior, este necesar ca pentru Texture Type să fie selectat *2D and UI*. În acest moment, după ce imaginea este adăugată în Hierarchy, Unity va crea un GameObject căruia îi va asocia imaginea 2D selectată (Sprite). Obiectul va conține SpriteRenderer, o componentă care face ca obiectul să afișeze o imagine 2D. Desigur, această imagine poate fi ulterior înlocuită cu altă imagine 2D.

Crearea animațiilor

Întregul proces pe care îl constituie etapele creării de animații este simplificat în prezent, deoarece Unity pune la dispoziția utilizatorilor un mod prin care pot, direct în Unity, să facă animații, fără a mai fi nevoie ca acestea să fie realizate în alte programe (precum Photoshop, de exemplu), pentru ca apoi să fie importate în motorul de joc.

Unity oferă utilizatorilor, pentru prelucrarea unui sprite sheet²⁵, un sistem intuitiv care eficientizează munca dezvoltatorilor de jocuri video. După importarea unui sprite sheet²⁶ în proiect - era o continuare aici?. Fiecare imagine mai mică din acest sprite sheet reprezintă o poziție în care se află obiectul într-un anumit moment. Pentru a-l anima, GameObject va trece suficient de rapid prin fiecare cadru din imagine și astfel apare senzația de mișcare a elementului pentru care se realizează animația.

²² ro. *mută*

²³ ro. imagine 2D folosită ca obiect în joc (ex. personaje)

²⁴ ro. ierarhie

²⁵ imagine 2D utilizată în crearea jocurilor video

²⁶ imagine 2D care conține mai multe mai multe imagini de dimensiuni mai mici

Odată importată imaginea ce conține cadrele, Texture Type trebuie să fie setat ca *Sprite(2D and UI)*, iar Sprite Mode trebuie schimbat în *Multiple*. Făcând această setare, după apăsarea butonului *Sprite editor* este posibilă identificarea automată și tăierea fiecărui cadru, după apăsarea butonului *Slice* și selectarea *Grid by Cell Count* pentru Type. La finalul acestui proces, după ce sunt aplicate schimbările, în directorul proiectului, după apăsarea săgeții existente pe imagine, apar toate celelalte cadre pe care imaginea inițială le conținea.

Un mod prin care se poate crea o nouă animație este selectarea tuturor cadrelor care alcătuiesc setul de imagini specifice unei anumite acțiuni ce se dorește a fi animată, urmată de plasarea lor în Hierarchy, care determină adăugarea unui nou GameObject, după ce utilizatorul selectează unde dorește să fie salvată animația, ulterior plasării imaginilor. Animația poate fi salvată în Assets. De asemenea, o nouă componentă este adăugată - componenta Animator. Aceasta determină ce animație va fi afișată și când trebuie să se treacă de la o animație la cealaltă. Și în directorul proiectului apar două elemente care au fost create automat, animația creată și Animation Controller, având numele primei imagini adăugate în Hierarchy. După apăsarea butonului *PLAY*, se poate observa animația.

Un alt mod prin care se pot crea animații este prin a selecta *Animation* (se deschide o fereastră în care se poate realiza animația), apoi *Create new*, urmând a se alege unde se dorește a fi salvată animația.

Din directorul Assets, cadrele potrivite care descriu acțiunea ce trebuie animată pot fi aduse pe rând sau împreună în fereastra deschisă. Este permisă manipularea duratei afișării fiecărei imagini, viteza de redare, și pot fi selectate și evenimente, de exemplu, funcții care determină un anumit comportament la finalul redării unei animații.

Physics²⁷ 2D

Pentru înțelegerea fizicii 2D în Unity, este necesară înțelegerea diferenței dintre modul 2D și modul 3D. Diferența majoră este faptul că în modul 2D axa z nu este luată în considerare și rotirea se poate face doar în jurul unei axe care este perpendiculară pe plan. Lumea creată există în continuare într-un spațiu 3D. Toate componentele pentru modul 2D au un corespondent pentru modul 3D, iar acestea nu interacționează, fiind vorba despre sisteme diferite.

²⁷ ro. fizică

Rigidbody2D

Rolul acestei componente este acela de a face GameObject să se supună legilor fizicii 2D. După ce îi este adăugat unui obiect un Rigidbody2D, acesta este afectat de gravitație și de Legile Mișcării ale lui Newton, sunt luate în considerare variațiile forțelor etc.

În mod normal, componenta Transform controlează felul în care se mișcă obiectul. Componenta Rigidbody2D poate suprascrie componenta Transform. Cu toate că se poate înlocui Rigidbody2D modificând componenta Transform (deoarece Unity expune toate proprietățile tuturor componentelor), acest lucru va provoca probleme, cum ar fi GameObjects care trec unul prin celălalt sau comportament imprevizibil al mișcării.

Adăugarea unui Rigidbody2D permite unei imagini 2D (sprite) să se deplaseze într-un mod ce respectă legile fizicii prin aplicarea forțelor definite în script în API. Când componenta adecvată a elementului collider²⁸ este atașată și la sprite GameObject, aceasta este afectată de coliziunile cu alte obiecte din joc aflate în mișcare. Utilizarea fizicii asigură crearea unui comportament al lumii verosimil, contribuind și la scăderea timpului necesar scrierii codului. (Unity Technologies, 2021)

Collider2D

Rolul acestei componente este acela de a face să se întâmple și de a înregistra coliziunile dintre două obiecte care au atașat un Collider2D. Există mai multe tipuri care diferă numai prin forma regiunii unde se realizează ciocnirea: PolygonCollider2D, EdgeCollider2D, BoxCollider2D, CircleCollider2D.

PhysicsMaterial2D

Acest asset, PhysicsMaterial2D, poate fi adăugat în câmpul pentru material din Collider2D. Acesta conține două valori care definesc suprafața materialului: *Friction* și *Bounciness*. Crearea acestuia se realizează dând click dreapta în ProjectPanel, selectând *Create* și *Physics Material 2D*.

²⁸ ro. colizor

Asset Packages

Pachetele Asset sunt o colecție de elemente comprimate și stocate într-un fișier cu extensia .unitypackage.

Asset Package permite exportarea unui subset de assets în Project Panel sub forma unei singure entități care poate fi apoi importată în alte proiecte.

Asset Store conține o bibliotecă de creații ale comunității, dar și creații puse la dispoziție de Unity Technologies - modele, sprite sheets, sunete, proiecte întregi etc.

Prefabs

Prefabs este un construct cu o utilitate crescută. De pildă, este util în cazul unor jocuri în care de-a lungul nivelelor este necesară utilizarea aceluiași personaj, a acelorași inamici, având comportamente identice ca într-un nivel anterior, se poate crea un GameObject unic ce poate fi copiat de fiecare dată și utilizat pentru toate acestea. Se pot crea construind un GameObject în Hierarchy, plasându-l apoi în Project Panel. Această acțiune face o copie, Prefab, a obiectului. Într-o nouă scenă, plasarea înapoi în Hierarchy a obiectului (prefab) va face să apară o copie a nivelului anterior. (Bowey, n.d.)

Cinemachine

Cinemachine este o unealtă oferită de către Unity ce ușurează munca utilizatorilor. Fiind perfecționată de-a lungul mai multor ani, poate susține cutsscenes²⁹, modulele pentru Cinemachine integrând camere pentru mediul 3D, precum și pentru mediul 2D.

Mediul 2D

Cinemachine pentru mediul 2D are o multitudine de caracteristici, de la redare³⁰ ortografică, la încadrare³¹ 2D.

²⁹ secvență animată

³⁰ eng. *rendering*

³¹ eng. *framing*

Custom Blends

Acest comportament fundamental determină relația dintre mai multe camere. Fiind un asset, se pot crea pentru fiecare nivel setări personalizate.

Transposer

Prin transposer se poate seta o cameră care urmărește un obiect, de asemenea se poate decide cum urmărește acel obiect.

Collider

Prin diferite metode puse la dispoziție, collider evită obiectele ce apar în cale.

Confiner

Pentru a nu depăși anumite arii, pentru cameră se poate utiliza confiner.

Follow zoom

În mod dinamic, se asigură că obiectul urmărit este prezentat la mărimea sa potrivită.

2.2 Limbajul C#

Pentru realizarea acestui proiect a fost folosit limbajul de programare C# dezvoltat de Microsoft, binecunoscut și utilizat în industria jocurilor video. În alegerea limbajului s-au avut în vedere mai multe aspecte ce vor fi prezentate în cele ce urmează.

În primul rând, C# prezintă reguli specifice, fapt care ușurează învățarea, deoarece pe măsura scrierii codului, în cazul apariției unor erori, Unity nu permite intrarea în modul de joc până când acestea nu sunt rezolvate.

În al doilea rând, spre deosebire de UnityScripts, în cazul în care se lucrează cu C#, atașarea fișierelor C# de GameObjects nu este necesară, lucru care este de preferat, având în vedere că ușurează munca utilizatorilor.

Un alt aspect important care s-a avut în vedere în alegerea limbajului a fost și numărul resurselor de informare disponibile atât online, cât și sub forma unor cărți, manuale etc. Pentru C# există o multitudine de resurse, iar limbajul nu este dedicat exclusiv dezvoltării de jocuri, așa cum este UnityScripts. Alegerea limbajului C# este motivată așadar de aceste trei aspecte, precum și de faptul că printre jocurile ce sunt sursă de inspirație pentru acest proiect, prezentate în *Capitolul 1*, se numără jocuri ce folosesc acest limbaj. (Norton, 2013)

2.3 Concluzii

Unity este un motor de joc preferat de către mulți dezvoltatori de jocuri datorită versatilității sale, oferind un mediu de lucru adecvat atât pentru jocurile 2D, cât și pentru jocurile 3D. Mai mult decât atât, aplicațiile create în Unity pot fi compatibile cu o gamă variată de platforme. Un alt avantaj îl reprezintă faptul că este o variantă de lucru la care oamenii au acces gratuit. Deși varianta Premium aduce beneficii în plus pentru crearea unui joc, varianta gratuită oferă suficiente opțiuni care susțin obținerea unui produs final complex.

Contribuția propriu-zisă

Capitolul 3

3.1 Prezentarea aplicației

3.1.1 Povestea jocului

Hero's Archetype este povestea unui erou care, atunci când își pierde identitatea, pornește într-o călătorie printr-o lume proiectată de inconștientul său, explorând arhetipul eroului. De data aceasta, eroul are misiunea de a se salva pe sine. După ce și-a pierdut reperele care îl ajutau să se recunoască, rămâne captiv într-un alt trup. Parcursul său devine o metaforă a acceptării faptului că procesul prin care trece este unul de metamorfoză. Pentru a se apropia mai mult de sinele său cel nou, trebuie să abandoneze ceea ce credea despre el însuși. Schimbarea este condiția pentru evoluție. Eroul își întâlnește Umbra și alte manifestări ale inconștientului său pe care trebuie să le înfrunte.

3.1.2 Descriere aplicație

Hero's Archetype este un joc 2D pe platformă pentru sistemul de operare Windows, fiind structurat în niveluri. Fiecare nivel aduce noi provocări pentru jucător, acesta putând explora în voie, fără a fi constrâns de o limită de timp.

Interacțiunea cu utilizatorul se realizează destul de subtil. Butoanele pentru mișcare stânga-dreapta, pentru atac, precum și butonul pentru sărit sunt cele clasice pentru jocurile pentru calculator. Utilizatorul are posibilitatea să exploreze de unul singur pentru a determina,

de exemplu, că săritura se realizează apăsând tasta *space*³². Este de evitat subestimarea abilităților utilizatorilor. La începutul jocului, se deschide o fereastră pentru meniu care îi permite jucătorului să înceapă jocul direct ori să afle mai multe informații despre joc sau să modifice setări pentru sunet; de asemenea, tot din meniu se poate opta și pentru părăsirea jocului.

Asemenea multor altor jocuri 2D, proiectul ce face obiectul acestei lucrări pune accentul pe oferirea utilizatorilor a posibilității de explorare, a unei povești atractive, punând la încercare perspicacitatea și îndemânarea jucătorilor.

3.1.3 Prezentarea personajului controlat de utilizator (Player)

Jocul fiind inspirat de *Hollow Knight*, pentru player s-a ales același design ca în jocul menționat anterior. Deși aspectul este simplu, tocmai această abordare îi conferă o latură ce invită la explorare și evocă prin maniera în care animația este concepută o contopire între ludic și mister. Pentru fluxul jocului, prezența unei astfel de imagini dă sens atât premisei poveștii, cât și modificării manierei integrării sale în universul jocului prin variația gravitației, a forței săriturilor ș.a.m.d.

Pentru a putea crea o experiență cât mai realistă în timpul jocului, s-au atașat jucătorului componentele *Rigidbody2D* și *Collider2D*, *SpriteRenderer* și *Animator*. Odată asigurată influența legilor fizicii asupra obiectului și fiind setată premisa coliziunii cu alte obiecte, următorul pas este configurarea mișcărilor pe care acesta le poate face în lumea jocului. Aceasta este o etapă importantă din crearea unui joc, este un moment crucial, fiind necesară adaptarea mișcărilor la mediul nou-creat. Pentru acest lucru, avem nevoie de animații pentru fiecare acțiune în parte, de conceperea unui sistem de cauzalitate între atacurile suferite și atacurile inițiate și, respectiv, evidențierea deteriorărilor produse. Totodată, e necesar ca jucătorul să se mențină pe platformă întotdeauna pentru a putea continua jocul.

Identificăm patru stări în care jucătorul s-ar putea afla: în așteptare, în mișcare (mers/alergat), sărind sau atacând. Fiecare dintre acestea are nevoie de o animație separată.

Animațiile au fost realizate folosind procesul descris în *Capitolul 2*. Pentru redarea la momentele potrivite, în *Animator* a fost concepută schema acestora, iar în cod au fost stabilite condițiile de redare a fiecărei animații.

Setări prestabilite pentru controlul personajului:

³² ro. spațiu

1. Gravitație: 38
2. Viteză săritură: 17
3. Viteză mișcare: 8

3.1.4 Tilemap

Pentru construirea unui nivel s-a folosit Tile Palette (rezultatul în *Fig. 5*), un instrument pus la dispoziție de Unity. Acesta face ușoară construirea platformei de către utilizatori chiar în motorul de joc. S-a optat pentru atașarea unui Rigidbody2D, a unui Collider2D și a unei componente Composite Collider 2D, pentru ca interacțiunea dintre personajele din joc și platformă să fie cât mai realistă.

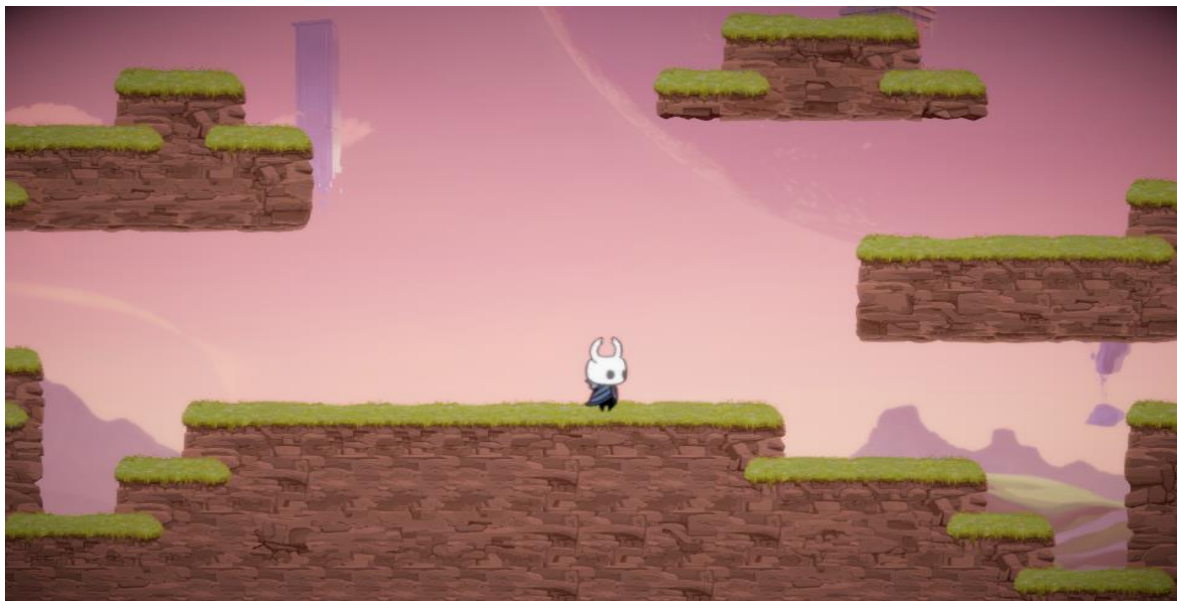


Fig. 5

3.1.5 Platformă care se mișcă

Aceasta conține un script pentru mișcarea acestuia, un Collider2D, Rigidbody2D și un script care permite obiectelor situate pe platformă să se miște.

În Scene View, se stabilește traiectoria mișcării platformei.

Un alt aspect controlat este viteza acesteia de mișcare.

3.1.6 Planurile și efectul de paralaxă

Cele două planuri utilizate în toate nivelurile jocului sunt cel mai din față, al platformei și planul din spate. Adăugarea celui de-al treilea plan într-un nivel evidențiază efectul de

paralaxă, astfel încât vitezele de mișcare ale planurilor, fiind diferite, întăresc senzația de adâncime și generează efectul de paralaxă.



Fig. 6

3.1.7 Armament/ Atac/ Luptă

Personajul posedă o singură armă (*Fig. 7*) pe care o folosește în momentul atacului. Mai exact, este redată o animație cu acesta utilizând arma în timp ce interacționează cu inamicul pe care îl și lovește, provocându-i deteriorarea. Se determină căror elemente li se poate aplica deteriorarea în urma atacului, iar acestea suferă efectul coliziunii în consecință.

Jucătorul poate ataca toate obiectele care sunt din categoria inamicilor. Concret, fiecare oponent care poartă această etichetă suferă o scădere a punctelor ce indică nivelul său de sănătate.



Fig. 7

3.1.8 Prezentarea unui inamic

Pentru designul fiecărui oponent am preluat elementele necesare din Asset Store.

Ideea principală pentru deplasarea acestuia pe platformă a fost, în cazul unui inamic ce avea ca obiectiv să patruleze o zonă alocată, să se împiedice căderea acestuia de pe platformă. Acest lucru a fost posibil atașându-i componenta Rigidbody2D care îl face să se supună legilor fizicii 2D și un Collider2D. Inamicul se mișcă într-o direcție până când se detectează marginea

platformei, după care se apelează funcția care îl determină să se întoarcă pentru a patrula în cealaltă direcție.

Fiecare inamic posedă un anumit număr de puncte ca indice al sănătății acestora, iar numărul scade în momentul în care acesta este atacat. Când un inamic rămâne fără puncte pentru sănătate, acesta moare, iar GameObject-ul corespunzător este distrus.

Un inamic de asemenea poate ataca jucătorul când se apropie de acesta, cauzând scăderea punctelor jucătorului. În cazul consumării tuturor punctelor, jucătorul este obligat să repete acea etapă a jocului.

3.1.9 Dialogue System

Deși acesta nu este un sistem de dialog în sensul clasic, s-a utilizat TextMeshPro (oferit de Unity) pentru crearea unor puncte cheie în călătoria personajului (ex. bucăți de poveste).

Capitolul 4

4.1 Aprecierile personale privind relevanța rezultatelor obținute

Dincolo de punerea în scenă a totalității obiectelor ce constituie lumea virtuală a jocului prin cunoștințe și abilități de programare, am reușit să creez o atmosferă cu un caracter unic ajutându-mă de elemente artistice - sunet, poveste și elemente vizuale -, am construit o lume verosimilă, fizica jocului este adaptată corespunzător fiecărui nivel, interacțiunea dintre personaje și interacțiunea acestora cu mediul se realizează natural datorită punerii în valoare a funcțiilor motorului de joc, au fost studiate aspectele apreciate în această industrie, rezultatul fiind o aplicație care satisface atât interesul utilizatorilor față de creativitate și descoperire, cât și interesul acestora pentru redescoperirea unor aspecte familiare de game design și apreciate de către publicul larg.

4.2 Obiective pentru viitor

Jocul permite o destul de mare flexibilitate și oferă o multitudine de posibilități pentru dezvoltare datorită premiselor inițiale inedite.

- Unul din obiectivele de viitor pentru dezvoltarea jocului este introducerea unei noi arme;
- Îmbunătățirea graficii, adăugarea unor desene realizate de către mine;
- Rezolvarea întrebării „ce se întâmplă cu eroul când moare?”; de exemplu, o umbră a sa se poate materializa în locul respectiv și să devină inamicul său când jucătorul revine în acea zonă;
- Introducerea mai multor personaje;
- Dezvoltarea poveștii;
- Introducerea posibilității realizării unor sărituri multiple;
- Crearea opțiunii pentru personaj de a se prinde cu mâinile de platformă pentru a se cățăra în locurile greu accesibile.

Concluzii

Hero's Archetype, deși este încă un demo, prin setarea unor premise inedite pentru universul virtual, îmbinarea de elemente promovate în cele mai de succes jocuri lansate în ultimii ani, fiind conceput după o atentă analiză atât a psihicului uman, cât și a tendințelor industriei jocurilor, întrunește condițiile pentru a putea fi considerat un joc modern, cu potențial mare de creștere pe viitor.

S-a urmărit satisfacerea așteptărilor utilizatorilor atât din punct de vedere vizual, cât și din punct de vedere al mecanicilor jocului. Există și este din ce în ce mai vizibilă tendința de a crea continuări sau universuri alternative pentru multe jocuri consacrate (fie că vorbim despre jocuri de la Nintendo, precum *The Legend of Zelda: Breath Of The Wild* și seria de jocuri *Mario* sau despre jocurile indie precum *Hollow Knight*), motiv pentru care se pot face două observații: pe de o parte, seriile cunoscute de jocuri promovează această activitate economică, dar și ramura aceasta de cercetare, invitând tot mai mulți oameni să se implice, iar pe de altă parte constatăm faptul că aceste alegeri sunt apreciate de un număr mare de oameni. Datorită legăturii tematice și conceptuale cu *Hollow Knight*, prezenta lucrare nu doar că își împlinește scopul de a fi un proiect de sine stătător, însă prin legăturile nuanțate pe care le face, omagiind un joc deja consacrat, are potențialul de a fi descoperit mult mai ușor de publicul țintă.

Așa cum am arătat în *Capitolul 1*, fiecare joc își aduce contribuția pentru creșterea industriei jocurilor; în timp ce unele jocuri conduc la o creștere exponențială pentru interesul față de o anumită categorie de jocuri, altele devin pilonii ce susțin viitoarele proiecte ale altor dezvoltatori de jocuri. Pentru imaginea de ansamblu a istoriei, fiecare contribuție își dezvăluie rolul odată cu trecerea timpului, dovada fiind influențele jocurilor extrem de apreciate de la finalul secolului XX, care încă rămân pregnante.

În *Capitolul 2* am analizat subiectul acestui proiect din punct de vedere tehnic, listând principalele tehnologii folosite și prezentând utilitatea fiecăreia dintre acestea.

Pe parcursul *Capitolului 3* am prezentat proiectul realizat în raport cu sursele de inspirație menționate anterior și cadrul teoretic al dezvoltării acestei aplicații.

Acestea fiind spuse, lucrarea de față își aduce contribuția la creșterea progresivă a domeniului de cercetare a dezvoltării jocurilor pentru calculator, punând la dispoziție și un model de aplicație ce susține modelul teoretic.

Bibliografie

- Anthropy, A., & Clark, N. (2014). *A game design vocabulary. Exploring the foundational principles behind good game design.* Addison-Wesley.
<https://ptgmedia.pearsoncmg.com/images/9780321886927/samplepages/0321886925.pdf>. Ultima accesare: 06.06.2021
- Arnott, L. (2017). Mapping Metroid: Narrative, Space, and Other M. *Games and Culture*, 12(1).
<https://ir.lib.uwo.ca/cgi/viewcontent.cgi?article=1127&context=fimspub>. Ultima accesare: 06.06.2021
- Bissell, T. (2010). *Extra lives: Why video games matter* (r. New York: Pantheon ed.).
- Bowey, J. (n.d.). *Learning to use the Unity 2D game engine.*
- Dillon, R. (2018). *On the way to fun.* CRC Press.
- Fanfarelli, J. R. (2014). *The effects of Narrative and achievements on learning in a 2D platformer video game.* Florida.
- Geig, M. (2021). *Unity game development in 24 hours.* SAMS TEACH YOURSELF.
- Glancey, P. (1996). *The complete history of computer and video games.*
- Griffiths, M. (2002). The educational benefits of video games. *Education and health*, 20(3).
https://www.researchgate.net/publication/284491180_The_educational_benefits_of_videogames. Ultima accesare: 06.06.2021
- Gutierrez Rodriguez, A., Cotta, C., & Fernández Leiva, A. (n.d.). An Evolutionary Approach to Metroidvania Videogame Design. *Conferencia de la Asociación Española para la Inteligencia Artificial*.
https://sci2s.ugr.es/caepia18/proceedings/docs/CAEPIA2018_paper_246.pdf. Ultima accesare: 06.06.2021
- Moon Studios. (n.d.). <https://www.orithegame.com>. Ultima accesare: 06.06.2021
- Norton, T. (2013). *Learning C# by developing games with unity 3D beginner's guide.* Birmingham: Packt Publishing Limited.
- Oliveira, B., Artur de Oliveira da Rocha Franco, & Jos'e Wellington Franco da Silva. (2020). A Framework for Metroidvania Games. *SBGames*.
- Smith, G., Cha, M., & Whitehead, J. (n.d.). *A Framework for Analysis of 2D Platformer Levels*.
https://www.researchgate.net/publication/229039146_A_framework_for_analysis_of_2D_platformer_levels. Ultima accesare: 25.05.2021

Team Cherry. (n.d.). *Hollow Knight*. <https://www.hollowknight.com>, Ultima accesare: 25.05.2021

Thorn, A. (n.d.). *Learn Unity for 2D Game Development*. Apress. [https://www.academia.edu/28371929/Learn Unity for 2D Game Development TI](https://www.academia.edu/28371929/Learn_Unity_for_2D_Game_Development_TI), Ultima accesare: 25.05.2021

Unity Technologies. (n.d.). <https://unity.com/unity/features/editor/art-and-design/cinemachine>. Ultima accesare: 25.05.2021

Unity Technologies. (n.d.). *Multiplatform*. <https://unity.com/features/multiplatform>. Ultima accesare: 25.05.2021

Unity Technologies. (2021). <https://docs.unity3d.com/Manual/class-Rigidbody2D.html>. Ultima accesare: 25.05.2021

Elemente din Asset Store utilizate

Pentru o bază estetică:

[2D Game Kit | Tutorials | Unity Asset Store](#). Ultima accesare: 25.05.2021

Altele:

<https://assetstore.unity.com/packages/2d/gui/fantasy-wooden-gui-free-103811>. Ultima accesare: 01.06.2021

Restul elementelor utilizate au fost preluate direct de la creatorii de conținut.