

UNIVERSITATEA „ALEXANDRU IOAN CUZA”
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

Coordonator științific

Lector Dr. Adrian Iftene

Absolvent

Niță Iulian-Alexandru

IAȘI

2013

UNIVERSITATEA „ALEXANDRU IOAN CUZA”

FACULTATEA DE INFORMATICĂ

WORMS

Joc 2D pentru platforma Android

Coordonator științific

Lector Dr. Adrian Iftene

Absolvent

Niță Iulian-Alexandru

IAȘI

2013

DECLARAȚIE PRIVIND ORIGINALITATE ȘI RESPECTAREA DREPTURILOR DE AUTOR

Prin prezenta declar că Lucrarea de licență cu titlul “Worms – Joc 2D pentru platforma Android” este scrisă de mine și nu a mai fost prezentată niciodată la o altă facultate sau instituție de învățământ superior din țară sau străinătate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referința precisă a sursei;
- reformularea în cuvinte proprii a textelor scrise de către alți autori deține referința precisă;
- codul sursă, imagini etc. preluate din proiecte open-source sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;
- rezumarea ideilor altor autori precizează referința precisă la textul original.

Iași, 03.07.2013

Absolvent Niță Iulian-Alexandru

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul “Worms – Joc 2D pentru platforma Android”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea Alexandru Ioan Cuza Iași să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, 03.07.2013

Absolvent Niță Iulian-Alexandru

Cuprins

Introducere	6
Capitolul 1. Platforma de dezvoltare Android	9
1.1 Aplicatii populare	10
1.2 Worms	14
1.2.1 Scurt istoric	14
1.2.2 Descriere	15
1.2.3 Versiunile pentru platforma Android	15
1.3 Obiectiv	18
1.4 Concluzii	18
Capitolul 2. Tehnologii folosite.....	20
2.1 Sistemul de operare Android	20
2.2 AndEngine	21
2.3 Box2D	22
2.4 Physics Body Editor	23
2.5 Concluzii	24
Capitolul 3. Prezentarea aplicației	25
3.1 Descriere și gameplay	25
3.2 Mediul virtual	28
3.3 Armament	30
3.4 Concluzii	32
Capitolul 4. Inteligența artificială	33
4.1 Obiective	33

4.2	Algoritmul A*	34
4.3	Strategii	37
4.4	Balansarea dinamică a dificultății	39
4.5	Concluzii	40
	Concluzii finale	41
	Bibilografie	42

Introducere

Considerate în jurul anilor 1950-1960 doar o simplă curiozitate în lumea calculatoarelor, jocurile video au reușit, din 1970 când au atins popularitate la scară largă și până în prezent, să contruiască în jurul lor una dintre industriile de top ale economiei mondiale.

Fiind disponibile pentru o multitudine de platforme precum calculatoare personale, console sau dispozitive mobile, jocurile video sunt alese de sute de milioane de persoane din întreaga lume ca modalitate de petrecere a timpului liber (Fig.1). Astfel, industria jocurilor video se află într-o continuă dezvoltare și ajunge la câștiguri anuale de zeci de miliarde de dolari. După cum sugerează și (Fig.2) aceste câștiguri au tendința de a crește de la an la an, principalele motive fiind progresul constant al tehnologiilor din industria calculatoarelor, dar și numărul tot mai mare al persoanelor care intră în contact cu această lume.

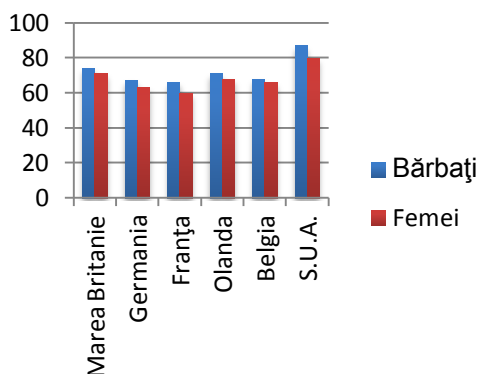


Fig.1 Procente din populație care aleg jocurile video ¹

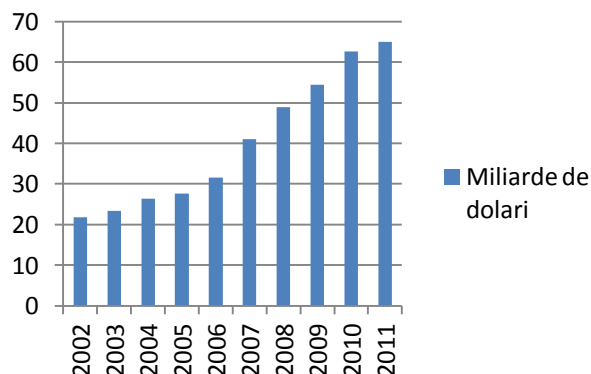


Fig. 2 Câștigurile anuale ale industrie jocurilor video

Deși fiecare platformă pentru dezvoltarea jocurilor video are un rol important și bine definit, atenția ne este atrasă de dispozitivele mobile, în special smartphone-uri și tablete, a căror ramură a cunoscut de-a lungul ultimilor ani o expansiune extraordinară.

1. Sondaj online realizat de *Today's Gamers* pe un eșantion de 13000 de persoane din țările menționate

De la 17 ani de la lansarea primului smartphone modern de către compania Nokia (modelul Nokia Communicator¹) și până în zilele noastre (a.c. 2013) numărul de dispozitive mobile inteligente a depășit pragul de un miliard la nivel mondial, creștere care a adus odată cu ea și explozia pieței jocurilor video dedicate acestor dispozitive mobile.

Pe lângă numărul mare de dispozitive și diversitatea categoriilor de jocuri disponibile, această explozie se datorează și avantajelor pe care această industrie le are asupra “surorilor” sale (industria jocurilor pentru calculatoare personale sau pentru console), atât din punct de vedere al utilizatorului final, cât și din punct de vedere al dezvoltatorilor de jocuri. Printre aceste avantaje se numără [1]:

- dimensiunea redusă a dispozitivelor care permite utilizatorilor să se bucure de aplicații în orice loc s-ar afla;
- modul de interacționare foarte intuitiv și natural dintre dispozitiv și utilizator prin intermediul ecranelor tactile sau accelerometrului;
- existența așa numitelor market-uri care facilitează din punct de vedere al utilizatorului achiziționarea jocurilor și din punct de vedere al dezvoltatorului publicarea acestora;
- numărul mare de unelte de dezvoltare pentru acest tip de aplicații mobile;
- prețul accesibil al aplicațiilor.

Astfel, nu este de mirare că un studiu realizat recent de compania *Flurry Analytics*² relevă faptul că jocurile sunt aplicațiile cele mai folosite de către deținătorii de dispozitive mobile inteligente. (Fig.3)

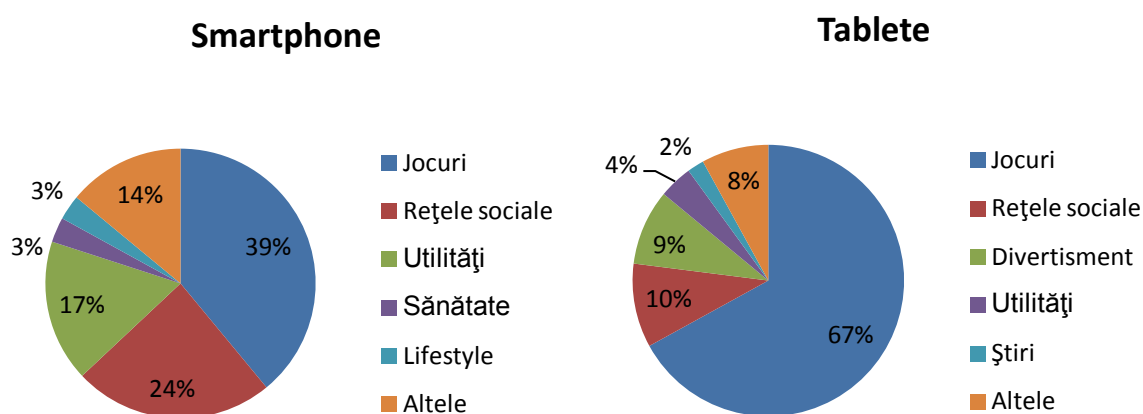


Fig.3 Cele mai folosite aplicații pe dispozitivele mobile inteligente

1. http://en.wikipedia.org/wiki/Nokia_Communicator
 2. <http://www.flurry.com/flurry-analytics.html>

Toate aspectele menționate mai sus date duc la concluzia că dispozitivele mobile inteligente au o influență mare asupra industriei jocurilor video și vor avea parte de un viitor îndelungat în cadrul acesteia.

Influențat de tendințele pieței prezentate anterior, îmi propun prin intermediul acestei lucrări implementarea unei astfel de aplicații pentru platforma mobilă Android. Aplicația, inspirată dintr-o serie de jocuri video populară, se încadrează în categoria strategie turn-based și oferă posibilitatea utilizatorilor de a-și testa îndemânarea atât împotriva unui jucător controlat de calculator, cât și împotriva unui alt utilizator uman în modul hotseat (dispozitivul este controlat pe rând de fiecare jucător). Accentul este pus pe oferirea unei experiențe de joc bogată și intuitivă, dar și pe dezvoltarea unui algoritm robust de inteligență artificială care să reușească să pună jucătorii în dificultate.

În Capitolul 1 este prezentat pe scurt mediul de dezvoltare mobilă Android, este realizat un top al celor mai populare jocuri publicate sub această platformă și sunt identificate și analizate soluțiile existente.

Capitolul 2 face o scurtă introducere în lumea dezvoltării de jocuri video și prezintă tehnologiile folosite pentru atingerea obiectivului acestei lucrări. Printre tehnologii se numără sistemul de operare Android, motorul de joc AndEngine și motorul de fizică Box2D.

Capitolul 3 descrie în detaliu toate funcționalitățile aplicației și metodele folosite pentru implementarea acestora.

Capitolul 4 pune accentul pe implementarea algoritmului de inteligență artificială. Sunt prezentate concepte generale ce au fost aplicate în procesul de implementare, modificările și adaptările făcute pentru a se potrivi contextului jocului dezvoltat.

Capitolul 1

Platforma de dezvoltare Android

Android¹ este în prezent cea mai populară și totodată cea mai puternică platformă de dezvoltare mobilă, ea fiind nucleul a sute de milioane de dispozitive mobile răspândite în întreaga lume. Prin colaborări cu experți de nivel mondial în domeniul software și hardware, platforma Android se află într-o continuă dezvoltare în încercarea de a oferi atât utilizatorilor, cât și dezvoltatorilor, performanțe din ce în ce mai ridicate și noi capabilități funcționale.

Din punct de vedere al dezvoltatorilor, Android pune la dispoziție o serie de unelte performante (SDK, debugger, IDE, etc.) și un model unic de aplicație, facilitând astfel crearea de aplicații care să profite din plin de capabilitățile software și hardware pentru o gamă largă de dispozitive. Din punct de vedere al utilizatorilor, Android oferă o experiență bogată prin design-ul și modul de interacțiune extrem de intuitiv, ușurința cu care se realizează multitasking, multitudinea de posibilități de customizare a dispozitivului sau modalitățile de stocare și sincronizare a datelor personale (cloud computing) [1].

Ca mai toate platformele de dezvoltare mobilă moderne, Android oferă utilizatorilor și dezvoltatorilor săi un loc în care aceștia pot, cu ușurință, regăsi, achiziționa sau publica aplicații. Cunoscut în momentul actual sub numele de *Google Play*², market-ul platformei Android pune la dispoziție o listă numeroasă de categorii de aplicații, aceasta întinzându-se de la cele cu iz financiar sau medical până la aplicații multimedia. *Google Play* menține un caracter deschis față de utilizatorii săi permițându-le acestora un control total asupra aplicațiilor personale și a contului Android. Astfel, în fiecare zi mii de persoane vin în contact pentru prima dată cu acesta platformă prin activarea unui dispozitiv mobil Android.

Din multiplele categorii de aplicații existente pe market, ies din nou în evidență jocurile. Acestea par a fi, conform statisticilor realizate de companii specializate în domeniu, tipul preferat de aplicații al utilizatorilor. Astfel, jocurile sunt atât cele mai descărcate cât și cele mai folosite aplicații la nivel de dispozitive mobile. Totodată, un studiu realizat de compania americană *Nielsen*³ relevă faptul că 93% dintre utilizatori ar fi dispuși să plătească pentru achiziționarea unei aplicații de tip joc. (Fig.4)

1. <http://www.android.com/>

2. <https://play.google.com/>

3. <http://www.nielsen.com/>

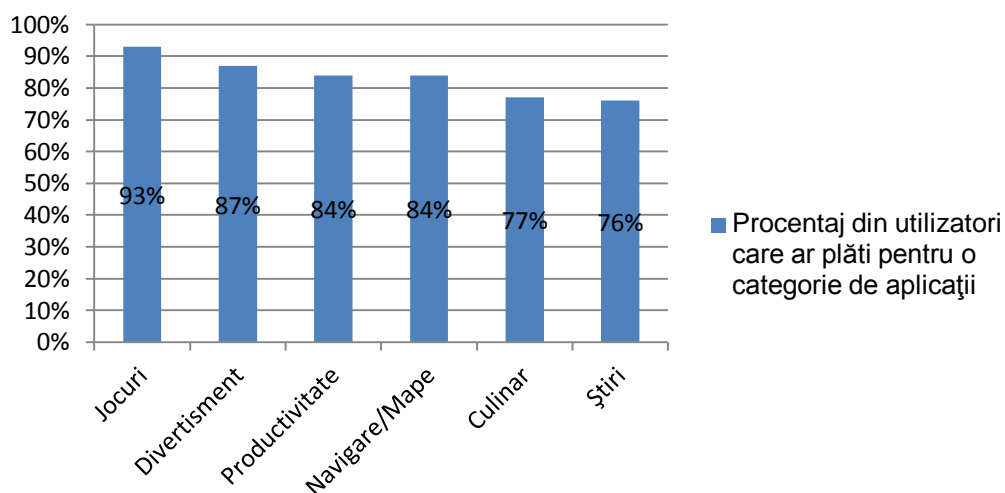


Fig.4 Jocurile domină și din punct de vedere financiar celelalte aplicații

1.1 Aplicații populare

Cu titluri care numără zeci de milioane de descărcări, jocurile sunt cele mai căutate aplicații mobile ale momentului. În continuare vom prezenta pe scurt câteva dintre cele mai populare titluri de jocuri printre utilizatorii de dispozitive mobile inteligente.

Angry birds

Cu șase titluri la activ și peste un miliard de descărcări cumulate pe toate platformele, *Angry birds* este fără îndoială cea mai de succes aplicație mobilă creată până în prezent. Jocul a cucerit prin simplitate, creativitate, personajele carismatice și iscusite, dar și prin fizica din spatele acestuia care aduce o nouă experiență de fiecare dată când este jucat.

Structurat pe nivele, povestea jocului este concentrată în jurul unui stol de păsări care trebuie să își recupereze ouăle furate de o grupare răuvoitoare de porcușori. Controlând o praștie uriașă, jucătorul trebuie să lanseze păsările către grupul de porcușori care sunt protejați de structuri construite din diverse materiale cu scopul de a-i elimina. Fiecare tip de pasăre este eficientă pentru un anumit tip de atac și un anumit tip de material. Astfel este lăsată la latitudinea jucătorului construirea unei strategii perfecte pentru câștigarea nivelului. (Fig. 5,6)

Combinăția de gameplay care conduce la dependență, caractere, stil comic și chiar preț redus au determinat lansarea jocului și pe alte platforme cum ar fi PC sau console unde acesta se bucura de aceeași popularitate. [3]



Fig.5 Pasăre distrugând structura protectivă a porcușorilor ¹



Fig. 6 Tipurile de păsări puse la dispoziție în joc ²

Drag Racing

Cu peste 70 de milioane de descărcări *Drag Racing* a devenit cel mai popular joc de curse pentru dispozitivele mobile. Jocul presupune câștigarea a diverse turnee de curse în linie dreaptă folosind cele peste 50 de modele de mașini disponibile și upgrade-uri pentru creșterea performanței acestora (Fig. 7,8). Cheia jocului este de a găsi echilibrul perfect între cai-putere, control, putere de frânare și accelerație având la dispoziție o anumită sumă de bani, lucru nu foarte facil cum ar părea la prima vedere. Pentru sporirea experienței de joc se dă posibilitatea organizării de curse cu jucători din întreaga lume, formarea de echipe pentru competiții online și împărtășirea de strategii câștigătoare. [4]

Simplitatea controalelor, multitudinea de echipamente și strategii abordabile și experiența online au făcut din *Drag Racing* o aplicație extrem de populară printre deținătorii de smartphone-uri și tablete.



Fig.7 Mașină aflată în cursă. Mașina este controlată doar prin cele 2 pedale de schimbare a vitezei, punctele critice fiind indicate de cadranul de pe mijlocul ecranului ³



Fig.8 Meniul din care mașina poate fi customizată după dorințele jucătorului ⁴

1. http://www.rovio.com/UserFiles/Image/RovioProducts//380x286_48.jpg
2. http://www.rovio.com/UserFiles/Image/RovioProducts//380x286_104.png
3. <https://lh6.ggpht.com/g84rWJeawTzqO3Y1xssR2FqRw8MfbBNxs10c1Ns7ILR3-UQoz6quPktmGWZJOqs0J9Ue>
4. https://lh4.ggpht.com/q81slIgdIz6sj9XY-ha1x4kyxK60pQfHVSW117t9uSovmWIE_45LWY-a_Ff2DLuv5Y=h230

Fruit Ninja¹

Un alt joc extrem de popular este *Fruit Ninja* care se mândrește cu peste 300 de milioane de descărcări cumulate pe toate platformele mobile. Pe cât de simplu este gameplay-ul pe atât de dependent poate deveni. Scopul jucătorului este ca folosind o sabie controlată prin intermediul ecranului tactil să taie diverse fructe care sunt aruncate în suprafața de joc și să evite în același timp bombele care apar la anumite intervale de timp. Jocul se termină atunci când scapă netăiate trei fructe sau la contactul sabiei cu o bombă capcană. Îndemânarea utilizatorilor poate fi testată și online în meciuri cu alți jucători din întreaga lume, iar spiritul competitiv este pus la încercare prin intermediul topurilor celor mai buni jucători (Fig. 9). [5]

Popularitatea aplicației a determinat producătorii să realizeze și versiuni pentru alte platforme cum ar fi Windows 8 sau Xbox 360, cea din urmă având parte de o priză mai bună la public datorită modului de joc care necesită folosirea corpului pentru interacțiunea cu jocul prin intermediul tehnologiei Kinect.



Fig. 9 Captură din timpul jocului în care se observă fructele și bomba capcană²

Subway surfers

Deși are în spate un concept simplist, *Subway surfers*³ a reușit să cucerească utilizatorii de dispozitive mobile, având la numai un an de la lansare peste 50 de milioane de descărcări. Jocul este practic o cursă “infinită” cu obstacole în care scopul jucătorului este să parcurgă, ajutat uneori de “super puteri”, o distanță cât mai mare (Fig. 10, 11). Grafica impresionantă de care beneficiază și posibilitatea de a concura cu jucători din toată lumea pentru depășirea de recorduri fac din acest joc rețeta perfectă pentru o aplicație destinată platformelor mobile [6].

1. <http://play.google.com/store/apps/details?id=com.halfbrick.fruitninja>
2. <http://lh6.ggpht.com/xmvuGIyFlegkKN8jdbMHJ5SheCe2Tc2KXV86bU5GWPGGrU16xhQwjMF2MDdynkRcFkcr>
3. <https://play.google.com/store/apps/details?id=com.kiloo.subwaysurf+>



Fig. 10 Caracterul trebuie sa ocolească în vederea atingerii obiectivului ¹



Fig. 11 Caracterul este ajutat de super-puteri pentru a-și atinge obiectivul ²

Where's my water

Având parte de o grafică uimitoare, o simulare fizică foarte fidelă lumii reale și o idee ingenioasă la bază, aplicația *Where's my water* se regăsește și ea în categoria celor mai descărcate jocuri pentru platforme mobile cu peste 100 de milioane de accesări. Făcând parte din genul de jocuri puzzle, acesta pune la încercare abilitățile strategice ale utilizatorilor “forțându-i” pe aceștia să își pună gândirea în mișcare. Jocul are ca scop dirijarea unui volum de apă de-a lungul unui traseu până într-un punct final, evitând diverse obstacole și eventual obținând anumite bonusuri (Fig.12) [7].



Fig. 12 Distrugând porțiuni de teren și evitând obstacolele, utilizatorul trebuie să conducă apa către personajul aflat în partea de jos a imaginii ³

1-2. <https://play.google.com/store/apps/details?id=com.kiloo.subwaysurf> (sursă imagini)

3. http://www.mododojo.com/news/wheres_my_water_downloaded_over_100_million_times_swampy_celebrates_1st_birthday

Analizând cele cinci exemple descrise mai sus observăm că indiferent că este vorba despre un joc strategic, sportiv sau de tip puzzle, există anumite puncte comune care fac dintr-o astfel de aplicație un succes în rândul utilizatorilor. Tendința nu este de a dezvolta jocuri cât mai complexe pentru platformele mobile, ci pornind de la un concept simplu, dar ingenios, să se contruiască o aplicație care să respecte următoarele principii:

- să fie intuitiv și natural din punct de vedere al experienței de joc și al modului de interacțiune cu dispozitivul;
- să conțină personaje carismatice, amuzante însoțite de efecte sonore și animații
- să ofere o gamă largă de variante din care utilizatorul să poate alege în timpul jocului (personaje, ustensile, upgrade-uri etc.);
- să dea posibilitatea utilizatorilor de a experimenta aplicația în mediul online alături de alți jucători;
- întrucât capacitatea unei baterii pentru dispozitivele mobile nu este foarte mare jocurile ar trebui să se poată desfășura în reprize scurte (aproximativ 5 minute) pentru a nu împiedica utilizatorul din îndeplinirea ulterioară a altor activități.

1.2 Worms

Printre numeroasele jocuri existente pentru platforma mobilă Android, regăsim și două titluri care continuă o îndelungată și celebră serie de jocuri video pentru calculatoare personale și console, anume seria “Worms”¹. Cele două jocuri menționate mai devreme sunt “Worms” și “Worms 2: Armageddon”.

1.2.1 Scurt istoric

Ideea jocului aparține lui **Andy Davidson** care a creat prima versiune a acestuia, numită „*Total Wormage*”, pentru a participa la un concurs de programare susținut de revista „*Amiga Format*”. Deși nu a câștigat competiția, Davidson nu s-a dat bătut și în septembrie 1994 la European Computer Trade Show¹ a fost remarcat de compania britanică *Team17* care s-a oferit să îl ajute la îmbunătățirea și publicarea proiectului. Din colaborarea celor două părți a luat naștere în anul 1995 seria „Worms” care s-a bucurat încă de la început de popularitate. Până în prezent seria numără peste 10 titluri (Worms, Worms 2, Worms Armageddon, Worms 3D, etc.) și are milioane de copii vândute în întreaga lume. [8]

1. <http://www.team17.com/>

2. http://en.wikipedia.org/wiki/European_Computer_Trade_Show

1.2.2 Descriere

Jocul este o strategie în stilul turn-based care poate fi jucat în modul singleplayer împotriva calculatorului sau online în modul multiplayer alături de alți utilizatori. Fiecare jucător controlează o “armată” de caractere asemănătoare unor viermi și ajutându-se de un set de arme și ustensile au ca scop eliminarea echipelor adverse. Pentru a diversifica experiența de joc mapele pe care se desfășoară acțiunea sunt generate în mod aleator și pot fi distruse cu ajutorul armelor puse la dispoziție, iar ocazional sunt introduse pachete surpriză pe suprafața acestora. Printre caracteristicile care au făcut acest joc atât de popular se numără: personajele carismatice înzestrate cu voci pițigăiate care fac deliciul jucătorilor, efectele sonore amuzante rulate pe fundal și nu în ultimul rând setul de peste 40 arme și ustensile bizare, creative dar în același timp și eficiente. [8]

1.2.3 Versiunile pentru platforma Android

Ținând cont de faptul că industria jocurilor video și-a îndreptat foarte mult atenția în ultimii ani asupra platformelor mobile, era aproape inevitabil ca o serie atât de populară cum este *Worms* să nu fie portată și la nivelul dispozitivelor mobile inteligente. Astfel, în anul 2011 dezvoltatorii de la *Electronic Arts Nederland BV* lansează primul titlu al seriei „*Worms*” pentru sistemul de operare Android .

Această primă versiune mobilă a jocului se bucură de multe dintre elementele definitorii care au făcut seria atât de celebră. Avem în continuare parte de mape destructibile generate în mod aleator pe care personajele carismatice și pline de umor cu care au fost obișnuiți fanii seriei sunt gata să se confrunte (Fig.13,14). Dialogurile pițigăiate și efectele sonore amuzante își fac și ele locul fiind un mare plus pentru experiența de joc.

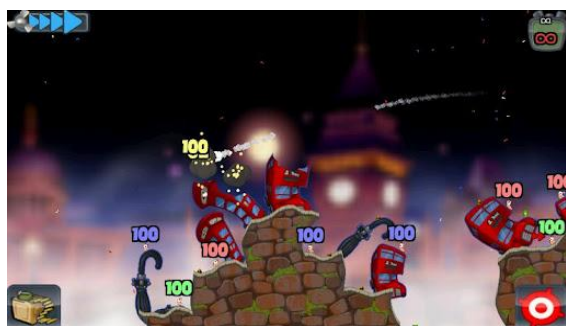


Fig. 13 Mapa generată aleator populată de caracterele gata de confruntare ¹



Fig. 14 Caracterele aflate în plin război; se observă influența exploziei asupra mapei ²

1-2. https://play.google.com/store/apps/details?id=com.ea.worms_row (sursă imagini)

Pentru obținerea victoriei, la dispoziția jucătorilor stă un set de 26 de arme și ustensile printre care se numără și elementele bizare și creative cu care jocul și-a obișnuit fanii. Dintre acestea amintim arme precum Banana Bomb, Holy Hand Grenade, The Sheep sau Concrete Donkey (Fig.15).



Fig. 15 Armamentul pus la dispoziția jucătorilor

Deși se bucură de multe din elementele definitorii ale seriei “Worms” și are potențialul unei aplicații de succes, lipsurile și limitările de care suferă această versiune fac din ea un produs nefinisat, fapt ce explică și rating-ul scăzut de care a avut parte aplicația (Fig. 16).

Ca mai toate aplicațiile pentru platforma Android, jocul “Worms” este disponibil pentru download pe Google Play în schimbul sumei de aproximativ 2 dolari. Analizând tendința prețurilor pe market se poate spune că prețul nu este tocmai potrivit pentru o aplicație cu multe goluri. Secțiunea de comentarii din cadrul Google Play dedicată jocului confirmă ideea prețului ridicat prin o parte din utilizatori care își exprimă regretul față de achiziționarea produsului pentru această sumă.

Versiunea pentru PC sau console a fost criticată de-a lungul timpului pentru lipsa de noutăți aduse jocului. Astfel, era de așteptat ca lansarea jocului pentru smartphone-urile Android să aducă odată cu ea noutăți în privința setului de arme și ustensile sau în privința stilului de joc. Modificările însă sunt aproape inexistente, iar în ceea ce privește armamentul pus la dispoziție, acesta este limitat față de celelalte versiuni ale jocului.

Pentru un utilizator care vine în contact pentru prima oară cu acest joc înțelegerea gameplay-ului și a modului de funcționare al armelor și ustensilelor este foarte important. De aceea tutorialele de îndrumare a utilizatorului sunt extrem de binevenite. Totuși, această versiune a jocului nu vine foarte mult în ajutorul utilizatorului oferind doar tutoriale text însoțite de imagini explicative care nu sunt întotdeauna îndeajuns pentru o bună înțelegere a jocului.

Tot în secțiunea de comentarii de pe Google Play vedem că jocul nu are o funcționare corectă pe tot parcursul folosirii acestuia. Astfel, utilizatorii reclamă lucruri precum: dispariția

sau funcționarea eronată a unor butoane din cadrul jocului, controlul îngreunat al caracterelor, închiderea forțată a aplicației sau imposibilitatea rulării acesteia pentru unele versiuni de sisteme de operare Android. Nu tocmai un comportament pe care o aplicație “finală” achiziționată contracost ar trebui să îl aibă.

De departe cel mai mare gol al aplicației este lipsa modului multiplayer. Deși beneficiază de opțiunea “multiplayer”, aceasta permite mai multor utilizatori să se confrunte doar de pe același dispozitiv mobil, modalitate ce limitează experimentarea jocului în compania altor jucători. Acest lucru aduce un minus semnificativ experienței per-totale a aplicației.

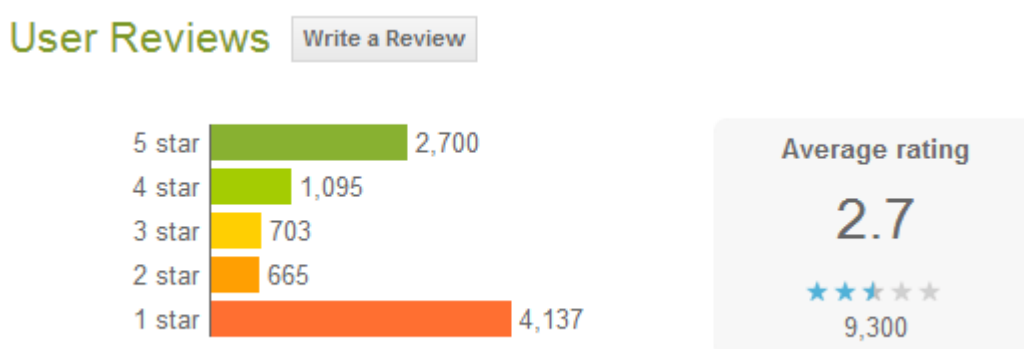


Fig. 16 Aplicația are parte de un rating dezamăgitor¹

Nemulțumirile fanilor seriei nu au rămas însă neascultate. Astfel, în luna aprilie a anului 2013 apare un nou titlu al seriei, el purtând numele „Worms 2: Armageddon”. În acest nou joc sunt rezolvate multe din problemele existente anterior și sunt aduse îmbunătățiri substanțiale experienței de joc, păstrându-se totodată elementele care au făcut seria celebră. Dintre acestea enumerăm [9]:

- rezolvarea problemelor de interacțiune dintre utilizator și dispozitiv;
- introducerea unor noi stiluri de joc și posibilități de customizare a experienței de joc;
- un nou set de arme specifice seriei;
- îmbunătățirea semnificativă a graficii (Fig. 17);
- cea mai importantă adădire este cu siguranță implementarea modului multiplayer care permite utilizatorilor să își testeze abilitățile strategice online alături de alți pasionați ai jocului.

1. https://play.google.com/store/apps/details?id=com.ea.worms_row

Toate aceste plusuri ale noii versiuni ale jocului au impresionat atât fanii seriei, cât și criticii din domeniu, ambele categorii având de spus doar lucruri pozitive la adresa aplicației. Acest lucru se poate observa în rating-ul mult mai mare față de versiune anterioară pe care jocul îl are și în numărul de descărcări de care aplicația a avut parte (peste 500.000 la câteva luni de la lansare) (Fig. 18).



Fig.17 Noul joc are parte, pe lângă elementele definitorii seriei, de o grafică impresionantă ¹

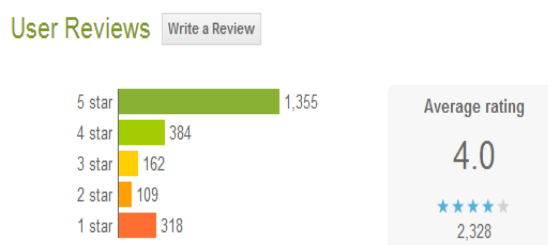


Fig. 18 Rating-ul aplicației pe Google Play ²

1.3 Obiectiv

Pornind de la conceptul care stă în spatele jocului Worms, această lucrare își propune abordarea dezvoltării unui joc pentru platforma Android care să păstreze, pe cât este posibil, caracteristicile specifice seriei. Accentul va cădea pe implementarea unui algoritm de inteligență artificială robust care să reprezinte o provocare pentru utilizatorul final.

Jocul “Worms” va fi implementat pentru sistemul de operare Android și se va avea la bază motorul de joc AndEngine și motorul de fizică Box2D. Algoritmul de inteligență artificială va folosi tehnici de căutare a drumului optimal, diverse metode de scanare a hărții jocului și balansarea dinamică a dificultății.

1.4 Concluzii

Piața platformelor pentru dispozitive mobile și-a îndreptat atenția în ultimii ani asupra dezvoltării de jocuri video. Au apărut ca rezultat aplicații ca „Angry birds” sau „Fruit Ninja” care au fost achiziționate pe zeci de milioane de dispozitive din întreaga lume. În multitudinea de jocuri existente regăsim și un titlu recent care întrunește toate caracteristicile unei rețete de

1-2. <https://play.google.com/store/apps/details?id=com.worms2armageddon.app> (sursă imagini)

success pentru piața dispozitivelor mobile. Este vorba versiunea mobilă a celebrului joc „Worms” care reușește să adune în doar câteva luni de la lansare sute de mii de descărcări.

Influențat de tendința pieței jocurilor video care se află într-o continuă dezvoltare și de conceptul din spatele seriei „Worms”, am decis să îmi îndrept atenția către implementarea unui joc asemănător celui mai sus menționat având ca țintă sistemul de operare Android, cea mai masivă și performantă platformă mobilă a momentului.

Capitolul 2

Tehnologii folosite

Crearea jocurilor video este un proces complex care poate dura de la câteva zile la câțiva ani și poate implica de la un singur dezvoltator până la o companie întreagă. Printre principalele etape ale acestui proces se numără: găsirea unui concept de bază, producția, crearea nivelelor și a resurselor grafice/audio și testarea jocului.

Fie că este vorba despre un joc creat de un dezvoltator independent sau de o companie din domeniu, în dezvoltarea unor astfel de aplicații intervin numeroase unelte și tehnologii care ajută și facilitează procesul. Cele mai folosite unelte software sunt: motoarele de joc (AndEngine¹, Unity², etc.), motoarele de fizică (Box2D³, Bullet⁴, Physz⁵, etc.), editoarele de nivele sau convertoarele de resurse.

Pentru dezvoltarea jocului „Worms” au fost folosite următoarele unelte și tehnologii:

- sistemul de operare Android
- motorul de joc AndEngine
- motorul de fizică Box2D
- editorul pentru obiecte fizice Physics Body Editor

2.1 Sistemul de operare Android

Android este un sistem de operare open source deținut de compania americană Google și destinat în principal dispozitivelor mobile cu ecrane tactile. Android se compune dintr-un nucleu Linux care stă la baza sistemului, o serie de librării și API de nivel mediu scrise în limbajul C, iar la nivelul aplicațiilor din framework-uri ce conțin librării compatibile cu Java. Limbajul de programare principal în care sunt scrise aplicațiile Android este Java. [10]

Android este un sistem de operare puternic și special optimizat crearea unei game largi de aplicații pentru dispozitivele mobile inteligente. Bazele unei aplicații Android sunt:

- sunt construite din una sau mai multe componente individuale (activități);

1. <http://www.andengine.org/>
 2. <http://unity3d.com/>
 3. <http://box2d.org/>
 4. <http://bulletphysics.org/wordpress/>
 5. <http://physz.ath.cx/>

- fiecare componentă execută o acțiune diferită, are un rol bine definit în cadrul aplicației și poate fi activată individual (chiar și de către alte aplicații);
- componentele permit un management facil în vederea multithreading-ului sau a multitasking-ului;
- conține un fișier numit manifest care declară toate componentele și configurația aplicației;
- fiecare aplicație rulează în sandbox și necesită acordul utilizatorului final pentru folosirea diverselor componente și funcții ale dispozitivului;
- permit o ușoară customizare a aplicației pentru diferite mărimi și modele de dispozitive. [11]

Puterea, viteza și optimizarea sistemului de operare Android¹, dar și faptul că face parte din cea mai mare platformă de dezvoltare mobilă la nivel mondial, au făcut din acesta alegerea perfectă pentru implementarea acestui proiect.

2.2 AndEngine

Un motor de joc oferă dezvoltatorilor o suită de unelte pentru facilitarea creării de jocuri video. Printre funcționalitățile de bază ale unui astfel de sistem se numără: un motor de redare pentru grafică 2D sau 3D, managementul efectelor sonore și al animațiilor, controlul asupra scenelor jocului sau comunicare în rețea.

AndEngine² este un motor de joc creat de Nicolas Gramlich pentru a ușura munca dezvoltatorilor de jocuri 2D pentru platforma Android. Fiind un proiect open source el este primește constant îmbunătățiri și noi funcționalități de la zeci de pasionați din întreaga lume. Bazat pe OpenGL, AndEngine oferă un nivel ridicat de abstractizare și acoperă toate necesitățile majore dezvoltării un joc 2D. Este rapid, bine organizat și implementat, printre funcționalitățile sale numărându-se scalarea pentru toate modelele de dispozitive mobile, suport pentru multitouch, comunicare în rețea, management al scenelor și entităților din joc, sisteme de particule etc.

Conceptele specifice jocurile video 2D acoperite de motorul AndEngine sunt:

- **Camera:** entitate abstractă care determină ce porțiuni din lumea virtuală creată în joc este prezentată utilizatorului; suportă acțiuni precum translația sau zoom-ul;
- **Scena:** entitate care leagă toate elementele unui joc formând lumea virtuală expusă utilizatorului;

1. <http://www.android.com/>

2. <http://www.andengine.org/>

- **Sprite:** reprezentarea vizuală a unei entități ce poate fi desenată precum un buton, un caracter din joc, un meniu, o hartă, etc.;
- **Modificatori:** modifică în timp diverse proprietăți ale entităților (culoare, rotație, scalare, etc.);
- **Motorul** propriu-zis: acesta este creierul sistemului el având ca atribuții expunerea scenei către utilizatorului, updatarea modificatorilor și animațiilor, desenarea entităților pe ecran și cu input-ul primit de la utilizatori. [12]

Nivelul ridicat de abstractizare și mulțimea de funcționalități și extensii oferite face din AndEngine o alegere ideală pentru dezvoltarea unui joc 2D cu o complexitate nu foarte ridicată.

2.3 Box2D

Un motor de fizică este un framework ce ajută la simularea sistemelor fizice precum dinamica corpurilor rigide în cadrul unui joc video. Deoarece în jocuri se pune mai mult accent pe timpul în care simularea este realizată și nu pe precizia acesteia, majoritatea motoarelor de fizică existente realizează această simulare într-o manieră aproximativă.

Aflat printre cele mai cunoscute motoare de fizică pentru dezvoltarea jocurilor video, Box2D¹ este un proiect open source realizat de Erin Catto în C++ care își propune simularea dinamicii corpurilor rigide în contextul unei lumi virtuale 2D. Folosind concepte teoretice din fizica newtoniană, Box2D pune la dispoziția dezvoltatorilor următoarele funcționalități:

- Posibilitatea creării unei lumi virtuale supusă constant unor entități fizice precum gravitația, forța de frecare, densitate și elasticitatea obiectelor;
- Diverse tipuri de corpuri fizice: dinamice, statice sau hibride;
- Modelarea obiectelor sub formă de poligoane convexe, cercuri sau forme complexe ce pot fi obținute prin uniunea celor două;
- Detecția continuă a coliziunilor dintre corpuri sau grupări de corpuri;
- Metode de rezolvarea a coliziunilor;
- Simularea fidelă a forțelor de reacțiune și a impulsurilor;
- Diverse tipuri de articulații sau legături între obiecte fizice. [13]

1. <http://box2d.org/>

Implementarea elegantă și ușurința în modul de folosire al acestor funcționalități în procesul de dezvoltare a jocurilor video, au făcut din Box2D unul din cele mai populare motoare de fizică. Astfel, Box2D se poate „lăuda” că stă la baza multor jocuri existente pe piață, printre acestea numărându-se și Angry birds (cel mai descărcat joc din istoric platformelor mobile). Popularitatea a adus odată cu ea și portarea acestui motor pe diverse platforme de dezvoltare, dintre care și Android. Mai precis, a fost implementată o extensie Box2D pentru motorul de jocuri AndEngine.

Funcționalitățile expuse, popularitatea și modul de îmbinare cu motorul AndEngine, fac din Box2D candidatul ideal pentru implementarea simulării fizice din jocul „Worms”.

2.4 Physics Body Editor

Aplicația *Physics Body Editor*¹ sare în ajutorul motoarelor de fizică care suportă reprezentarea obiectelor doar sub formă de poligoane convexe. În multe situații jocurile conțin obiecte complexe ce prezintă zone concave greu de administrat din punct de vedere al fizicii coliziunilor. Dezvoltatorilor nu le rămâne decât să descompună manual obiectele dorite în poligoane convexe. Aici intervine însă Physics Body Editor care descompune forme oricât de complexe ale obiectelor furnizate programului prin intermediul unui bitmap. (Fig. 19)

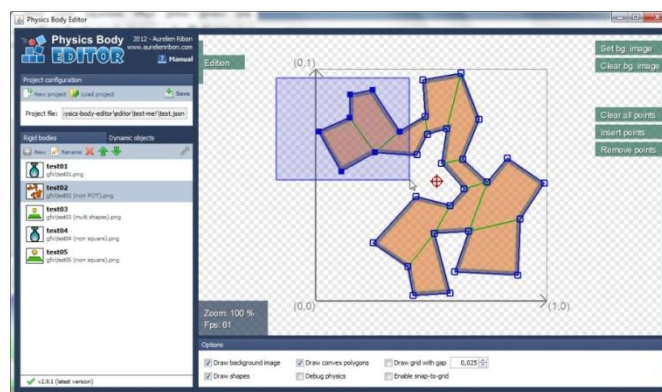


Fig. 19 Descompunerea unei forme complexe în multiple poligoane convexe

După încărcarea unei imagini cu forma dorită în program (similară cu cea din Fig. 19), Physics Body Editor oferă utilizatorului posibilitatea de a calcula în mod automat conturul obiectului sau îi oferă libertatea acestuia de a plasa singur nodurile dorite pe granița obiectului. Acestea sunt necesare pentru descompunerea formei în multiple poligoane convexe.

1. <http://www.aurelienribon.com/blog/projects/physics-body-editor/>

După terminarea pasului descris anterior, programul folosește nodurile determinate și formează multiple poligoane convexe care reunite compun obiectul inițial. Fiecare poligon este reprezentat prin coordonatele colțurilor sale. Aceste date sunt exportate mai departe în format **json** și sunt apoi folosite în joc pentru crearea formelor fizice complexe. Înainte de export, programul oferă utilizatorului posibilitatea testării formei obținute pentru eventuale modificări ale acesteia.

Physics Body Editor este folosit în procesul de dezvoltare a jocului „Worms” pentru a descompune forma hărții unui nivel în multiple poligoane convexe necesare simulării fizicii coliziunilor dintre obiecte.

2.5 Concluzii

Fie că este vorba de un joc dezvoltat de o întreagă echipă sau de un singur dezvoltator independent, uneltele software ușurează extrem de mult procesul de creare a jocurilor video.

Motoarele de joc ajută la administrarea resurselor, redarea elementelor vizuale și controlul logicii aplicației oferind per total un nivel de abstractizare în procesul de dezvoltare. Motoarele de fizică ajută la transpunerea unor aspecte din realitate în lumea virtuală a jocurilor. Sunt simulate astfel concepte precum dinamica obiectelor rigide, detecția coliziunii dintre corpuri sau existența unor forțe constante (gravitația, forța de frecare, elasticitate etc.). Editoarele de nivel ajută dezvoltatorii în conceperea structurii nivelurilor unui joc și exportarea rezultatelor într-o multitudine de formate în funcție de nevoile fiecăruia.

Pentru dezvoltarea jocului Worms am ales ca unelte de dezvoltare motorul de joc AndEngine, motorul de fizică Box2D și editorul nivele Physics Body Editor. Datorită funcționalităților disponibile, modului de organizare și extensiilor oferite, AndEngine a fost alegerea potrivită pentru implementarea unui joc cu un astfel de nivel de complexitate. Motorul Box2D împreună cu editorul de nivele Physics Body Editor, oferă exact performanța, scalabilitatea și funcționalitățile ce au fost necesare pentru simularea aspectelor realității în lumea jocului Worms.

Capitolul 3

Prezentarea aplicației

Această lucrare are ca obiectiv crearea jocului 2D cu numele „Worms” pentru platforma Android. Implementarea acestui obiectiv este realizată prin îmbinarea conceptelor și tehnologiilor prezentate în capitolele anterioare. Astfel, jocul are la bază motorul AndEngine prin intermediul căruia sunt controlate și expuse utilizatorului aspectele vizuale și audio (sprite-uri, animații, efecte sonore), sunt tratate input-urile dispozitivului și sunt administrate activitățile și scene componente ale aplicației.

Box2D crează prin simularea fizicii obiectelor (harta, personajele, armele, proiectilele etc.) o lume virtuală care oferă utilizatorului impresia realității. Motorul de fizică face astfel posibilă existența gravitației, forțelor de frecare, a obiectelor fizice cu proprietăți precum densitate sau elasticitate și a coliziunilor în timp real dintre acestea.

Jocul este construit în așa fel încât să fie intuitiv, ușor de jucat, dar totodată să pună la încercare abilitățile strategice ale utilizatorilor. Sesiunile de joc sunt scurte având o durată de timp de până la 10 minute și permit deținătorilor de dispozitive mobile să se bucure de aplicația în orice loc și moment.

Descriere

Worms face parte din categoria jocurilor strategice de tip turn-based, ceea ce înseamnă că acțiunea acestuia este împărțită în runde. Utilizatorii pot alege între două moduri de joc: singleplayer (împotriva unui NPC¹) sau hotseat multiplayer (împotriva unui alt jucător uman folosind același dispozitiv mobil).

Având la dispoziție 40 de secunde în fiecare rundă, jucătorul este nevoit ca folosind armamentul pus la îndemână să construiască o tactică prin care să elimine echipa adversă. În fiecare rundă doar unul dintre jucători este în controlul caracterelor sale. Jucătorul are la dispoziție o echipă formată din 4 caractere capabile să se miște în mediul înconjurător și să folosească o serie de arme cu ajutorul cărora să elimine componentii echipei adverse.

1. Non-Player Character sau Non-Person Character; caracter controlat prin intermediul unui algoritm de inteligență artificială

Un caracter este eliminat atunci când punctele lui de viață ajung la 0. Jocul se termină atunci când toate caracterele unei echipe au punctele de viață ajunse la 0, echipa adversă fiind declarată învingătoare.

Pentru a putea fi integrate în simularea fizică, personajelor din joc le sunt atașate corpuri fizice dinamice care au proprietăți precum masă, densitate sau elasticitate. Ele sunt afectate de gravitație și alte forțe sau impulsuri externe. Astfel, mișcările caracterelor sunt obținute prin aplicarea unor forțe externe în centrul de greutate al corpului. Pentru mișcarea la nivelul solului de-a lungul mediului înconjurător se aplică o forță care are componenta x semnificativ mai mare decât componenta y (astfel încât corpul să nu se ridice de la sol, dar să fie capabil să se miște de-a lungul pantelor) (Fig. 20), iar pentru sărituri este aplicată o forță care are componenta y semnificativ mai mare decât componenta x (permite corpului să se eleveze de la sol) (Fig. 21).

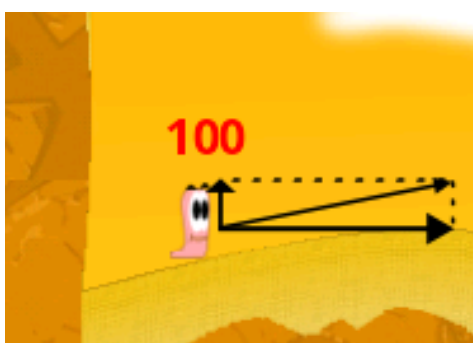


Fig. 20 Forța aplicată pentru deplasarea la nivelul solului

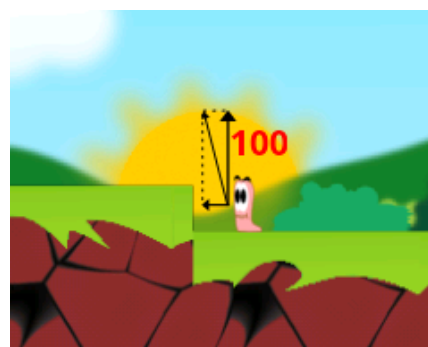


Fig. 21 Forța aplicată pentru sărituri

În modul singleplayer sunt create două tipuri de caractere: cele controlate de jucătorul uman și cele controlate de algoritmul de inteligență artificială. Cele două diferă prin proprietăți ce țin de logica aplicației, dar în momentul creării unei echipe nu se ține cont de tipul caracterului. Pentru a modela această situație am apelat la factory design pattern (Fig. 22). Interfața CharacterFactory are cunoștință despre clasa generală Character din care moștenesc cele două tipuri de caractere. Astfel, în momentul creării unui caracter nu se specifică tipul, această decizie rămânând pe seama claselor care implementează interfața.

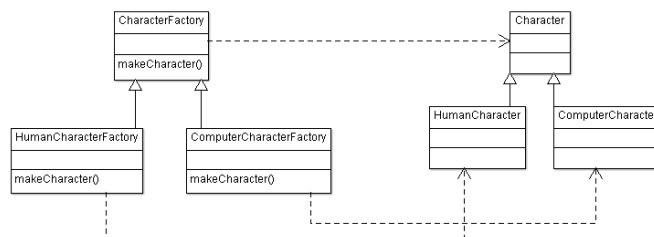


Fig. 22 Factory design pattern pentru crearea caracterelor

În ceea ce privește interacțiunea cu utilizatorul în timpul jocului, această este realizată într-un mod cât se poate de intuitiv și natural. Butoanele și informațiile sunt bine distribuite pe suprafața ecranului în încercarea de a nu îngreuna experiența de joc (Fig. 23). Astfel, zonele hașurate cu roșu servesc ca butoane pentru mișcarea la stânga sau la dreapta la nivelul solului a caracterului curent. Săriturile sunt activate printr-o simplă apăsare pe zona ce definește caracterul. Butonul din stânga jos (1) deschide meniul din care se pot alege armele, iar cel din dreapta jos (2) este pentru a trage cu arma curentă. Direcția de tragere este reglată cu ajutorul țintei (3) care se poate plimba împrejurul caracterului. În partea de dreapta sus a ecranului (4) utilizatorul este atenționat de timpul care a rămas din runda curentă. Porțiune centrală din josul ecranului (5) afișează progresul jocului (punctele de viață rămase fiecărei echipe).

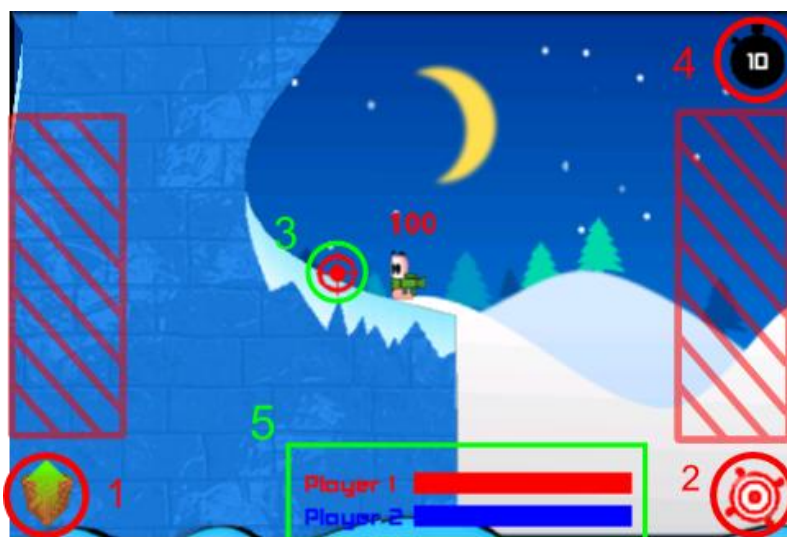


Fig. 22 Suprafețele din ecran destinate interacțiunii cu jucătorul

3.2 Mediul virtual

Mediul virtual al jocului Worms constă într-o suprafață cu forme complexe (Fig.23) pe care se desfășoară bătălia dintre cele două echipe. Ele sunt create astfel încât de-a lungul acestora să existe zone de câmp deschis, zone de retragere și componente la care nu se poate ajunge prin simplele mișcări pe care caracterele sunt capabile să le execute.

Hărțile sunt create în momentul lansării jocului din resurse bitmap alb-negru (pixelii negri reprezintă suprafața solidă, pixelii albi sunt aerul). Pentru a oferi o oarecare diversitate jocului, au fost create 3 teme vizuale: deșert, polul nord și mediu montan. La pornirea jocului este încărcată imaginea corespunzătoare nivelului selectat și este aleasă o temă vizuală în mod aleatoriu. Este aplicată terenului textura specifică temei alese și apoi este setată imaginea de background corespunzătoare.

Suprafața de joc are un rol important în oferirea unei experiențe de joc complete. Aceasta ia parte în simularea fizică fiind suprafața de suport pentru celelalte entități din joc. Ea însă nu este afectată de forța gravitațională sau orice altă forță sau impuls din exterior. Prin definiție, harta este un corp rigid static. Datorită formei complexe cu porțiuni concave, motorului de fizică Box2D îi este imposibil să reprezinte suprafața sub forma unui singur corp rigid poligonal. Forma trebuie astfel simplificată prin descompunerea în poligoane convexe. Acest lucru este realizat cu ajutorul unelei Physics Body Editor. (Fig. 23)

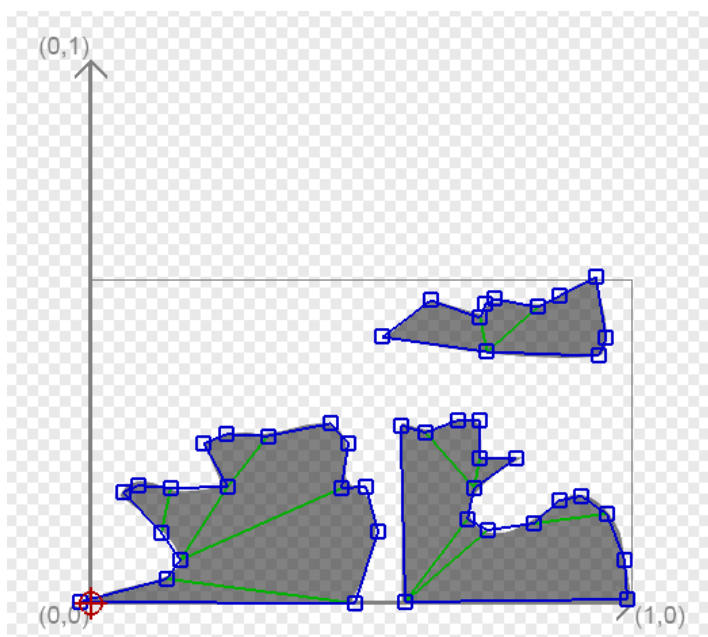


Fig. 23 Forma complexă a unui nivel descompusă în poligoane convexe

Obținerea corpurilor rigide complexe care să reprezinte suprafața de joc constă în parcurgerea următorilor pași:

- Se încarcă în programul Physics Body Editor bitmap-ul alb-negru care conține forma unui nivel din joc;
- Se determină conturul formei fie cu funcția autotrace din program, fie manual prin poziționarea nodurilor pe canvas;
- Se rulează algoritmul de descompunere poligonală, în urma căruia se obține un fișier în format **.json** (Tab. 1) care conține o listă de puncte ce reprezintă colțurile poligoanelor;
- Fișierul **.json** se adaugă ca resursă în pachetul jocului;
- În momentul pornirii unei instanțe a jocului, se identifică fișierul corespunzător nivelului selectat și cu ajutorul loaderului special implementat pentru a interpreta conținutul acelui fișier se crează o listă de obiecte reprezentând poligoanele dorite;
- Folosind punctele din lista poligoanelor convexe obținută anterior se crează un corp rigid static pentru fiecare dintre acestea; toate aceste corpuri reunite formează suprafața de joc.

```

    „polygons”: [
      [ { „x”: 0.669921875,
          „y”: 0.2509765625 },
        { „x”: 0.646484375,
          „y”: 0.271484375 },
        { „x”: 0.6376953125,
          „y”: 0.267578125 },
        { „x”: 0.599609375,
          „y”: 0.19921875 }
      ],
      [ { „x”: 0.599609375,
          „y”: 0.19921875 },
        { „x”: 0.6865234375,
          „y”: 0.2451171875 },
        { „x”: 0.669921875,
          „y”: 0.2509765625 }
      ],
      ...

```

Tab. 1 Fragment dintr-un fișier .json cu informațiile despre poligoanele convexe obținute

3.3 Armament

Probabil cea mai importantă trăsătură a jocului Worms sunt armele care sunt puse la dispoziție (Fig. 24). Ele sunt uneltele cu ajutorul cărora jocul poate fi câștigat. Fiecare dintre acestea au caracteristici diferite precum raza de acțiune sau puterea exploziei, făcându-le potrivite doar în anumite situații. (Tab. 2)



Fig. 24 Meniul cu armele disponibile în joc

În continuare vor fi prezentate în detaliu cele 6 arme aflate în dotarea fiecărui caracter din joc:

- **Grenade:** este o armă ce poate fi lansată de la distanță și descriere o traiectorie balistică în aer; ea este afectată de gravitație și de impactul cu suprafața de joc; produce o explozie la 3 secunde după lansare având capacitatea da a produce pagube de până la 50 de puncte de viață;
- **Rocket launcher:** este cea mai puternică armă din arsenal, dar totodată și cea mai greu de stăpânit; poate fi folosită de la distanță și proiectul lansat de aceasta descrie o traiectorie balistică în aer; proiectilul este afectat de gravitație și produce o explozie la contactul cu orice corp fizic având capacitatea da a produce pagube de până la 70 de puncte de viață;
- **Sniper:** este caracterizată de precizie; poate fi folosită de la distanță și proiectilul lansat de aceasta descrie o traiectorie liniară; provoacă pagube de până la 35 puncte de viață la impactul direct cu un alt caracter;
- **Baseball bat:** poate fi folosită doar în apropierea inamicului; dacă există contact direct cu un caracter provoacă o pagubă fixă de 30 de puncte de viață;

- **Dynamite**: poate fi folosită doar din apropiere; este afectată doar de gravitație nu și de alte forțe sau impulsuri exterioare; după 2 secunde de la lansare produce o explozie ce produce pagube de până la 50 de puncte de viață;
- **Teleport**: nu produce pagube altor caractere; folosită pentru a transporta instantaneu caracterul în orice loc de pe harta de joc.

Nume	Distanță	Rază de acțiune (pixeli)	Pagubă maximă	Timp de efect
Grenade	depărtare	64	50	3 secunde
Rocket launcher	depărtare	64	70	la impact
Sniper	depărtare	2	35	la impact
Baseball bat	apropiere	10	30	instant
Dynamite	apropiere	96	50	2 secunde
Teleport	depărtare	-	-	instant

Tab 2. Caracteristicile fiecărei arme

Armele sunt folosite o singură dată pe rundă de către un singur caracter. Ar fi ineficient dacă la fiecare rundă sau de fiecare dată când jucătorul schimbă armele s-ar crea o nouă instanță a clasei respective. Soluția este oferită de object pool pattern, un design pattern creațional care se folosește de un set de obiecte inițializate și le păstrează mereu gata pentru folosire. Astfel, a fost creată clasa WeaponsPool care la startul jocului inițializează câte un obiect din clasa specifică fiecărei arme și în momentul în care primește o cerere oferă o referință la unul din obiectele inițializate. După folosire, fiecare obiect este resetat la o stare inițială. (Fig. 25)

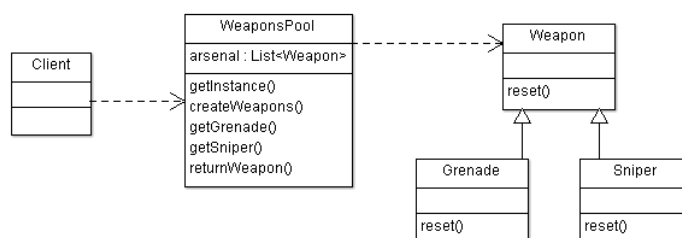


Fig. 25 Object pool pentru clasele de arme

3.4 Concluzii

Folosind instrumentele prezentate în Capitolul 2, jocul Worms a fost implementat astfel încât să ofere utilizatorului final o experiență de joc plăcută, intuitivă și cât mai naturală.

La nivelul jocului s-a încercat obținerea diversității prin crearea de numeroase nivele unice prin caracteristicile oferite, prin temele de texturare disponibile și prin instrumentele ce pot fi folosite de utilizator în timpul jocului gândite astfel încât să acopere toate situațiile ce pot apărea.

La un nivel transparent utilizatorului s-au folosit concepte fizice și de inginerie a programării pentru a obține o aplicație robustă, scalabilă și eficientă din punct de vedere al algoritmilor utilizați.

Capitolul 4

Inteligența artificială

În jocurile video, inteligența artificială își propune crearea de comportamente aparent inteligente a entităților NPC pentru a oferi utilizatorului final o provocare și pentru a îmbogăți experiența de joc. Se încearcă simularea unui comportament uman și evitarea folosirii avantajelor clare pe care o entitate NPC le are față de un jucător uman. Printre cei mai cunoscute concepte din domeniul inteligenței artificiale se numără arborii de decizie, rețelele neuronale sau algoritmi genetici.

În jocurile de acțiune în timp real, abilitățile potențiale ale unui NPC sunt net superioare oricărui jucător uman: acuratețe perfectă, reflexe fulgerătoare, etc. Provocarea este în acest caz crearea unui AI care să prezinte un comportament uman, nu perfecționist, oferind astfel utilizatorului posibilitatea de a câștiga jocul în fața acestuia.

În jocurile de strategie de tip turn-based lucruri se schimbă, acuratețea și reflexele nemaifiind cele mai importante abilități necesare victoriei. Intervin lucruri precum înscusimea și intuiția umană, factori care pot fi decisivi întrecerea cu o entitate NPC. În acest caz se dorește proiectarea unui AI care să prezinte un comportament inteligent în alegerea strategiilor de apărare și atac, să ofere provocare jucătorului, dar în final să existe posibilitatea înfrângerii acestuia. [15]

4.1 Obiective

Este evidentă necesitatea implementării unui algoritm de inteligență artificială pentru jocul Worms. Entitatea NPC controlată de acest algoritm trebuie să fie capabilă să se miște în mediul înconjurător, să își identifice adversarii și să fie cunoască modul de utilizare a tuturor armelor din gama pusă la dispoziție. Pentru a păstra un nivel de corectitudine față de jucătorul uman, AI-ul trebuie să își construiască strategiile pe baza acelorași informații pe care le deține și utilizatorul, evitând astfel să folosească date sau calcule din interiorul sistemului jocului.

Algoritmul A*

Căutarea drumului dintre două puncte aparținând unei configurații cunoscute este printre cele mai comune probleme pe care algoritmi de inteligență artificială își propun să le rezolve. Printre acești algoritmi se numără și A*, cunoscut pentru performanța și acuratețea oferită. Extins din algoritmul lui Dijkstra¹, A* este folosit pentru a căuta drumul optim dintre două puncte ținând cont de obstacolele care pot apărea în cale².

Ideea generală a algoritmului este ca plecând de la un nod de start și folosind o tehnică de căutare de tip best-first în lista nodurilor adiacente să se găsească drumul de cost minim până la nodul final. Alegerea de tip best-first este făcută ținând cont de costul fiecărui nod, numit și costul F. Acesta se calculează ca fiind suma dintre costul necesar deplasării din punctul de start până la nodul curent (costul G) și costul estimativ al deplasării de la nodul curent la nodul final (costul H). Costul G se determină prin atribuirea unui cost deplasărilor cu un nod pe orizontală, verticală sau diagonală; se presupune că mișcările se pot face doar în nodurile adiacente. Pentru costul H este folosită o euristică. De-a lungul căutării algoritmul reține o listă de noduri vizitate în cazul în care calea urmată la un moment dat nu oferă nicio soluție finală. La fiecare pas este ales nodul având costul cel mai mic, este adăugat în lista de vizitați și sunt verificate nodurile adiacente. Fiecărui nod din drumul parcurs îi este atribuit un nod părinte astfel încât în momentul în care a fost găsit nodul final să se poată reface drumul către poziția inițială (Fig. 26) [16]

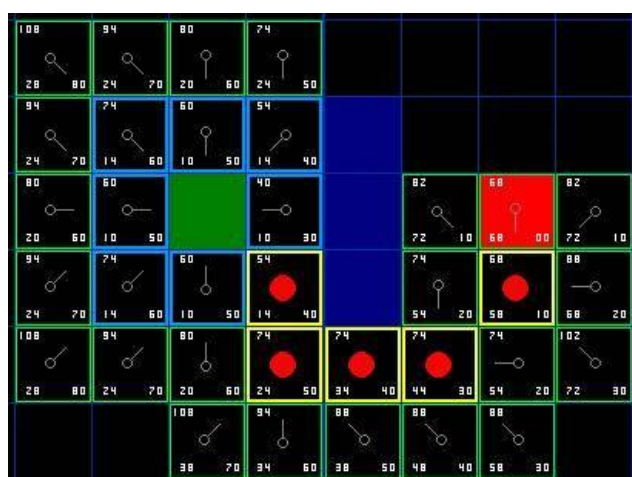


Fig. 26 Nodurile parcurse de A* pentru găsirea drumului optimal³

1. <http://www.cs.auckland.ac.nz/software/AlgAnim/dijkstra.html>
2. <http://www.policyalmanac.org/games/aStarTutorial.htm>
3. <http://www.policyalmanac.org/games/aStarT3.jpg>

În Worms, algoritmul A* este folosit pentru a verifica existența și eventual a determina drumul dintre o entitate NPC și ținta aleasă de acesta. Algoritmul este implementat cu puține modificări pentru a respecta configurația lumii create în joc.

Primul pas făcut în determinarea unui drum este simplificarea ariei de căutare. Aceasta se realizează prin împărțirea imaginii a terenului în dreptunghiuri de mărimea unui caracter (se obține o grilă de noduri). Utilizând bitmap-ul cu forma terenului se determină care dintre aceste noduri ale grilei obținute reprezintă eventuale obstacole și care sunt candidate pentru un posibil drum optimal. Se determină apoi nodul de start în care se află caracterul NPC și nodul final în care se află inamicul. Având aceste informații se rulează implementarea algoritmului A* pentru determinarea drumului între cele două noduri. Modificările aduse algoritmului sunt următoarele:

- La verificarea vecinilor nodului curent dintr-un pas al algoritmului se sare peste nodurile nord și sud (dacă privim nodurile adiacente ca pe un sistem de coordonate carteziane) întrucât caracterele nu pot efectua astfel de deplasări în contextul lumii create
- Deoarece caracterele sunt capabile să sară peste obstacole, la verificarea vecinilor nu sunt trecute în revistă doar nodurile aflate în imediata apropiere, ci și cele aflate la un nod distanță; de exemplu, dacă nodul din est este detectat ca obstacol sunt imediat verificate nodurile adiacente cu acesta pentru a se confirma dacă se poate realiza o deplasare prin săritură peste acest nod. Această modificare permite atât ocolirea obstacolelor, cât și îmbunătățirea performanței algoritmului în contextul dat. (Fig. 27)

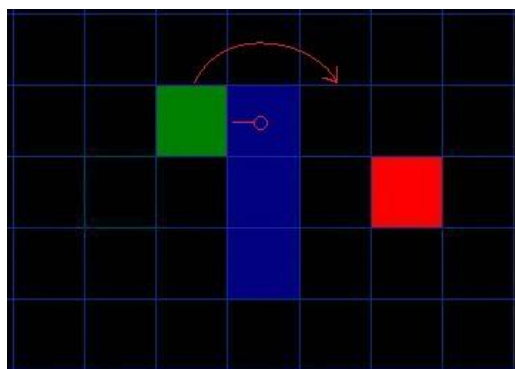


Fig. 27 Algoritmul găsește o posibilă săritură peste un obstacol

```

public LinkedList<Node> AStar(int startRow, int startCol, int endRow, int endCol) {
    closedSet = new LinkedList<Node>();
    openSet = new LinkedList<Node>();
    path = new LinkedList<Node>();
    parentNode = new HashMap<Node, Node>();

    openSet.add(map[startRow][startCol]);
    map[startRow][startCol].setG(0);
    map[startRow][startCol].setH((Math.abs(endRow - startRow) +
                                   Math.abs(endCol - startCol)) * 10);

    while(!openSet.isEmpty()) {
        int index = getLowestFCost(openSet);
        Node current = openSet.get(index);

        if (current.getRow() == endRow && current.getCol() == endCol) {
            backtrackPath(startRow, startCol, endRow, endCol);
            return path;
        }

        openSet.remove(index);
        closedSet.add(current);

        boolean ne = checkNeighbour(current.getRow(), current.getCol(), -1, +1,
                                     endRow, endCol, 7);
        boolean e = checkNeighbour(current.getRow(), current.getCol(), 0, +1,
                                    endRow, endCol, 10);
        boolean se = checkNeighbour(current.getRow(), current.getCol(), +1, +1,
                                    endRow, endCol, 7);
        boolean sw = checkNeighbour(current.getRow(), current.getCol(), +1, -1,
                                    endRow, endCol, 7);
        boolean w = checkNeighbour(current.getRow(), current.getCol(), 0, -1,
                                    endRow, endCol, 10);
        boolean nw = checkNeighbour(current.getRow(), current.getCol(), -1, -1,
                                    endRow, endCol, 7);
        if (!e && ne)
            checkNeighbour(current.getRow(), current.getCol(), 0, +2,
                           endRow, endCol, 5);
        else if (!w && nw)
            checkNeighbour(current.getRow(), current.getCol(), 0, -2,
                           endRow, endCol, 7);
    }

    return null;
}

```

Funcția `AStar(int, int, int, int)` determină drumul optimal dintre două puncte de pe hartă ([startRow, startCol] și [endRow, endCol]). Întâi este determinat costul nodului de start și acesta este adăugat într-o listă de prelucrare `openList`. Cât timp această listă nu este vidă sau nu s-a găsit soluția, se alege din `openList` nodul cu costul cel mai mic (`getLowestFCost`). Acesta se introduce în lista nodurilor vizitate `closedList` și apoi îi sunt verificați vecinii care candidează pentru intrarea în `openList`. Dacă unul din vecinii est sau vest sunt detectați ca obstacole se verifică posibilitatea unui salt.

4.3 Strategii

Algoritmul de inteligență artificială pentru jocul Worms este structurat cu ajutorul design pattern-ului strategy. Strategy permite implementarea unei familii de algoritmi care moștenesc aceeași interfață și interschimbarea lor în mod dinamic la runtime. Se poate descrie astfel comportamentul unei entități NPC care poate lua decizii diferite de la o rundă a jocului la alta.

Sunt identificate două strategii în implementarea algoritmului de inteligență artificială după care caracterele NPC să poată lua decizii: strategia alegerii inamicului ales ca țintă de atac și strategia alegerii armei cu care va fi realizat atacul.

Strategia de alegere a inamicului ce urmează să fie atacat conține cinci algoritmi: inamicul cel mai apropiat ca distanță, inamicul cel mai depărtat ca distanță, inamicul cu cele mai multe puncte de viață, inamicul cu cele mai puține puncte de viață și alegerea unui caracter în mod aleatoriu. La crearea fiecărui caracter NPC acestuia îi este atașată una dintre aceste strategii, ea rămânând neschimbată pe tot parcursul jocului. (Fig. 28)

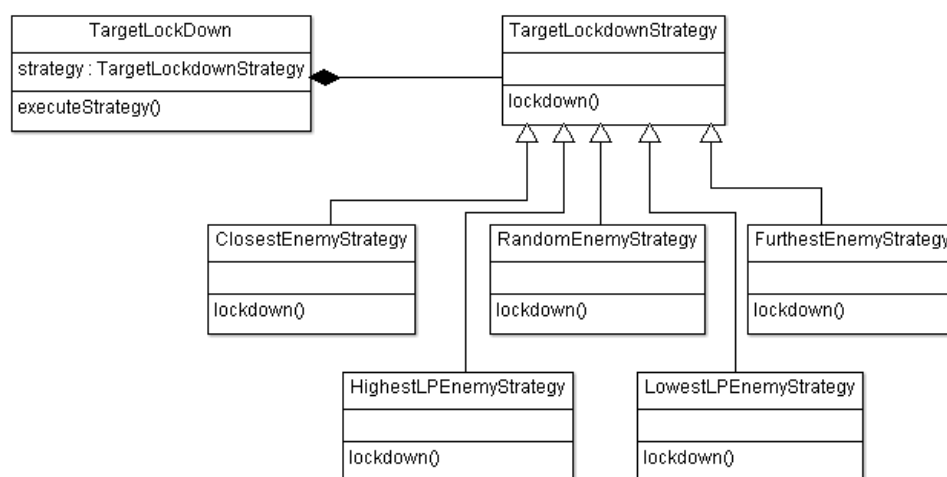


Fig. 28 Strategia de alegere a următorului atac

A doua strategie, cea de alegere a armei de atac, este mai complexă decât prima ea necesitând mai multe calcule și timp de procesare. Sunt determinați trei algoritmi aparținând acestei strategii între care entitățile NPC pot interschimba de la o rundă la alta. Alegerea strategiei se face astfel (Fig. 29):

- **Strategia atacului direct:** primul pas este aplicarea algoritmului A* pentru găsirea unui drum de la NPC la inamicul ales; dacă acesta nu există se trece la testarea pentru următoarea strategie. Dacă acesta există AI-ul parcurge drumul găsit de A*; ajuns la capătul drumului el scanează împrejurimile inamicului căutând coechipieri. În cazul în care au fost identificați coechipieri în zona vizată, AI alege să folosească arma **Baseball bat** pentru a nu cauza pagube celorlalți din echipa sa; în caz contrar el folosește arma **Dynamite** care produce mai multe pagube decât **Baseball bat**. Pentru a nu fi afectat de explozie AI parcurge drumul găsit de A* în ordine inversă pentru a se îndepărta de țintă;
- **Strategia atacului din poziția optimală:** dacă algoritmul A* nu a întors nicio soluție se parcurg nodurile vizitate de acestea (aflate în closedList) și se verifică dacă vreunul dintre acestea are vedere directă către țintă. În caz negativ, se trece la următoarea strategie; în caz pozitiv se reface drumul către acel nod, AI parcurge drumul către acel nod și ajuns la capăt folosește aceeași scanare pentru verificarea împrejurimilor de coechipieri. În cazul în care au fost identificați coechipieri în zona vizată, AI alege să folosească arma **Sniper** pentru a nu cauza pagube celorlalți din echipa sa; în caz contrar el folosește arma **Rocket Launcher** care produce mai multe pagube decât **Sniper**;
- **Strategia atacului indirect:** dacă nu a fost găsit nici drumul către inamic, nici o poziție optimală din care să existe vedere liberă către țintă, AI alege să folosească arma **Rocket Launcher**; acesta își fixează ținta, caută cel mai ascuțit unghi de tragere care nu intersectează niciun obstacol și lansează atacul către inamic. Dacă mai mult în mai mult de 2 runde entitatea NPC nu a reușit să cauzeze pagube datorită obstacolelor, el va folosi arma **Teleport** pentru a se deplasa instant într-o poziție favorabilă.

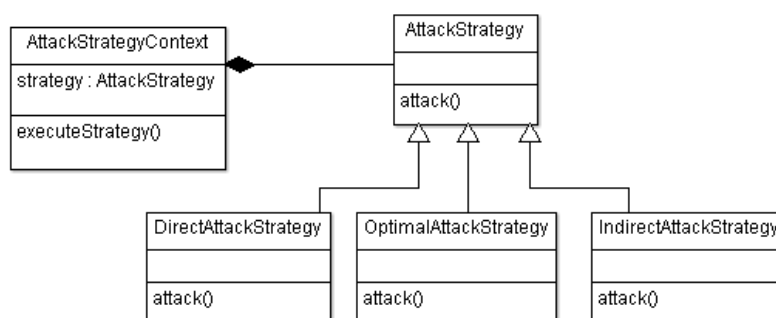


Fig. 29 Strategia alegerii tipului de atac executat

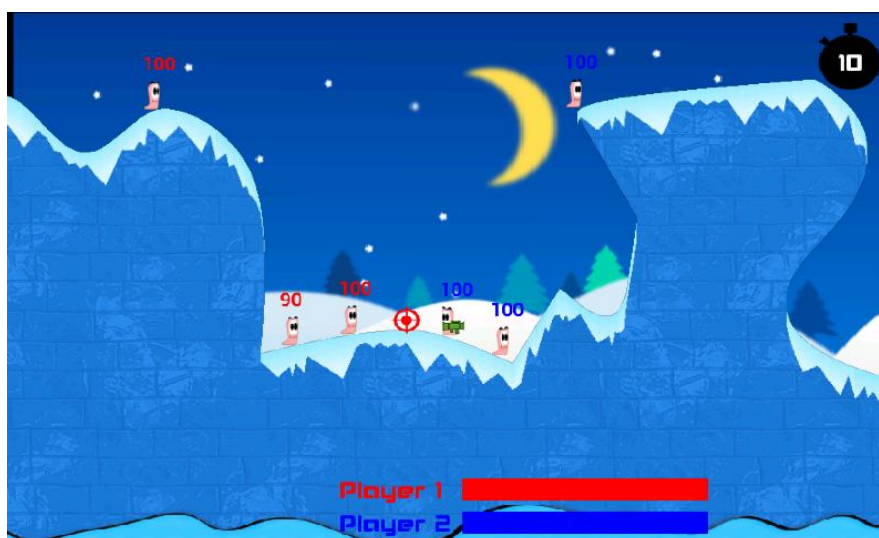


Fig. 30 Captură din timpul jocului; AI execută un atac direct asupra inamicului

4.4 Balansarea dinamică a dificultății

Algoritmul de inteligență artificială trebuie să fie capabil să identifice momentele în care oponentul nu are rezultate bune în privința jocului sau din contra, demonstrează abilități de nivel superiorca jucător. În funcție de aceste date AI trebuie să își schimbe comportamentul, scenariile urmate sau parametri ai jocului în mod dinamic pentru a păstra utilizatorul interesat de la începutul până la sfârșitul jocului.

În Worms aceasta tehnică a fost implementată cu ajutorul unui scor care se păstrează pentru echipa utilizatorului uman. Acest scor măsoară raportul dintre punctele de viață ale celor două echipe, diversitatea armelor folosite, numărul de runde consecutive în care acesta nu a produs pagube sau a cauzat pagube propriiei echipe.

Folosind toate aceste date AI își reglează acuratețea cu care lansează atacurile, face alegeri nu tocmai potrivite în ceea ce privește strategia abordată, mărește durata unei runde pentru a acorda timp utilizatorului să își construiască strategia și în cazuri extreme sare peste runde.

4.5 Concluzii

Cu siguranță inteligența artificială joacă un rol important în construirea unei experiențe de joc complete. Provocarea este implementarea unui algoritm să pună jucătorul în dificultate păstrând un nivel de corectitudine față de acesta. În final jucătorul trebuie să poată fi capabil să iasă învingător.

Algoritmul de inteligență artificială din jocul Worms își bazează structura pe șablonul creațional strategy. Strategiile sunt create în jurul unor algoritmi de determinare a drumului optim și a unor funcții de scanare a hărții. Aceste strategii sunt atașate entităților NPC care sunt astfel capabile să ia decizii în timpul jocului și să ofere o provocare utilizatorului uman. Algoritmul de inteligență artificială reușește să adapteze dificultatea jocului folosind tehnici balansare dinamică oferind în final șansa jucătorului de a ieși învingător.

Concluzii finale

Concluzii

Lucrarea „Worms – Joc 2D pentru platforma Android” surprinde tendințele existente pe piața platformelor de dezvoltare mobilă și își propune dezvoltarea unui joc care să întrunească toate caracteristicile unei aplicații reușite, accentul căzând pe implementarea unui algoritm de inteligență artificială eficient și care să ofere utilizatorului o provocare și o experiență de joc dinamică.

Contribuții personale

Am reușit prin îmbinarea elementelor vizuale și audio cu implementarea simulării fizice și a diversității de aspecte ale jocului (arme, hărți, teme) să creez o aplicație ce oferă o experiență generală plăcută, intuitivă și naturală.

Experiența este completată de implementarea algoritmului de inteligență artificială care oferă impresia generală de comportament inteligent al entității NPC. Pentru obținerea acestui rezultat am adaptat algoritmul de pathfinding A* la contextul jocului și am implementat un număr de funcții de scanare a hărții ce ajută la stabilirea celei mai potrivite strategii de joc pentru AI. Am reușit prin balansarea dinamică a dificultății să adaptez jocul pentru nivelul de îndemânare a fiecărui jucător, oferind astfel în final posibilitatea obținerii victoriei.

Direcții de viitor

Pentru viitor există mai multe direcții în care aplicația poate fi dezvoltată:

- Trecerea la grafică vectorială care să permită scalarea ecranului de joc pe dispozitive de orice rezoluție fără pierderea calității;
- Introducerea de noi arme și ustensile în arsenalul jocului;
- După cum a fost prezentat pe parcursul lucrării, motorul de fizică lucrează doar cu poligoane convexe ceea ce face din implementarea unui teren destructibil o sarcină dificilă; această funcționalitate poate fi atinsă prin implementarea unui algoritm de

decupare a poligoanelor (sunt decupate poligoanele ce formează terenul la un moment dat, se creează unele noi și se înlocuiește structura hărții);

- Implementarea modului multiplayer online; aici provocarea va veni atât pe partea de comunicare în rețea datorită capacităților reduse a dispozitivelor mobile, cât și pe partea aplicației în ceea ce privește sincronizarea simulării fizice între jucători.

Bibliografie

- [1] ***, *Ten Reasons why SmartPhones are killing the Gaming consoles*, <http://www.gadgetcage.com/ten-reasons-why-smartphones-are-killing-the-gaming-consoles/23268/>, 15.11.2012
- [2] ***, *Android, the world's most popular mobile platform*, <http://developer.android.com/about/>, 15.11.2012
- [3] ***, *Angry Birds (video game)*, [http://en.wikipedia.org/wiki/Angry_Birds_\(video_game\)](http://en.wikipedia.org/wiki/Angry_Birds_(video_game)), 16.11.2012
- [4] ***, *Drag Racing*, <https://play.google.com/store/apps/details?id=com.creativemobile.DragRacing>, 16.11.2012
- [5] ***, *Fruit Ninja*, http://en.wikipedia.org/wiki/Fruit_Ninja, 16.11.2012
- [6] ***, *Subway surfers*, http://en.wikipedia.org/wiki/Subway_Surfers, 16.11.2012
- [7] ***, *Where's my water*, <https://play.google.com/store/apps/details?id=com.disney.WMWLite>, 16.11.2012
- [8] ***, *Worms(series)*, [http://en.wikipedia.org/wiki/Worms_\(series\)](http://en.wikipedia.org/wiki/Worms_(series)), 20.11.2012
- [9] ***, *Worms 2: Armageddon*, <http://www.team17.com/games/worms/worms-2-armageddon>, 16.06.2013
- [10] ***, *Android operating system*, [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)), 05.03.2013
- [11] ***, *What is Android?*, <http://www.engineersgarage.com/articles/what-is-android-introduction>, 05.03.2013
- [12] Rick Rogers, *Learning Android game programming*, 01.12.2011
- [13] Erin Catto, *About Box2D*, <http://box2d.org/about>, 10.03.2013
- [14] Aurelian Ribon, *Physics Body Editor*, <http://www.aurelienribon.com/blog/projects/physics-body-editor/>, 11.03.2013
- [15] Ed Welch, *Designing AI algorithms for turn based strategy games*, http://www.gamasutra.com/view/feature/129959/designing_ai_algorithms_for_.php, 15.06.2013
- [16] Patrick Lester, *A* Pathfinding for Beginners*, <http://www.policyalmanac.org/games/aStarTutorial.htm>, 16.06.2013