LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

"Jifo spēles programmēšana"

KVALIFIKĀCIJAS DARBS

Autors: Andrejs Jauja
Studenta apliecības Nr.: AJ13029
Darba vadītājs: Dr. dat. Jānis Zuters

2015

ANOTĀCIJA

Šajā kvalifikācijas darbā ir aprakstīta 2D puzles veida spēle radīta iericem ar dažādām operētājsistēmam(Windows,IOS,Android).Spēle piedāvā vairākus līmeņus ar vairākiem pretiniekiem un dažādiem speles elementiem piedavajot mainīgu sarežģītības pakāpes lidz ko spēle progresē. Tā kā spēles grafiskio daļu izmeri ir diezgan lieli speles izmers lidz ar to ir liels un spēle parstaj darboties uz vecākām iericēm kurmam nav pietiekami RAM atmiņas.Sakumā spēle bija paredzēta tikai Ipad, bet spēle ir izveidota, lai ta darbotos uz iepriekš minētajam operētājsistēmam. Spēle izveidota C# programmēšanas valodā, Unity vidē.(izmantojot tā dzinuli)

Atslēgas vārdi : C# , Unity, Android, IOS

ABSTRACT

This document describes a 2D puzzle type of game designed to work on several platforms(Windows,IOS,Android). Game features multiple levels with multiple enemies and different gameplay elements for varying difficulty levels as game progresses. As the game has rather large graphical objects, size of the game is rather large and will crash on older devices that doesn't have enough RAM memory.But overall game was supposed to be developed in mind to work on newest Ipad only, yet game has been built to run on previously mentioned platforms as well.

Game has been developed in C# programming language in Unity envieroment(using its engine).

Keywords : C# , Unity, Android, IOS

Saturs

IEVADS

Android operētājsistēma komerciāli pieejama veidtālruņos kļuva 2008. gadā, un 6 gadu laikā ir kļuvusi ļoti plaši izplatīta ne tikai veidtālruņu operētājsistēma, bet arī atradusi pielietojumu vairākās citās tehnoloģijās, kā piemēram televizoros. Pēdējo gadu laikā tirgū arī sāk strauji parādīties ar Android operētājsistēmu aprīkoti mini datori, kurus var pieslēgt jebkurai modernai ierīcei aprīkotai ar HDMI portu.

IOS operētajsistema darbina neskaitamas Apple ierices sakot no Ipod touch,Iphone un beidzot ar Ipad sakot jau no 2007 gada.Lai gan šī operētājsistēma darbijna tikai Apple produktus to skaits tomēr ir iespaidīgs, gandrīz tik pat liels kā Android ierīču skaits.

Visbeidzot Windows operētajsistēma darbina gandrīz visus galda datorus un šī ir vis izplatītākā operētajsistema pasaulē un šī paša iemesla dēļ šī kvalifikācijas darba ietvaros spēle tika veidota uz Windows OS, un tā kā spēlei obligāti bija jaiet uz IOS, tad Android OS ari tika iekļauta sarkstā, lai produktu varētu izmantot uz teju jebkuras ierīces.

Spēle dotajā momentā piedāvā 3 līmeņus ar vairakiem pretiniekiem un ar katru līmeni sarežģītības pakape paaugstinās palielinot spēles elementu daudzumu ka arī pieveinojot dažādus šķēršļu elementus.

Kvalifikācijas darba ietvaros tika izstrādāta dokumentācija. Dokumentācija sastāv no:

- programmatūras prasību specifikācijas;
- testēšanas dokumentācijas;
- konfigurācijas pārvaldības;
- kvalitātes nodrošināšanas;
- darbietilpības novērtējuma;
- programmatūras pirmkoda fragmentiem.

DEFINĪCIJAS

OS - operētājsistēma

C# - programmēšanas valoda

PPS - Programmatūras prasību specifikācija

2D - 2 dimensijas (x un y asis)

RAM - random access memory - Brīvpiekļuves atmiņa

MB - Mega byte - datu daudzuma mērvienība

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Nolūks

Programmatūras prasību specifikācija (turpmāk tekstā - PPS) ir izstrādāta Latvijas Universitātes 2. kursa kvalifikācijas darba ietvaros un ir paredzēta programmprodukta “Jifo” prasību aprakstīšanai.

Dokumenta nolūks ir precīzi un viennozīmīgi formulēt sistēmas prasības un raksturot tās funkcionalitāti. PPS ir paredzēta izstrādātājiem, kuri veiks programmprodukta projektējuma un paša programmprodukta izstrādi, un pasūtītāja pārstāvjiem, kuri apstiprina PPS.

1.1.2. Darbības sfēra

Programmprodukts "Jifo" ir spele primāri paredzēta reklamēt infogr.web lapu. Dotajā momentā pieejami ir 3 līmeņi ar vairaka veida pretiniekiem un spēles darbības veidiem(gameplay types) un to elementiem, paredzēts ir pievienot gane vel līmeņus,gan elementus uc. detaļas(punkti utt.). Spēli palaist un izmantot var bez interneta pieslēguma.

Programma ir puzles tipa 2D spēle kurā spēlētāja uzdevums ir uzveikt dažādus pretiniekus dažādos līmeņo.Lai uzveiktu pretinieku tam uzbrūkot ir jasaliek tāds kā diagrammas tipa objekts izmantojot dažādu krāsu mazakus aplīšus noteiktā laikā.

Pati programma var darboties ar Windows,IOS un Android OS, lai gan paredzets bija darboties tikai uz jaunākā Ipad.

1.1.4. Saistība ar citiem dokumentiem

Dokumenta noformēšanā ievērotas standarta LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis” prasības.

1.1.5. Pārskats

Dokuments sastāv no 4 daļām:

1. **Ievads.** Ievadinformācija, kas satur dokumenta nolūku, identificē programmproduktu, sniedz visu nepieciešamo terminu skaidrojumus, kas sastopami šajā dokumentā, un norāda saistību ar citiem dokumentiem.

2. **Vispārējs apraksts.** Aprakstīta produkta perspektīva un galvenās tā funkcijas, galvenās lietotāju grupas, to raksturiezīmes, kā arī galvenie produkta ierobežojumi.

3. **Funkcionālās prasības.** Detalizēti aprakstīti funkciju mērķi, ievaddati, apstrāde,
izvaddati un kļūdu paziņojumi.

4. **Nefunkcionālās prasības.** Apraksta prasības pret sistēmu kopumā (pieejamība, datu drošība utt.), nevis pret katru sistēmas sastāvdaļu.

1.2. Vispārīgs sistēmas apraksts

1.2.1. Produkta perspektīva

Programma ir patstāvīga un nav nepieciešama interneta sasaiste un nepieciešama Android 2.3 ,Windows 2000 vai IOS 8.3 OS versijas un minimums 100mb RAM. Dotā lietotne – programmatūra ir paredzēta lietotājiem izklaidei, īpaši cilvēkiem kas ir aizrauti ar spēlēm un digitālo izklaides industriju. Produkts ir veidots ar tālākas attīstības, uzlabojumu iespējām. Ja produkts gūs noteiktu sociālu atbalstu tad visticamāk tas tiks attīstīts tālāk.

1.2.3. Produkta funkcijas

Produkta funkcijas ir iedalītas pēc loģiski sazarotiem moduļiem, sadaļām. Lietotnei ir jāsatur:

- Atbalstam uz dažādu izšķirtspēju veidtālruņiem
- Jadarbojas obligati uz jaunākā Ipad ar 30 kadriem sekundē

Speles sākuma aina

- Start - poga, kas ved uz līmeņu izvēlles ainu
- Info - (i) poga, kas ved uz spēles izstrādātāju ainu
- Quit - (x) poga lai izietu no spēles
- Options - (zobrata ikona) poga kas ved uz iespēju ainu

Iestatijumu aina

- Music On/Off - Ieslēdz/izslēdz mūziku visās ainās
- Sound On/Off - Ieslēdz/izslēdz visas skaņas spēlē
- Reset game data - Poga kas izdzēš esošo spēles progresu un atjauno noklusētās vērtības
- Home - (mājas ikona) Ved atpakaļ uz iepriekšējo ainu

Starpskata filmiņas aina

- Nospiežot jebkur uz ekrāna,bilde nomainas uz nākamo, līdz nonak līdz pēdējai, kad nospiežot uz ekrāna tiek iekrauta nākama aina. (izmantojot iekrauanas ainu

Iekraušanas aina

- Iekrauj nākamo ainu, bet ja ir pieejama filmiņa, kas iepriekš nav nospēlēta, tiek palaista Starpskata filmiņas aina, pirms iekrauj nākamo ainu

Līmeņu izvēles aina

- Home - (mājas ikona) Ved atpakaļ uz Spēles sākuma ainu
- Options - (zobrata ikona) poga kas ved uz iespēju ainu
- Bultiņas sanos ļauj nomainīt līmeni izēlei
- Novelkot ar kursoru vai pirkstu no labas uz kreiso vai pretējā virzienā arī tiek nomainīts līmenis ko izvēlas.
- Play -(bultiņa zem līmeņa nosaukuma) aina tiek nomainīta uz iekraušanas ainu, kas tālāk, ja nepieciešams plaiž filmiņu un pēc tam iekrauj nepieciešamo līmeņa ainu

Izstrādātāju aina

- Nospiežot jebkur uz ekrāna nomainās bilde līdz tās beidzas, kad atdriežās iepriekšējā ainā.

Spēles līmeņa aina

- Home - (mājas ikona) ved atpakaļ uz līmeņa izvēles ainu
- Tēla kustība pa speles laukumu
- Uzbrukšana pretiniekam
- Spēles elementu radīšana un to vizualizācija
- Fona objektu(mākoņu un klinšu) kustināšana

1.2.3 Lietotāja raksturiezīmes

Šāda tipa lietotne būs interesanta cilvēkiem kas brīvajā laikā spēlē spēles. Spele dotajā momentā ir tikai angļu valodā līdz ar to lietotājam jāsaprot angļu valoda.

1.2.4 Vispārējie ierobežojumi

iPhone viedtālrunim jābūt vismaz sākot no 8.3 Versijas.
Android viedtālruņa OS jābūt vismaz sākot no versijas 2.3
Windows lietotājiem jabūt vismaz Windows 2000.
Kā arī ierīcei jābūt vismaz 100MB RAM atmiņai.
Dotajā momentā aplikācija būs pieejama tikai AppStore Apple lietotājiem, bet iespēja ir palaist programmu uz iepriekš minētajam OS.(pagaidām nav pieejama aplikacija, jo notiek planošana tālākajam darbam)

1.3 Konkrētas prasības

Produktam jābūt izstrādātam izmantojot Unity3D spēļu dzinēja vidi.

1.3.1 Funkcionālās prasības

1.3.1.1 Spēles sāuma aina

Šī aina paredzēta kā titul skats, pirms jebkādām spēles darbībam.

1. Poga - Start
2. Poga - Options
3. Poga - Credits
4. Poga - Quit

Apstrāde un Izvade:

1. Tiek ieladēta nākamā, Līmeņu izvēles aina, bet vispirmstiek parādīta maza "filmiņa".
2. Tiek ielādēta iespēju aina, pec kuras nospiežot uz mājas ikonas atgriežas sākuma ainā
3. Tiek ilādēta Izstrādātāju aina pēc kuras atgriežas sākuma ainā
4. Programma tiek izslēgta

1.3.1.2 Izstrādātāju aina

Šī aina pardzēta parādītsponsorus,firmas utt, kas iesaitīti šajā projektā.

1. Attēli maiņa

Apstrāde un Izvade:

1. Nospiežot jebkur uz ekrāna attēli mainās līdz vairāk nav attēli ko mainīt un šajā gadijumā atgriežās uz iepriekšējo ainu

1.3.1.3 Iespēju aina

Šī aina paredzēta dažadu iestatījumu maiņai

1. Poga - home
2. Poga On/off - Music
3. Poga On/off - Sound
4. Poga - Reset game data

Apstrāde un Izvade:

1. Ved atpakaļ uz iepriekšējo ainu
2. Nomaina mainīgā vērtību globālajos mainīgajos, lai mūzika netiktu spēlēta turpākajās ainās
3. Nomaina mainīgā vērtību globālajos mainīgajos, lai skaņa,muzika utt. netiktu spēlēta turpmākajās ainās
4. Atjauno noklusētos spēles datus(Līmeņus,pretiniekus uc.)

1.3.1.4 Līmeŋu izvēles aina

Šī aina paredzēta līmeŋa izvēlei

1. Poga - Home
2. Poga - Options
3. Bulta - Level change
   (3.1)Žests (nobraucot ar pirkstu no labas uz kreiso vai otrādi) - Level change
4. Poga - Play

Apstrāde un Izvade:

1. Ved atpakaļ uz Sākuma ainu
2. Tiek ielādēta iespēju aina, pec kuras nospiežot uz mājas ikonas atgriežas Līmeņa izvēles ainā
3. nomaina izvēlēto līmeni (lai maiŋa notiktu ar žsetu vajag braukt ar pirkstu vai kursoru 1 virzienā , ja kadā brīdī virziens tiek mainīts žests netiek ŋemts vērā)
4. Tiek ielādēts izvēlētais līmenis(bet ja līmenis tiek spēlēts pirmo,neskaitot līmeni nr1 reizi tiek parādīta maza filmiŋa)

1.3.1.5 Speles līmenis

Šajā ainā paredzēts uzrādīt,kontrolēt spēles varoni,pretiniekus un galvenos elementus.

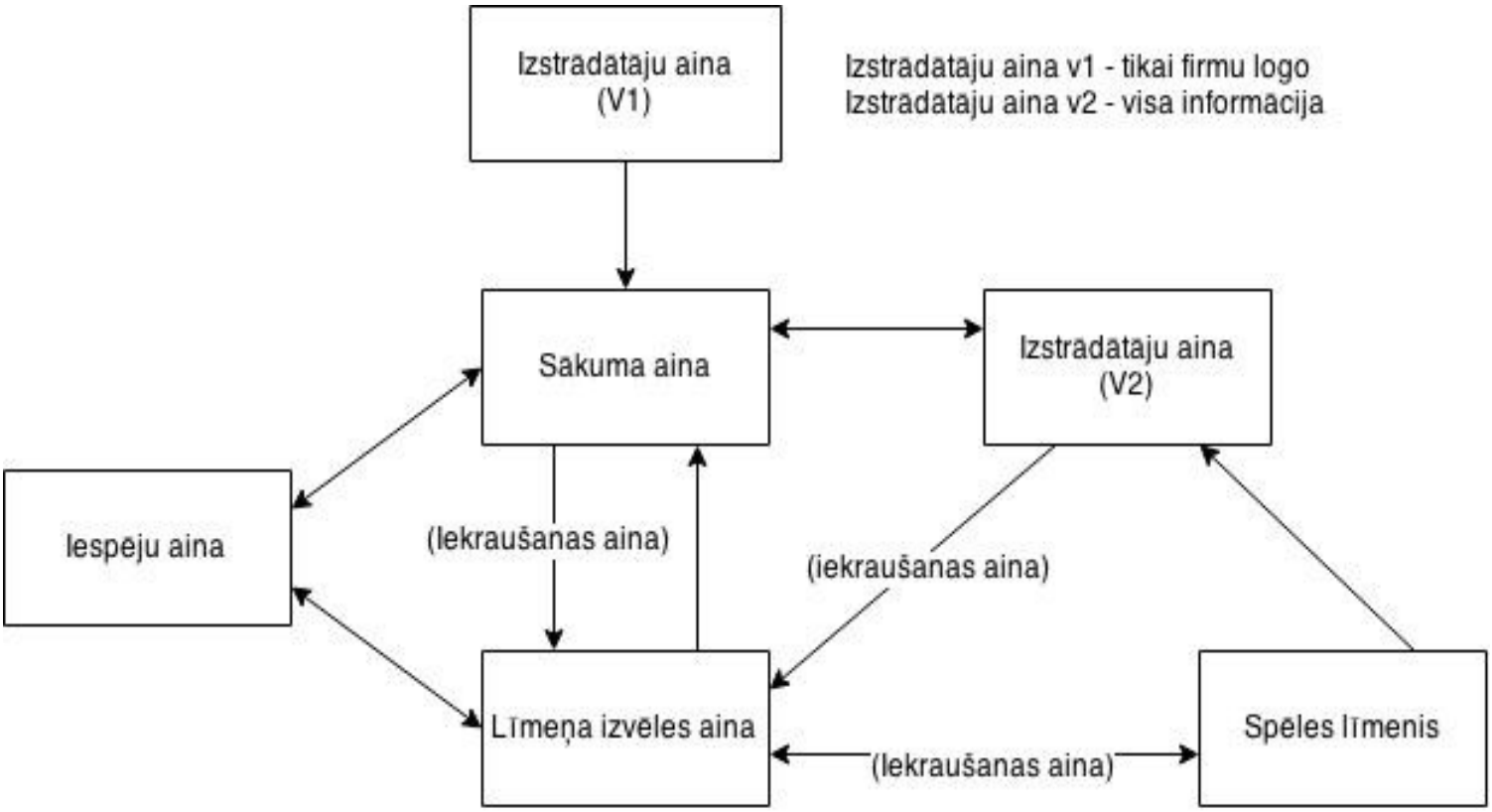
1. Varoņa vadība
2. Pretinieku vadība
3. Poga - Home
4. Gip papildvaroņa vadība
5. Galvenās spēles darbības

Apstrāde un Izvade:

1. Varonim jakust s pa spēles laukumu, kad tiek nospiests pieņemaja zonā. Kad nospiež uz pretinieka, varonim ir jadodas pie pretinieka un tad jauzbrūk un ne ātrāk.
2. Pretiniekam nav kustība, bet jārada skaņa, jo tuvāk spēles varonim, jo skaļāk.Ka arī jāpalaiž galvenās spēles daļas.
3. Ved atpakaļ uz līmeņu izvēli.
4. Vienkāršs sekotāj-algoritms. Seko spēles varonim un spēlē tās pašas animācijas
5. Tiklīdz uzbrūk pretiniekam atveras spēle kurai ir pēc brīvas izvēles jaievieto vajadzīgie elementi un japarāda pareizais spēles veids, ka arī jaapstiprina spēle kad poareiza kombinācija salikta.Pēc spēles parezas apstiprināšanas jaatņem pretiniekam 1 dzīvība, bet ja laiks spēlei beidzies tad jatgriežas spēles laukumā.

2. EKRĀNFORMU PROJEKTĒJUMS

2.1 Ainu karte



Apraksts.

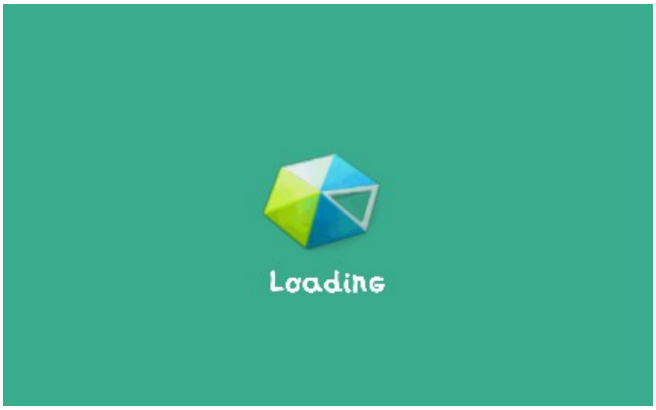
Spēle sākas ar Izsādātāju ainu kura parāda tikai iesaistīto pušu logo.Kad visi logo parādīti talāk dodās uz sākuma ainu no kuras tālak var izslēgt spēli, doties uz līmeņu izvēles,iespēju vai pilno izstrādātāju ainām. Iekraušanas ainas tiek izmantotas, lai iekrautu nākamo ainu un vispirms palaistu filmiņas ainu, ja nepieciešams.

2.2 Spēles sākums

Apraksts.

Pirmais attēls ir tikko spēli palaižot(visi firmu logo) un pēc tam tam nāk sākuma aina kura piedāvā iepēju doties uz iespēju(zobrata ikona),pilno izstrādātāju(i ikona) un līmeņu(start poga) ainām.Ir arī poga kas atļauj iziet no spēles(x).

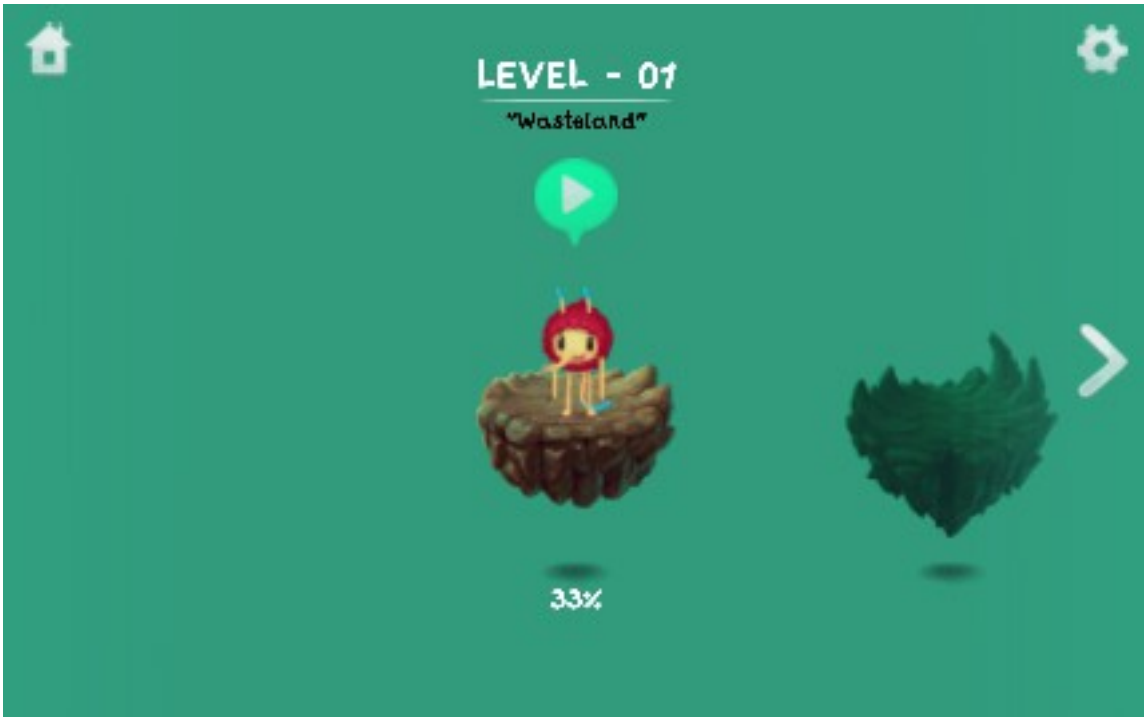
2.3 Iekraušanas un Filmiņas ainas



Apraksts.

Iekrušanas aina ir būtībā uzgaidāmā telpa pirms tiek iekrauta nākamā aina, tas tiek darīts, jo ainas, kas seko paņem vairak laika, lai iekrautu.Ja līmenis, ko vēlās ielādēt, vel nav spēlēts šī aina vispirms parādīs filmiņas ainu un tikai pēc tam pašu līmeni.Filmiņas ainā nospiežot jebkur uz ekrāna nomainās bilde, līdz vairs nav ko rādīt un tad arī nomaina ainas.

2.4 Līmeņa izvēle



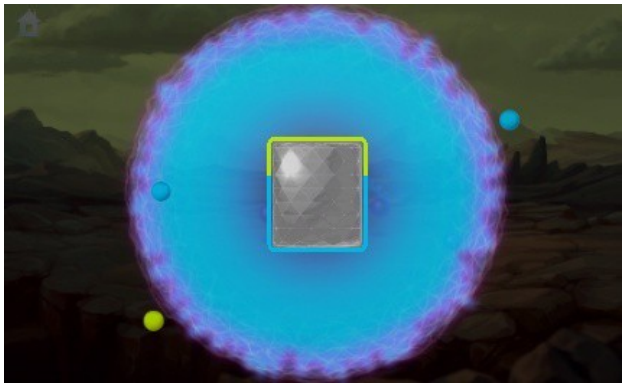
Apraksts

Šajā ainā var izvēleties līmeni ko spēlēt. Lai nomainītu vai janospiež uz bultiņām ekrāna malās vai ar pirkstu viena virzienā janobrauc(no labas uz kreiso vai otrādi) nemainot virzienu. No šīs ainas var doties atpakaļ uz sākuma ainu un iespēju ainu vai spēlēt izvēlēto līmeni nospiežot bultiņas pogu dialoga burbulī virs spēles varoņa.

2.5 Spēles līmenis

Apraksts.

Dotajā attēlā redzams 2. līmenis un katrā līmenī,kur ir kads dialogs pieejams un vel nav redzēts tiek uzrādīts kamēr pa spēles laukumu kustēties nevar līdz dialogs nav pabeigts.Dialoga nākamo teksta rindiņu uzrādīs tiklīdz nospiež jebkur uz ekrāna.Kad dialogs ir pabeigts pa spēles laukumu var virzīties un uzbrukt pretiniekiem. Līmenis iziets,kad visi pretinieki uzvarēti.



Nākamajā attēlā redzam kā izskatās pati spēles darbība (gameplay) un tās elementi. Būtībā ideja ir vienkārša :
Pievieno mazos aplīšus centra objektam un ja uzspiežot uz centra krāsu proporcijas sakrīt ar nepie ciešamajām krāsām pretinieks ir uzvarēts vai vismaz 1 dzīviba tam atņemas(ja ir vairākas).

*Šajā vitā japiezīmē ka pie projekta funkcionalitātes strādaja 2 cilveki, kolēģis sastādija tikai attēla redzamas spēles daļu(Elementu pievienošanu centram, to funkcijas un centra funkcijas ka arī daļu no grafiskā atainojuma) un vel paris sīkas detaļas ārpus šīs daļas.

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1 Ievads

Testēšanas dokumentācijā ir aprakstītas vienībtestēšanas metodes un rezultāti.

3.1.1. Nolūks

Šī dokumenta nolūks ir aprakstīt testēšanas norisi un testēšanas rezultātus. Dokuments domāts programmprodukta pasūtītājam, programmētājam darba pārbaudei un neatkarīgajiem testētājiem.

3.1.2. Testu sagatavošana

Izstrādes laikā, testēšana tika veikta pēc baltās kastes metodes, kad ir zināma informācija par programmas uzbūvi, algoritmiem. Tā tika veikta pēc katra jauna moduļa vai "fīčas" izveides, lai kļūdas gadījumā to varētu izlabot uzreiz.

Izstrādes beigās, testēšanas procesam pieslēdzās neizstrādātāji – testētāji, kas loģiski testēja pēc melnās kastes principa, nezinot loģiku vai struktūru, tie orientējās tikai pēc vizuālā produkta prasībām, vajadzībām.

Kopumā, testēšana bija pašu izstrādātāju veikta, pēc "post factum" metodes, uzreiz pēc katra jauna maza moduļa izveides. Līdz ar nebija strikti noteikti testēšanas plāni, tā kā mēs paši nemitīgi papildinājām spēles saturu, paši arī dabiski jutām, kā tam ir jāizskatās un jāstrādā.

3.2 Testēšanas plāns

Šeit ir aprakstīti kritiskātie testēšanas punkti, kurus varētu skatīt, kā nopietnos testa plānus. Jo teorētiski bija simtiem mazu testu, un rakstīt vai dokumentēt tos makulatūras pēc nebūtu saprātīgi.

Testējamie vienumi:

Sākuma aina:
- Iespēju iziet no lietotnes
- Pogu darbība(jaunu ainu ielāde)

Izstrādātāju aina:
- Bilžu maiņa
- iepriekšējās ainas ielāde

Iespēju aina:
- On/Off pogu darbība
- Noklusēto vērtību atjaunošana

Iekraušanas aina:
- "Filmiņas" ielāde
- Nākamās ainas ielāde

Līmeņu izvēles aina:
- Līmeņu spārslēgšana
- Vizuālai reprezentējums
- Pogu darbība un jaunu ainu ielāde
- Līmeņa ielāde

Spēles līmeņi
- Varoņa kustība
- Varoņa iespēja uzbrukt pretiniekam
- Varoņau un pretinieku vizuālais reprezentējums
- Spēles galvenās darbības(objektu ieslēgšana/izslēgšana,elementu radīšana,pareiza rezultāta apstiprināšana uc.)
- Fona objektu kustibas

Visi testu rezultāti ir pozitīvi, vai apmierinoši. Testēt gatavus moduļus, kā viņi šeit ir izklāstīti nav/nebija iespējams, jo spēles struktūra tika veidota pakāpeniski no visa pa druskai.

4. KVALITĀTES NODROŠINĀŠANA

Programmu veidoja 2 cilveki gandrīz neatkarīgi viens no otra, tik beigās abas programmas daļas tika apvienotas. Komentari manam kodam sarakstīti pietiekami lai nākamais cilvēks kas strādātu saprastu, kas un kur atrodas, bet tā kā lielako daļu projekta veidoju es, izvērstu komentēšanu nevajadzēja ne arī papildus dokumentāciju.

Kvalitāte tiek nodrošināta ar apstiprinājumu no testētaju un pasūtītāju puses, ka viņus viss apmierina. Pēc kā kvalitates nodrošināšana pārtop par kvalitātes uzturēšanu, ja gadījumā pasūtītītājiem ir vēl kādas vēlmes.
Pašas programmatūras izstrāde tika veikta maksimāli objektorientētā programmēšana stilā un piegājienā, kas maksimāli atvieglo darbu jaunu ideju ieviešanai.

5. KONFIGURĀCIJAS PĀRVALDE

Programmai kopijas tika veidotas ik pa nedēļai un vairākas uzbūvētās programmas versijas testēšanai.

Projekta programmatūra bija nepieciešama man un retu reizi otram projekta programmētājam, līdz ar to GIT vai citus versijas salīdzināšanas, kopošnas rīkus izmantot nebija vajadzīgs.

6. DARBIETILPĪBAS NOVĒRTĒŠANA

Izstrādes sākumā projekts bija plānots ar vairāk iespējam un lietām,bet tā kā termiņi spieda ļoti, nācas dažas lietas atlikt, bet kopumā projekta izmērs ir diezgan lielas prieks 1 vai pat 2 cilvēkiem. Darbietilpības noteikšana šādam darbam ir ļoti grūta, un viss beigās izriet no paša izstrādātāja pieredzes un spējas strādāt ātri. Noteikt darba ietilpību pēc kodu rindiņu skaita būtu muļķīgi, tā kā spēļu rakstīšana nav vienkārši kodēšana, bet gan interesantu struktūru izveide – programmēšana, kods pats tiek nemitigi mainītas un labots lai konkrētas funkcijas darbotos. Spēles kodu rindiņu skaits, ko uzrakstīju, mērojams plus/mīnus 5600.
Kopumā spēle tika izstrādāta 2,5 mēnešos strādājot katru dien 8h

Tā kā nebiju strādājis pie neviena projekta iepriekš nespēju īsti noteikt cik ilgi izstrāde aizņems.

NOBEIGUMS

Pie šāda projekta darbojos pirmoriez kas iekļauj Unity izstrādes vidi un C# valodu.Darbs šķita ļoti interesants, kas iedvesmo vel turpināt spēļu izstrādi nākotnē. Pati programma lai gan spēlējama uz Windows,Android un IOS operētājsistēmām dotajā momentā plānota palaist tikai uz IOS un bus pieejama "App store" tuvākajālaikā.Pagaidām spēle ir lokals risinājums, bet tas vai būs vēlak pieejami rezultātu tablo globāli vel ir nenoteikts.

IZMANTOTĀ LITERATŪRA UN AVOTI

1. „LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis”
http://estudijas.lu.lv/mod/resource/view.php?id=131427

2. „LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai" -
http://estudijas.lu.lv/mod/resource/view.php?id=131428

3. Ar Unity3D dzinēju saistītām problēmām un aprakstiem

http://answers.unity3d.com/index.html
http://forum.unity3d.com/

4. uz daudzi citi mazi artikuli no neskaitāmiem forumiem saistība ar kļūdu labojumiem vai funkciju realizācijām.

PIELIKUMI

Programmatūras koda fragmenti

Gtimer.cs

```
/*---------------------------------------------------------------------------------------------------------
V1 code put together at : 6/5/2015
About : Gameplay timer - controls when,if and how gameplay is finished
Code is UNOPTIMIZED
By Andrejs Jauja
---------------------------------------------------------------------------------------------------------*/
using UnityEngine;
using System.Collections;

public class GTimer : MonoBehaviour {

    public float TimeSet;
    public bool Active = false,
                Completed = false,
                paused = false;
    public float TimeLeft;
    GameObject Trans;
    Animator TAnim;
    GEventHandler GEH;
    GameplayController GC;
    JifoActions JA;
    SoundController SC;

    void Start()
    {
        Trans = GameObject.Find("Transition");
        TAnim = Trans.GetComponent<Animator>();
        GC = GameObject.Find("Gameplay").GetComponent<GameplayController>();
        JA = GameObject.Find("Jifo").GetComponent<JifoActions>();
        GEH = GameObject.Find("Gameplay").GetComponent<GEventHandler>();
        SC = GameObject.Find("SoundManager").GetComponent<SoundController>();
    }

            // Update is called once per frame
            void Update ()
    {
        //active timer
        if ((Active)&&(!paused))
        {
            TimeLeft -= Time.deltaTime;

            float scale = 2 + (5 * (TimeLeft/TimeSet));
            transform.localScale = new Vector3(scale,scale,1);
            //if one of the end statements are true
            if ((TimeLeft < 0)||(Completed))
            {
                Active = false;

                //enable normal gameplay
                GameObject Jifo = GameObject.Find("Jifo");
                Jifo.GetComponent<Animator>().SetBool("Shoot",false);
                Trans.GetComponent<SpriteRenderer>().enabled = true;
```

```
            TAnim.enabled = true;
            JA.Trans_out = true;
            JA.Reset();
            //change corrupted being
            if (Completed)
            {
                GEH.PutEvent(0, GEventType.GameplayEndGood);
                JA.EnemyToAttack.GetComponent<ChartInteraction>().Die();
                Trans.GetComponent<SpriteRenderer>().color = new Color(1f, 1f, 1f);
            }
            else
            {
                //play failed
                SC.PlaySound(20);
                GEH.PutEvent(0, GEventType.GameplayEndBad);
                Trans.GetComponent<SpriteRenderer>().color = new Color(1f, 0f, 1f);
            }
            Trans.GetComponent<Animator>().Play("Trans_Out");
            //disable gameplay
            GameObject.Find("Gameplay").GetComponent<GameplayController>().HideGameplay();
        }
    }
        }

    //put timer on pause & disables coliders so they acnnot be moved
    //(needed for dialogs but can be use for other stuff)
    public void Pause(bool b)
    {
        paused = b;
        foreach(Transform t in transform.parent)
        {
            int GP = GlobalVars.Levels[GC.LevelId].Enemies[GC.MonstId].Lives -
JA.EnemyToAttack.GetComponent<ChartInteraction>().Lives;
            if (((GlobalVars.Levels[GC.LevelId].Enemies[GC.MonstId].Gameplay[GP].Type == 0) && (t.name ==
"CirclePref"))||
                ((GlobalVars.Levels[GC.LevelId].Enemies[GC.MonstId].Gameplay[GP].Type == 1) && (t.name ==
"squarePref"))||
                ((t.name != "squarePref")&&(t.name != "CirclePref")))
                foreach (Transform t2 in t)
                {
                    if (t2.collider2D)
                        t2.collider2D.enabled = !b;
                }
            if (t.collider2D)
                t.collider2D.enabled = !b;
        }
    }
    //sets up timer for work
    public void StartTimer(float time)
    {
        if (!Active)
        {
            TimeSet = time;
            TimeLeft = time;
            Active = true;
            Completed = false;
            paused = false;
            GEH.PutEvent(0, GEventType.GameplayStart);
        }
    }
```

ButtonPositioner.cs

```
/*-------------------------------------------------------------------------------------------------
```

```
V1 code put together at : 6/5/2015
About : Positions object this script is attached to to one of the sides or corners of the screen
 * needs colider(2d box one)
Code is UNOPTIMIZED
By Andrejs Jauja
----------------------------------------------------------------------------------------------------*/
using UnityEngine;
using System.Collections;

public class ButtonPositioner : MonoBehaviour {

    public enum PlaceOnScreen
    {
        Top,
        TopRight,
        TopLeft,
        Left,
        Bottom,
        BottomRight,
        BottomLeft,
        Right
    };

    public PlaceOnScreen POS;

            // Use this for initialization
            void Start () {
        //top / bot CENTER!
        Vector3 place = Camera.main.WorldToScreenPoint(transform.position);
        switch (POS)
        {
            case PlaceOnScreen.Top:
            case PlaceOnScreen.TopLeft:
            case PlaceOnScreen.TopRight:
                place.y = Screen.height;
                break;
            case PlaceOnScreen.Bottom:
            case PlaceOnScreen.BottomLeft:
            case PlaceOnScreen.BottomRight:
                place.y = 0;
                break;
        }
        switch (POS)
        {
            case PlaceOnScreen.BottomLeft:
            case PlaceOnScreen.TopLeft:
            case PlaceOnScreen.Left:
                place.x = 0;
                break;
            case PlaceOnScreen.BottomRight:
            case PlaceOnScreen.TopRight:
            case PlaceOnScreen.Right:
                place.x = Screen.width;
                break;

        }
        transform.position = Camera.main.ScreenToWorldPoint(place);
        //NOT THE MOEST EFFICIENT WAY OF DOING!
        switch (POS)
        {
            case PlaceOnScreen.TopRight:
            case PlaceOnScreen.BottomRight:
```

```
            case PlaceOnScreen.Right:
                transform.position -= new Vector3(GetComponent<BoxCollider2D>().size.x / 2, 0, 0) *
transform.localScale.x * (transform.localScale.x < 0 ? -1 : 1);
                break;
            case PlaceOnScreen.BottomLeft:
            case PlaceOnScreen.TopLeft:
            case PlaceOnScreen.Left:
                transform.position += new Vector3(GetComponent<BoxCollider2D>().size.x / 2, 0, 0) *
transform.localScale.x * (transform.localScale.x < 0 ? -1 : 1);
                break;
        }
        switch (POS)
        {
            case PlaceOnScreen.Top:
            case PlaceOnScreen.TopLeft:
            case PlaceOnScreen.TopRight:
                transform.position -= new Vector3(0, GetComponent<BoxCollider2D>().size.y / 2, 0) *
transform.localScale.y * (transform.localScale.y < 0 ? -1 : 1);
                break;
            case PlaceOnScreen.Bottom:
            case PlaceOnScreen.BottomLeft:
            case PlaceOnScreen.BottomRight:
                transform.position += new Vector3(0, GetComponent<BoxCollider2D>().size.y / 2, 0) *
transform.localScale.y * (transform.localScale.y < 0 ? -1 : 1);
                break;
        }
            }

}
```

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: ***Andrejs Jauja***_____ **___.05.2015**.

Rekomendēju darbu aizstāvēšanai

Darba vadītāj**s**: ***Dr. dat. Jānis Zuters*** _____ **___.05.2015.**

Recenzents: ***M.dat. Jānis Mārtužs***

Darbs iesniegts 01.06.2015.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova*_____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

___.06.2015. prot. Nr. _____

Komisijas sekretārs(-e): _____