Razvoj računalne igre u okruženju Cocos2d-x

Ivanda, Tomislav

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:559804

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-16**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

TOMISLAV IVANDA

RAZVOJ RAČUNALNE IGRE U OKRUŽENJU COCOS2D-X

Završni rad

Pula, rujan, 2019.

Sveučilište Jurja Dobrile u Puli

Fakultet informatike u Puli

TOMISLAV IVANDA

RAZVOJ RAČUNALNE IGRE U OKRUŽENJU COCOS2D-X

Završni rad

JMBAG: 0303069498, redoviti student

Studijski smjer: Informatika

Predmet: Programsko inženjerstvo

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. dr. sc. Tihomir Orehovački

Pula, rujan, 2019.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Tomislav Ivanda, kandidat za prvostupnika Informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, 17.9. 2019 godine

IZJAVA

o korištenju autorskog djela

Ja, Tomislav Ivanda dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom “Razvoj računalne igre u okruženju Cocos2d-x” koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 17.9.2019 godine

Potpis

_____

Sažetak

Igra GokuFly vraća igrače u vremena kada je igrica „Flappy Birds“ bila aktualna. Igrica se sastoji od lika kojim upravlja igrač stiskanjem tipke enter i cijevi koje se nasumično pojavljuju ne bi li svojim zahtjevnim pozicijama što prije omele lika i tako mu nanijele poraz. Cilj ove igre je pokušati proći kroz što više cijevi ne dirajući ih i tako postavljati osobne rekorde. Kretanje samog lika nije kao u običajnoj prvobitnoj igrici da ga gravitacija vuče prema dolje nego je ona izmijenjena i vuče ga prema gore što još više otežava samu igru.

KLJUČNE RIJEČI: *Cocos2dx, GokuFly, 2d igra, Notepad++,Html5, JS*

Abstract

GokuFly brings players back to the time when Flappy Birds was current. The game consists of a character controlled by the player by pressing the enter key and pipes that appear randomly in order to distract the character as quickly as possible and cause him defeat. The goal of this game is to try to get through as many tubes as possible without touching them and thus setting personal records. The movement of the character itself is not like in the original original game that gravity pulls it down but it is altered and pulls it up which further complicates the game itself

KEYWORDS: *Cocos2dx*, *GokuFly, 2d game , Notepad++, Html5, JS*

Sadržaj

1. Uvod 7
2. Prezentacija igrice 8
2.1 Početni zaslon sa uputama 8
2.2 Početak igrice 9
2.3 Igrica u tijeku 10
2.4 Kraj igrice 11
3. Prikaz stvaranja igrice 12
3.1 Dodavanje i namještanje pozadine 12
3.2 Detekcija dodira na ekranu 14
3.3 Dodavanje lika 16
3.4 Dodavanje funkcije skakanja 18
3.5 Dodavanje kretnji sporednih elemenata 19
3.6 Dodavanje cijevi i njihov načina rada 21
3.7 Stvaranje cijevi na ekranu 23
3.8 Dodavanje kolizije 24
3.9 Dodavanje teksta 25
4. Zaključak 26
5. Literatura 27
6. Popis slika 28

1.Uvod

GokuFly je igrica napravljena po uzoru na prijašnju igricu „Flappy Birds. Ovdje se radi o 2d igrici koja se sastoji od jednog lika kojemu je ime Goku i koji je ime dobio po liku iz crtane serije Zmajeva kugla. Njegov cilj je proći kroz što više cijevi a da ih pritom ne dira. Cijevi se stalno stvaraju i nikad se ne zna gdje će se sljedeća stvoriti što daje samoj igri dodatnu čar i napetost . Sama igrica zahtjeva maksimalnu koncentriranost igrača da što bolje procjeni padanje lika i vrijeme kada mora stisnuti tipku za novi skok . Komande za igranje ove igrice nisu kompleksne te se radi samo o pritisku tipke na mišu da bi se dobio novi skok našeg lika. Jednostavnost komande je dodana da bi se mogla prilagoditi široj masi ljudi počevši od najmlađih. Smatram da kod njih još nije toliko razvijena sama motorika i osjećaj za dodir više tipki za kretanje po ekranu te da bi igranost bila minimalna.

Ono što daje drugu dimenziju i novitet igrici je to što lik ne skače prema gore i gravitacija ga ne vuče prema gore , nego ga gravitacija vuče prema dolje i tako pritiskom tipke on pokušava ostati što bliže zemlji.

Za stvaranje ove aplikacije korišteno je razvojno okruženje Cocos2d-x.Ovo razvojno okruženje pruža više načina za stvaranje igrica kao što je Cocos Creator i HTML5Lite. Cocos2d-x je 2D platforma za razvoj mobilnih igara s otvorenom i cross platformom otvorenog koda, poznat po brzini, stabilnosti i lakoći uporabe.

Ova dokumentacija sadrži prikaz izrade same igrice od početka do kraja . U prvom dijelu predočit ću Vam kroz slikovni prikaz kako izgleda finalna verzija igrice i detaljno opisati prikaze zaslona , dok ću u idućim dijelovima opisivati dijelove koda što znači same funkcije koje su bitne da igrica uopće radi . Kod prikazan na slikama neće biti cjelovit te će biti izdvojeni samo bitni dijelovi jer neki se atributi prepliću i pozivaju na više mjesta.

2. Prezentacija igrice

2.1 Početni zaslon sa uputama

Početni zaslon sastoji se od imena same igrice i opisa što igrač mora raditi da bi ostao što duže na životu. Ispod toga nalazi se naputak da pritiskom bilo gdje na ekranu igrač pokreče samu igricu .

Slika 1. Početni zaslon sa uputama

2.2 Početak igrice

Na ovoj slici vidimo lika te okruženje u kojem se on nalazi . Cijevi u igrici ne vidimo jer se one ne pojavljuju odmah . Uz lika vidimo još stabla i oblake koji su u stalnoj kretnji kako bi dali doživljaj stvarne kretnje lika . U gornjem lijevom kutu vide se dva broja . Prvi gornji broj označava trenutni rezultat koji je ostvario lik , a broj ispod njega označava najbolji postignuti rezultat koji je ostvario sami lik.

Slika 2. Početak igrice

2.3 Igrica u tijeku

Kada je igrica u tijeku na ekranu se stvaraju cijevi koje se nasumično pojavljuju te se nikad ne zna gdje će biti sljedeća. Gore u kutu vidimo kako se brojevi mijenjaju i kako prvi broj označava da je lik prošao do sad tri cijevi , a da mu je dosadašnji najbolji rezultat bio dva .

Slika 3. Igrica u tijeku

2.4 Kraj igrice

Na samome kraju igrice kada igrač dodirne cijev ili nebo igra prestaje. Pred igračem se pojavljuje natpis na ekranu da je igra gotova , a ispod toga stoji natpis vezan uz kreatora igrice ,njegovog mentora i ustanovu. U gornjem lijevom kutu se vidi da je trenutni rezultat izbrisan zato što je igra stala , a da se novi najbolji rezultat upisao dolje. Na slici se vidi zadnja pozicija lika prije nego što je igrač pogriješio i ta scena traje 4 sekunde dok se ponovno ne učita početni ekran gdje igrač može ponovno započeti igru.

Slika 4. Kraj igrice i prikaz credita

3. Prikaz stvaranja igrice

3.1 Dodavanje i namještanje pozadine

Prva stvar koja je bitna za napraviti je učiniti to da se ekran igrice može razvlačiti te da se tim razvlačenjem ne gubi slika nego da se automatski i proporcionalno smanjuje/povećava. U kodu sa slike 5 su dodane i dimenzije samog ekrana , kao primjer koda skaliranja poslužila je igrica „Moon Warrior“ [2][3] .

```
window.onload = function(){

    var targetWidth = 960;
    var targetHeight = 640;

    cc.game.onStart = function(){

        cc.view.adjustViewPort(false);
        cc.view.setDesignResolutionSize(targetWidth, targetHeight, cc.ResolutionPolicy.SHOW_ALL);
        cc.view.resizeWithBrowserSize(true);
        //load resources
        cc.LoaderScene.preload(["slike/ed"], function () {
        cc.director.runScene(GameLayer.scene());
      }, this);
    };
    cc.game.run("gameCanvas");
};
```

Slika 5. Mogućnost skaliranja

```
    init:function() {

        this._super();
        var size = cc.director.getWinSize();


        var bgSprite = cc.Sprite.create(res.bg_slike);
        bgSprite.setPosition(size.width / 2, size.height / 2);
        this.addChild(bgSprite, indexbg);

        this._floor = cc.Sprite.create(res.floor_slike);
        this._floor.setPosition(0, 0);
        this._floor.setAnchorPoint(0,0);
        this.addChild(this._floor, indexfloor);
```

Slika 6. Game.js kod za dodavanje pozadine

```
var res = {};

res.bg_slike = 'slike/bg.jpg';
res.floor_slike = 'slike/floor.png';
```

Slika 7. gamemanager.js kod za prikazivanje puta

Slike šest i sedam prikazuju dodavanje pozadinske slike , poda i načina na koji se slike vade iz datoteka [4] . Indeks komponenti(slika) ima funkciju prikaza kako će koja komponenta stajati, hoće li pozadina biti vidljiva ispred poda ili obrnuto , a taj sami indeks je dodan isto u skriptu „gamemanager.js“ pod nazivima indeks poda(„indexfloor“), i indeks pozadine(„indeksbg“). Svaki taj indeks sadrži svoj broj[1].

3.2 Detekcija dodira na ekranu

Ovdje vidimo sličnu sintaksu kao u C++ za dodavanje detekcije dodira. Potrebno je bilo dodati „Listener“ koji će sadržavati argumente koji će na kraju slušati sami „Listener“. Dodavanje dodira jedan po jedan kako bi detektirali pritisak miša . Uz sve to vidimo kod koji sadrži funkcije kada je kraj, početak ili samo vrijeme dodira koji javlja kada je došlo do detekcije pokreta , a te podatke možemo vidjeti kada pritisnemo provjeri u pregledniku . Na samom kraju funkcije „On Touch Began „ korištena je negacija jer program ne treba javiti lokaciju miša kada je miš stisnut i kada se kreče nego samo kada korisnik stisne bez zadrške[2][4].

```
onEnter:function() {
    this._super();
    cc.eventManager.addListener({
        event: cc.EventListener.TOUCH_ONE_BY_ONE,
        swallowTouches: TRUE,
        onTouchBegan: this.onTouchBegan,
        onTouchMoved: this.onTouchMoved,
        onTouchEnded: this.onTouchEnded
    }, this);

    this.schedule(this.onTick);

    this.StopGame();
    this._processTouch = TRUE;

},
```

Slika 8. Prikaz koda za detekciju pokreta

```
    onTouchBegan:function(touch, event) {
        var tp = touch.getLocation();
        var tar = event.getCurrentTarget();

        if(tar._processTouch == TRUE) {
            tar._goku.SetStartspeed();

            if(tar._gameStarted == FALSE) {
                tar.StartGame();

            }
        }

        return false;
    },

    onTouchMoved: function(touch, event) {
        var tp = touch.getLocation();
        console.log('onTouchMoved:' + tp.x.toFixed(2) + ','  + tp.y.toFixed(2));
    },

    onTouchEnded:function(touch, event) {
        var tp = touch.getLocation();
        console.log('onTouchEnded:' + tp.x.toFixed(2) + ','  + tp.y.toFixed(2));
    },
```

Slika 9. Prikaz koda u game.js-u

3.3 Dodavanje lika

Slika broj 9 prikazuje skriptu „Goku.js“ koja je kreirana za dodavanje funkcija i kreiranje samog lika [2] . Skripta se sastoji od „Update“ , „Reset“ , „Set Start Speed“ i „Tube Collision Box“ . Svaka od njih ima svoju ulogu. Funkcija „Updateggoku“ ima dt(delta time) što prikazuje vrijeme od zadnjeg ponovnog stvaranja samog lika. „Reset“ funkcija vrača lika na početnu poziciju , a „SetStartSpeed“ vuče podatak o brzini lika iz skripte „Gamemanager.js“ .Funkcija „TubeCollisionBox“ predstavlja mjere koje lik ima a nisu vidljive, a korigirane su tako da odgovaraju propozicijama lika da kad rub lika dotakne cijev igrica odmah detektira dodir.

```
var gokuSprite = cc.Sprite.extend( {

    state: gokuStateStopped,
    speedY: 0.0,
    topOfScreen: 0,

    ctor: function (SpriteFrameName) {
        this._super(SpriteFrameName);

    },

    Updategoku: function(dt) {
        if(this.state == gokuStateMoving) {
            var distance = 0;
            var newSpeed = 0;

            distance = this.speedY * dt + 0.5 * GRAVITY * dt * dt;
            newSpeed = this.speedY + GRAVITY * dt;

            this.y = this.y + distance;
            this.speedY = newSpeed;

            if(this.y < 0 ) {

                this.y = 0;
                this.speedY = 0,0;
            }
        }
    },

    Reset : function() {
        this.state = gokuStateStopped;
        this.SetStartSpeed;
        },

    SetStartspeed: function(){
        this.speedY = gokuStartSpeedY;
    },

    TubeCollisionBox: function() {
        return new cc.Rect(
```

Slika 10. Prikaz koda za dodavanje lika

Na slici je prikazan kod koji služi za stvaranje lika u skripti „game.js“ . Postupak je sličan kao za dodavanje pozadine i poda. Poziva se slika i indeks samog lika koji se nalaze u skripti „gamemanager.js“ .

```
this._goku = new gokuSprite(res.goku_slike);
this._goku.x = gokuStartX;
this._goku.Reset();
this.addChild(this._goku, indexgoku);
```

Slika 11. Prikaz kreiranja lika u game.js

3.4 Dodavanje funkcije skakanja

Na slici broj 11 vidimo funkciju kojom se definira mogućnost skakanja lika u igrici. Kada lik dirne strop igrica staje i lik se vrača na početnu poziciju iz koje je krenuo [1].

```
    onTick:function(dt){
        if(this._goku.y < this.topOfScreen) {
            this._goku.Reset();
            this._goku.< = cc.director.getWinSize().height /2;
        }
        this._goku.Updategoku(dt);
    },
```

```
    Reset : function() {
        this.state = gokuStateStopped;
        this.SetStartSpeed;

    },
```

Slika 12. Prikaz funkcije namijenjene skakanju

3.5 Dodavanje kretnji sporednih elemenata

U ovom dijelu igrice dodana je skripta „oblak.js“ , napravljena je po uzoru na skriptu „goku.js“ zato i sadrži slične varijable . Svi dijelove oblaka i stabala mogu se svrstati pod jednu funkciju „AddOblak“ jer će se svi kretati identično s desna na lijevo pa ih je zato bolje grupirati da se ne radi više nepotrebnih funkcija . Skripta „Oblak.js“ sadrži funkcije „Stop“, „Start“, „ReachedDestination“ [4] . Kod sa slike označava distance koju oblak mora preći kretanjem s desne na lijevu stranu i kada dođe do kraja ekrana tada „xoffset“ prelazi u negativno i onda je željena destinacija ispunjena . Izvršenjem zadatka i dobivanjem željene destinacije oblak i stabla se resetiraju opet na desnu stanu. U „gamemanageru.js“ je sadržan indexi, brzina i mjesto odakle će se vaditi slike . Ovdje prikazane funkcije „OblakSlowSpeed“ i „OblakFastSpeed“ predstavljaju velike i male oblake te njihovu brzinu .

```
    Stop : function() {
        this.stopAllActions();

    },

    ReachedDestination : function(sender) {
        sender.x = sender.xOffset + sender.screenWidth;
        sender.Start();
    },


    Start:function() {
        this.stopAllActions();

        var currentX = this.x;
        var distance = currentX - -(this.xOffset);
        var time = distance / this.pixelsPerSecond
        var destination = cc.p(-this.xOffset, this.y);

        var actionMove = cc.moveTo(time, destination);
        var actionMoveDone = cc.callFunc(this.ReachedDestination, this);

        this.runAction(cc.sequence(actionMove, actionMoveDone));

    }

);
```

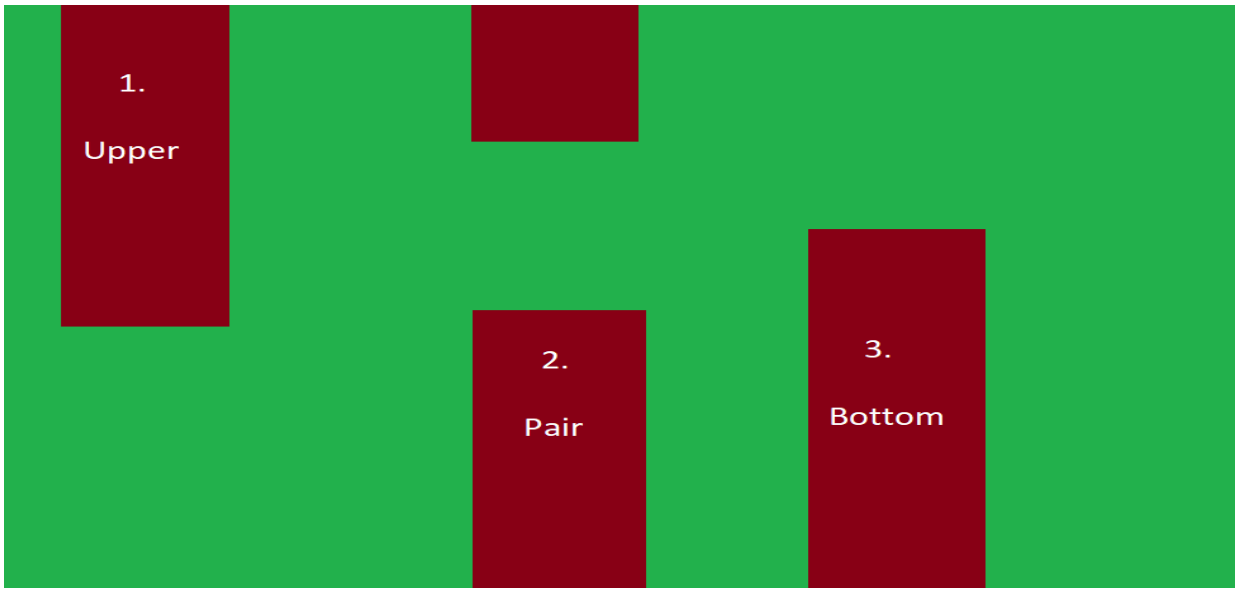
Slika 13. Funkcije u oblak.js

```
AddOblak: function(speed, position, scale, zIndex, name, XOffset) {
    var screenSize = cc.director.getWinSize();
    var Oblak = new OblakSprite(name);
    Oblak.SetSpeedAndWidth(speed, screenSize.width, XOffset);
    Oblak.x = position.x;
    Oblak.y = position.y;
    Oblak.setScale(scale);
    this.addChild(Oblak, zIndex);
    ArrayOblaks[ArrayOblaks.length] = Oblak;

},

CreateOblaks: function() {
    var FileName = res.oblak_slike;

    this.AddOblak(OblakSpeedSlow, cc.p(700,610), OblakScaleSlow, indexOblakSlow, FileName, OblakRestartX);
    this.AddOblak(OblakSpeedSlow, cc.p(150,570), OblakScaleSlow, indexOblakSlow, FileName, OblakRestartX);

    this.AddOblak(OblakSpeedFast, cc.p(150,300), OblakScaleFast, indexOblakFast, FileName, OblakRestartX);
    this.AddOblak(OblakSpeedFast, cc.p(400,500), OblakScaleFast, indexOblakFast, FileName, OblakRestartX);
    this.AddOblak(OblakSpeedFast, cc.p(880,400), OblakScaleFast, indexOblakFast, FileName, OblakRestartX);

    FileName = res.stablo_slike;
    this.AddOblak(TreeSpeed, cc.p(128,72), TreeScale, indexTree, FileName, OblakRestartX);
    this.AddOblak(TreeSpeed, cc.p(624,72), TreeScale, indexTree, FileName, OblakRestartX);
    this.AddOblak(TreeSpeed, cc.p(864,72), TreeScale, indexTree, FileName, OblakRestartX);


},
```

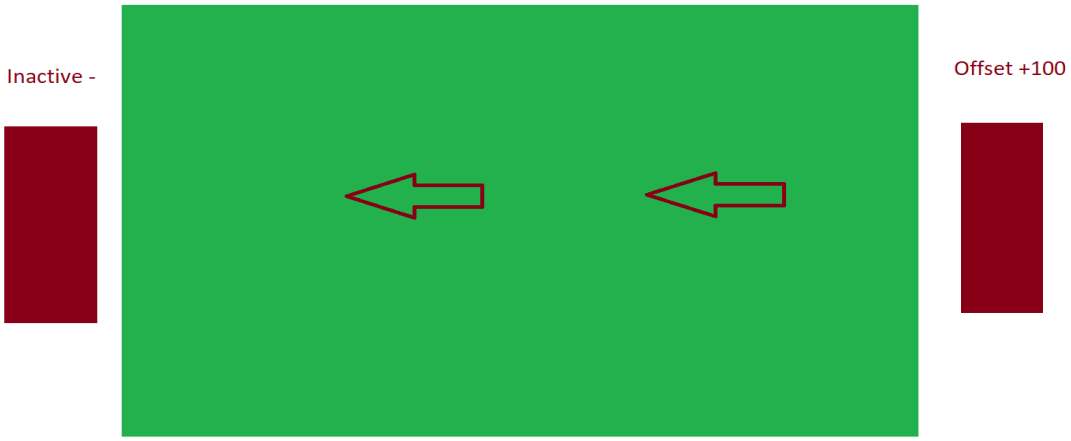
Slika 14. Funkcije vezane za kretanje u game.js

3.6 Dodavanje cijevi i njihov načina rada

Postoje 3 para cijevi u igrici : „pair“ , „upper“ , „lower“. Da se izbjegne preveliko odmicanje cijevi na ekranu definirani su „SingleGapTop“ i „SingleGapBottom“ koji prate kretanje tubi u novom stvaranju , ali sa određenom granicom i to vrijedi za sve [1] . Kao na prijašnjoj slici „xOffset“ služi za kontrolu micanja cijevi , kada cijev dođe do kraja ekrana „xOffset“ prelazi u ne aktivno stanje tj. prelazi u minus.



Slika 15. Vrste cijevi



Slika 16. Prikaz kako je zamišljen „Offset“

Slika broj 17 predstavlja jednu od 3 funkcije za stvaranje cijevi. Ovdje se bira koja će se vrsta cijevi prikazati , a to se određuje narednom „if“ . Problem je ovdje što igrica neće znat koji je zadnji oblik cijevi bio prikazan u igrici zato se dodaje „while“ za provjeru[1].

```
SpawnNewTubes: function(){

    var ourChance = Math.floor((Math.random() * 3) +1);

    while(1){
        if(this._lastTubeType == TubeTypeUpper && ourChance == 1){
            ourChance = Math.floor((Math.random() * 3) + 1);
        } else if(this._lastTubeType == TubeTypeLower && ourChance == 2){
            ourChance = Math.floor((Math.random() *3) +1);
        } else if(this._lastTubeType == TubeTypePair && ourChance == 3){
            ourChance = Math.floor((Math.random() *3) +1);
        }else {
            break;
        }
    }

    if(ourChance == 1) {
    this.SpawnUpperOrLower(TRUE);
    }else if (ourChance == 2) {
        this.SpawnUpperOrLower(FALSE);
    }else {
        this.SpawnTubePair();
    }
},
```

Slika 17. Dio koda za stvaranje cijevi

3.7 Stvaranje cijevi na ekranu

Za stvaranje stvarnog prikaza cijevi bilo je potrebno dodati novu skriptu koja je bila nazvana „Tube.js“ . Po svom izgledu jako je slična skripti „Oblak.js“ , ali se naravno razlikuje po nekim funkcijama . Slika cijevi dodana je u „game.js“ točnije u funkciji „GetNextTube“ te nije kao u prijašnjim stvaranjima dodana u „init“ funkciju . „GetStart“ funkcija sadrži stopiranje radnje , kalkulira se distanca i vrijeme koje treba proći da dođe do korištenja „offseta“ te se onda stvaraju vidljive cijevi i naredba „action move“ pokreće njihovo kretanje.

```
    Stop : function() {
        this.stopAllActions();

    },

    ReachedDestination : function(sender) {
        sender.x = sender.xOffset + sender.screenWidth;
        sender.Start();
    },


    Start:function() {
        this.stopAllActions();

        var currentX = this.x;
        var distance = currentX - -(this.xOffset);
        var time = distance / this.pixelsPerSecond
        var destination = cc.p(-this.xOffset, this.y);

        var actionMove = cc.moveTo(time, destination);
        var actionMoveDone = cc.callFunc(this.ReachedDestination, this);

        this.runAction(cc.sequence(actionMove, actionMoveDone));

    }

});
```

Slika 18. Funkcije u skripti „Tube.js“

3.8 Dodavanje kolizije

U funkciji „OnTick“ dodana je od prije „if“ funkcija koja radi to da ako lik dodirne nebo igrica završava . Sada je tu još dodan „else“ koji za svako diranje cijevi prekida igru. Na slici 20 vidimo „Boundingbox“ koji predstavlja mjere koje lik ima a nisu vidljive. Mjere su korigirane tako da odgovaraju propozicijama lika da kada rub lika dotakne cijev ili nebo igrica odmah detektira koliziju [1][2][5] .

```
if(this._goku.y > this._goku.topOfScreen){
        gameOver = TRUE;
}else {
    var GokuCollBox = this._goku.TubeCollisionBox();
    for(var i = 0 , len = ArrayTubes.length; i < len; i++) {
        if(ArrayTubes[i].state == TubeStateActive) {
            if(cc.rectIntersectsRect(ArrayTubes[i].getBoundingBox(), GokuCollBox) == TRUE) {
                console.log('COLLISION');
                gameOver = TRUE;
```

Slika 19. Dodavanje „else“ u „onTick“ funkciju

```
    TubeCollisionBox: function() {
        return new cc.Rect(
        this.getBoundingBox().x + 25,
        this.getBoundingBox().y + 5,
        this.getBoundingBox().width  -40,
        this.getBoundingBox().height +30,
        );
    }
```

Slika 20. Korigiranje mjera za veličinu našeg lika

3.9 Dodavanje teksta

Na slici 21 definirane su varijable u skripti „gamemanager.js“ sa svojim značenjima i vrijednostima te su poslije kao takve pozvane u skriptu „game.js“ gdje su bile potrebne . Za prikazivanje/ skrivanje teksta koristila se mogućnost “.visible = true/false.” . Na slici 22 prikazan je način namještanja teksta na pozadini i dodavanje same boje slovima [5].

```
var FontSizeGameOver = 48;
var FontSizeScore = 24;
var FontName = 'Arial'
var FontSize = 24;
var ScoreX = 25;
var ScoreY = 25;
```

Slika 21. Namještanje varijabli u gamemanager.js koje poslije zovemo u game.js

```
this._gameOverLabel = this.addLabel("Game Over!",size.width / 2, size.height /1.7, FALSE, indexgoku,
cc.color.RED , FontSizeGameOver);
this._gameOverCredits = this.addLabel(":CREDITS:",size.width / 2, size.height /4.8*2, FALSE, indexgoku,
cc.color.BLACK);
this._gameOOverLabel = this.addLabel("Igra napravljena samo u svrhu završnog rada",size.width / 2, size.height /5.8*2, FALSE, indexgoku,
cc.color.BLACK );
this._gameOOverColl = this.addLabel("Fakultet Informatike u Puli",size.width / 2, size.height /6.8*2, FALSE, indexgoku,
cc.color.BLACK );
this._gameOverLabell = this.addLabel("Mentor : doc. dr. sc. Tihomir Orehovački",size.width / 2, size.height /7.8*2, FALSE, indexgoku,
cc.color.BLACK );
this._gameOverLlabell = this.addLabel("Student: Tomislav Ivanda",size.width / 2, size.height /8.8*2, FALSE, indexgoku,
cc.color.BLACK );
this._gameStartLabel = this.addLabel("Click for Start!",size.width / 2, size.height / 3* 2, TRUE, indexgoku,
cc.color.YELLOW , FontSizeGameOver);
this._scoreLabel = this.addLabel("000",ScoreX, size.height - ScoreY, TRUE, indexgoku,
cc.color.WHITE , FontSizeScore);
this._topScoreLabel = this.addLabel("100",ScoreX, size.height - ScoreY * 3, TRUE, indexgoku,
cc.color.WHITE , FontSizeScore);
```

Slika 22. Postavljanje teksta za prikaz u igrici

4. Zaključak

Današnjica u kojoj živimo se jako brzo razvija kao i sva tehnologija oko nas. Ljudi ne mogu više zamisliti odlazak na kavu ,a da ne koriste svoj mobilni telefon . Možemo reći da je doba kompjuterskih igrica prošlo, ali ne i web igrica . Većina mladih ljudi više nema naviku skidati nove igrice koje izađu na Internet , nego svoje vrijeme bolje vole provesti igrajući igrice koje im se nude na internetu i oko kojih nema posla s kompliciranim instalacijama .

Tako sam i ja kao malo dijete na nastavi informatike jedva čekao zadnjih pola sata koje bi nam profesor ostavio za uživanje u igrama i uvijek sam se pitao tko radi ovakve igrice i kakvi to mozgovi moraju biti . Sada kada sam završio ovaj rad i zavirio u svijet programiranja igara samo želim nastaviti i upoznavati nova razvojna okruženja .

Moj izbor za pravljenje ovako lagane stare igrice je bio taj što kada je originalna verzija za mobitel došla nakon samo par tjedana više nije bila dostupna na internetu i nisam ju mogao skinut, zato sam nekako uvijek čekao da netko postavi neku sličnu njoj na internet.

Kao dobru korist ovakvih tema i radova istaknuo bih stalnu primoranost na traženje rješenja i analiziranje stvari koje se nalaze na internetu a u jednu ruku mogu Vam pomoć pri kompletiranju programa i otklanjanje poteškoća sa kojima se suočite. Iz vlastitog primjera mogu reći da mi Cocos2d u početku nije bio previše poznat i imao sam poteškoće kako krenuti , ali sam tražeći na internetu i pitajući ljude koji se time bave polako slagao vlastiti projekt koji je napokon dovršen . I na posljetku samo znanje koje se dobije istraživanjem i vlastitim stvaranjem je ne zamjenjivo , te vam može pomoći u budućnosti pri daljnjem studiranju a ili na posljetku poslu koji budete radili .

5. Literatura

[1]. https://discuss.cocos2d-x.org, 12.09.2019.

[2]. https://github.com/cocos2d/cocos2d-js 12.09.2019.

[3].https://www.gamefromscratch.com/post/2014/09/29/Cocos2D-x-Tutorial-Series-Installation-Creating-a-Project-and-Hello-World.aspx 12.09.2019.

[4]. https://github.com/SonarSystems/Cocos-Helper, 12.09.2019.

[5].https://www.gamefromscratch.com/post/2012/06/03/Cocos2D-HTML-release-a-tech-demo-Moon-Warriors.aspx, 12.09.2019.

.

6. Popis slika

Slika 1. Početni zalon sa uputama……9

Slika 2. Početak igrice 10

Slika 3. Igrica u tijeku ..... **Error! Bookmark not defined.**

Slika 4. Kraj igrice i prikaz credita..... **Error! Bookmark not defined.**

Slika 5. Mogučnost skaliranja 11

Slika 6. Game.js kod za dodavanje pozadine 11

Slika 7. Gamemanager.js kod za prikaz puta 12

TOMISLAV IVANDA 1

RAZVOJ RAČUNALNE IGRE U OKRUŽENJU COCOS2D-X 1

Završni rad 1

Pula, rujan, 2019. 1

TOMISLAV IVANDA 2

RAZVOJ RAČUNALNE IGRE U OKRUŽENJU COCOS2D-X 2

Završni rad 2

Predmet: Programsko inženjerstvo 2

Znanstveno područje: Društvene znanosti 2

Znanstveno polje: Informacijske i komunikacijske znanosti 2

Znanstvena grana: Informacijski sustavi i informatologija 2

Pula, rujan, 2019. 2

1.Uvod 7

2. Prezentacija igrice 8

2.1 Početni zaslon sa uputama 8

2.2 Početak igrice 9

2.3 Igrica u tijeku 10

2.4 Kraj igrice 11

3. Prikaz stvaranja igrice 12

3.1 Dodavanje i namještanje pozadine 12

3.2 Detekcija dodira na ekranu 14

3.3 Dodavanje lika 16

3.4 Dodavanje funkcije skakanja ... 18
3.5 Dodavanje kretnji sporednih elemenata ... 19
3.6 Dodavanje cijevi i njihov načina rada ... 21
3.7 Stvaranje cijevi na ekranu ... 23
3.8 Dodavanje kolizije ... 24
3.9 Dodavanje teksta ... 25
4. Zaključak ... 26
5. Literatura ... 27
6. Popis slika ... 28

TOMISLAV IVANDA ... 1
RAZVOJ RAČUNALNE IGRE U OKRUŽENJU COCOS2D-X ... 1
Završni rad ... 1
Pula, rujan, 2019. ... 1
TOMISLAV IVANDA ... 2
RAZVOJ RAČUNALNE IGRE U OKRUŽENJU COCOS2D-X ... 2
Završni rad ... 2
Predmet: Programsko inženjerstvo ... 2
Znanstveno područje: Društvene znanosti ... 2
Znanstveno polje: Informacijske i komunikacijske znanosti ... 2
Znanstvena grana: Informacijski sustavi i informatologija ... 2
Pula, rujan, 2019. ... 2
1.Uvod ... 7
2. Prezentacija igrice ... 8
2.1 Početni zaslon sa uputama ... 8
2.2 Početak igrice ... 9
2.3 Igrica u tijeku ... 10
2.4 Kraj igrice ... 11
3. Prikaz stvaranja igrice ... 12
3.1 Dodavanje i namještanje pozadine ... 12
3.2 Detekcija dodira na ekranu ... 14
3.3 Dodavanje lika ... 16
3.4 Dodavanje funkcije skakanja ... 18
3.5 Dodavanje kretnji sporednih elemenata ... 19
3.6 Dodavanje cijevi i njihov načina rada ... 21
3.7 Stvaranje cijevi na ekranu ... 23

3.8 Dodavanje kolizije 24
3.9 Dodavanje teksta 25
4. Zaključak 26
5. Literatura 27
6. Popis slika 28