

Projektna naloga
Head Rush

Urban Burja

25.marec.2018

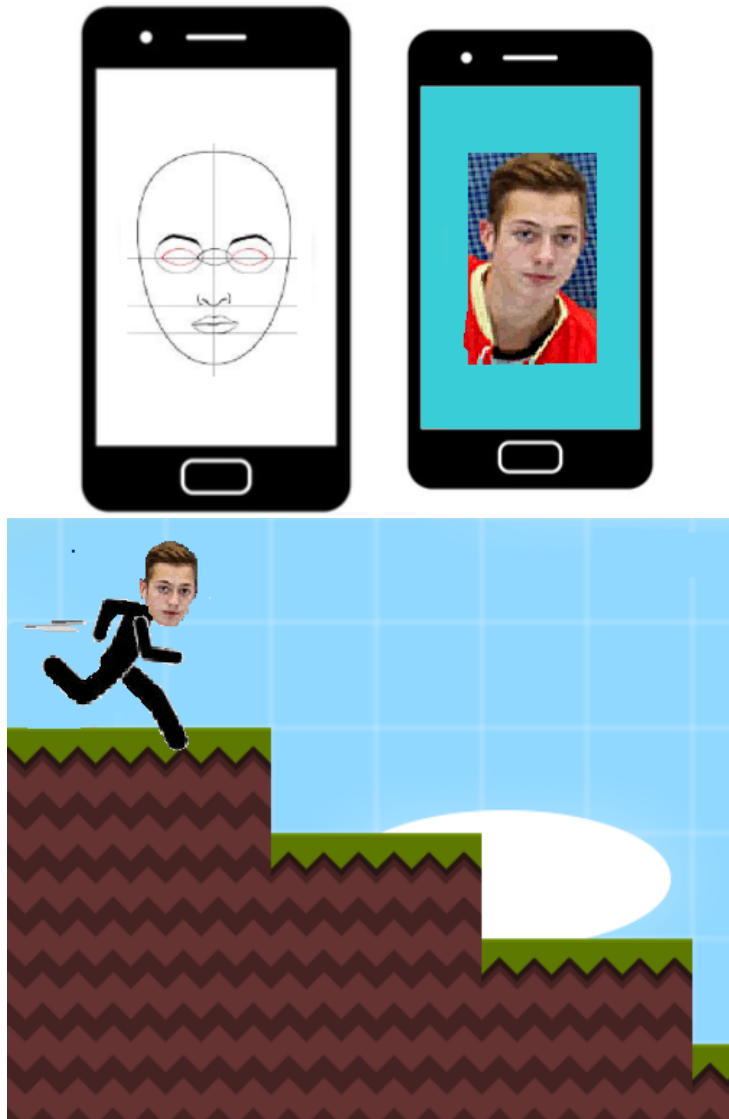
Contents

1	Žaner igre	3
1.1	Opis	3
2	Scenarij igre	4
2.1	Računalniški vid	4
2.2	Logične strukture in sistemi	4
2.3	Namenska programska oprema	4
3	Cilji	5
3.1	Character	5
3.1.1	Slikanje lastnega obraza	5
3.1.2	Deafult carecter	5
3.2	Ovire	5
3.3	Interakcije	6
3.4	Premikanje	6
3.4.1	Kompleksno premikanje	6
3.4.2	Premikanje z eno tipko	7
3.5	High Score	7
4	Razvojno okolje	8
4.1	Unity	8

1 Žaner igre

1.1 Opis

Head Rush je 2D platformer oziroma infinite runner, katere cilj je priti čim dlje in zbrati čim več točk. Inspiracija za igro Googlova T-Rex game. Igra se bo izvajala vodoravno na telefonu. Ideja igre:



2 Scenarij igre

2.1 Računalniški vid

Najbolj očitna vključitev v igro je računalniški vid, ki bo omogočal, da si igralec lahko slika obraz katerega bo imel carecter. S pomočjo Emgu knjižnic katere so podprte v Unity engine. S pomočjo Emgu knjižnice bo potrebno prepoznati obraz, ga zajeti in uporabiti le potreben del.

2.2 Logične strukture in sistemi

S predmetom Logične strukture in sistemi se bo v igro vključilo accelerometer s katerim bo možno nadzorovati skakanje in usmerjanje glavnega junaka.

2.3 Namenska programska oprema

? Kanal med telefonom in računalnikom primerjava High scora z ostalimi igralci.

3 Cilji

Cilj igre je priti čim dlje in nabrati čim več točk, medtem ko se mora igralec izogibati različnim oviram. Igralec prav tako ne sme biti prepočasen saj ga lovi poplava. Igralec ima le eno življenje zato mora biti previden.

3.1 Character

Glavni junak je igralec, ki si pred začetkom igre ustvari izgled svojega characterja to stori na dva načina.

- Slikanje lastnega obraza
- Default carecter

Možno je tudi spreminjanje barve characterja.

3.1.1 Slikanje lastnega obraza

Igralec si s pomočjo sprednje kamere na telefonu slika obraz. Carecter bo imel igralčev obraz za svojo glavo. Slika ostane shranjena in igralcu ni potrebno ob vsakem pričetku igre ponovno slikati obraza.

3.1.2 Default carecter

Carecter je že vnaprej izdelan in igralcu ni potrebno slikati obraza.

3.2 Ovire

Ovire se bodo po nekem postopku prikazovale po igralni površini.

- Zidovi
- Fall pit
- Poplava
- Sovražniki
- Glue
- Coins
- Hitrost
- SpeedBust

Igralec se mora izogibati različnim oviram, ki mu onemogočajo napredovanje kot so zidovi v katere se igralec samo zaleti ne da umre. Prav tako so tudi fall pits v katere lahko igralec pade in izgubi igro.

Igralec beži pred poplavo in v primeru da se prevečkrat zaleti v steno ga poplava dohiti.

Prav tako bodo igralcu oteževali delo sovražniki, ki se bodo premikali po igralni površini. V primeru da se igralec dotakne sovražnika je igre konec.

Glue bo upočasnjevalo igralca. Igralec prav tako lahko pobira Coinse, ki se bodo prikazovali po igralni površini. Ko ga igralec pobere se mu s tem poveča tudi score.

Hitrost premikanja glavnega junaka se bo postopoma povečevala in s tem otežila igro.

Igralec lahko pobere tudi SpeedBust, kar ponovno poveča njegovo hitrost.

3.3 Interakcije

Zidovi-igralec: če se igralec zaleti v steno ne bo umrl takoj, ga pa to močno zaustavi.

Fall pit-igralec: če igralec pade v luknjo takoj umre. Game over, njegove točke pa se shranijo v high score.

Poplava-igralec: v primeru, da igralca dohiti poplava le ta umre. Game over, točke se shranijo v high score.

Sovražnik-igralec: če se igralec dotakne sovražnika, igralec takoj umre. Game over, točke se shranijo v high score.

Glue-igralec: v primeru da igralec skoči na glue, ga ta močno upočasni.

Coins-igralec: igralec lahko pobere coin, kar mu poveča končni rezultat.

SpeedBust-igralec: v primeru, da igralec pobere SpeedBust se njegova hitrost poveča.

Hitrost-igralec: hitrost se se postopoma povečuje in se tem oteži igro

3.4 Premikanje

Za premikanje igralca sta dve možnosti:

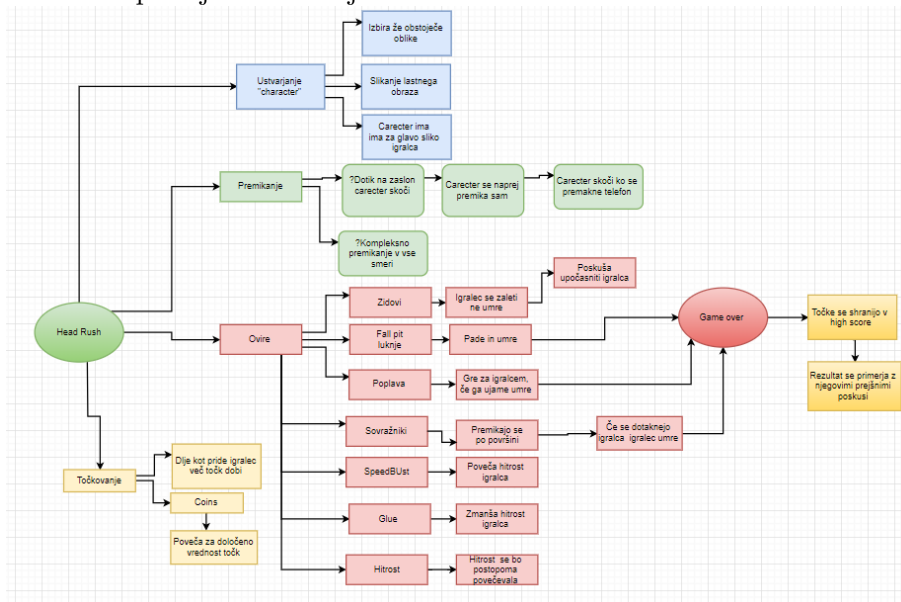
- Kompleksno premikanje
- Premikanje z eno tipko

3.4.1 Kompleksno premikanje

Igralec se lahko premika v 4 smeri plus tipka za skok. Za skok se lahko uporabi tudi accelerometer s katerim bo ob premiku telefona carecter skočil.

3.4.2 Premikanje z eno tipko

Character se naprej pomika sam, igralec pa ob pritisku na zaslon ali s pomočjo accelerometer upravlja le skakanje.



3.5 High Score

Po končani igri se v tabelo zabeleži dosežene točke, ki se primerjajo s preostalimi poskusi.

4 Razvojno okolje

4.1 Unity



Za izdelavo bom uporabil Unity engine, saj je eno najbolj primernih okolij za razvijanje iger na telefonu. Prav tako podpira tudi Emgu razred za delo s slikami. Unity je prav tako zelo prijazen za nove uporabnike, saj ponuja veliko pomoči, ki se jo najde na internetu.

Znotraj Unitya se uporablja C#.