Aghanatiq – The Game

Xırdalana bir nəfər.

Technical Document   
(Homework No.3)

Project team: Survivors

Instructor: Dr. Araz Yusubov

Submitted in partial fulfillment of the requirements of the CSCI 4836: Game Development Fundamentals course project

|  |  |
| --- | --- |
| Version date | Version information |
| 10.30.2018 | Initial draft |
| 11.3.2018 | <Version description> |

|  |  |
| --- | --- |
| Other documents in the package | |
| File name | Brief description of the document |
| <File name> | <Description of the document> |
| <File name> | <Description of the document> |

|  |  |  |
| --- | --- | --- |
| Team member | Contribution to this homework (NOT the project) | Estimated % |
| Nijat Mursali | Wrote the 4th part of the report | 25% |
| Nuru Nurdil | Wrote the 1st part of the report | 25% |
| <Student Name 3> |  |  |
| <Student Name 4> |  |  |

# Table of Contents

# Introduction

This is part of the Game Design Document for a hypothetical project “Aghanatiq – The Game” submitted for partial fulfillment of the requirements of the Game Development Fundamentals course in the School of Information Technologies and Engineering at ADA University, Baku, Azerbaijan.

The following file contains the necessary system requirements for the game, the required resources for the development of the game, the planned audio and visual contents; and the structure that the code would fit in. The game is self-contained and doesn’t require external libraries and applications. Overall, a clear description of the technical prototype for the game can be understood from this file.

**Bonus:** Get extra **5%** (x2) points for revising and SUBSTANTIALLY improving each of the Homework 1/2 content.

## System Requirements

The system requirements for the games include the following:

* Operating System

OS for our game will include Windows, Android and IOS because these systems are the most famous ones in today’s world.

* Processor Speed

For the Windows operating system the requirement for processor will be Intel Core I3. For the Android the requirement is Android 5.0+. For the IOS operating system the requirement will be IOS 7.0+.

* Memory

Our game doesn’t need much memory to run it, so the memory (RAM) will be minimum 1GB RAM for the Windows. For the Android OS the requirement for memory is 1GB RAM, so it is same with IOS.

* Graphics card

Any graphics card will support our game because our game doesn’t take much memory on the system.

* Hard disk space

It will not take much space on the OS, so the maximum size will be 100MBs.

## Concerns and Alternatives

None

## Resources

The following tools are used during the development of the game:

* Unity – for building the game
* Photoshop – for editing pictures that are inserted into the game in Unity
* Other open source images or code

# Visual Content

This part of technical document is about the visual content of “Aganatiq” game and it has some sections: General, player elements, heads up display (HuD), antagonistic elements and global elements.

* In ‘General’ section of visual content, information about size, format, quality and scale of file is given.
  + About File Size Restrictions of “Aganatiq” game, it can be said that each file size is maximum 5 MB and maximum size of game should be 100MB. This file size restriction is very efficient size especially in phone mode for this game.
  + As file format type, png and jpg are used in this game.
  + Additionally, there is File Quality Type which depends on the type of device: PC or phone. In “Aganatiq” game, as a quality, we used very low, low, medium, high, very high and ultra-levels, while we use only very low level for phone mode.
  + Visual Scale is also appropriate to the scales of PC and phone in the game.
* ‘Player Elements’ section
  + There are different types of states in each game and in “Aganatiq” game you can observe default, damage and destroyed states. In ‘default’ state, it can be seen that our main object taxi which is driven by Aganatiq move horizontally, in X axis. In ‘damage’ state, life time of main car of “Aganatiq” decreases in health bar due to the crash or different damages. In ‘destroyed’ state, game finish for some misfortunes and while game finishes new screen comes out which includes ‘try again’ (restart button) and ‘end game’ buttons.
  + As Animation Frame in player elements, movement of taxi can be demonstrated.
* Heads Up Display (HUD) represents the info which appears on the screen while the game is played by user/player. In “AGANATIQ” game, heads up display (HuD) includes following sections:
  + Type Icons of this game are health bar, score bar, timer, backward and forward buttons.
  + States of this game are splash page which shows logo “Aganatiq” in initial page, main menu for directory to main game and game play screen.
  + Font Type of elements in this game is Arial font.
* ‘Antagonistic Elements’ section
  + Type of States in term of Antagonistic elements are following ones: In ‘default’ state, collision with police, mayor and call to Aganatiq can be indicated. Problems in this state can be solved with two options relatively: going to parking lots and paying money. Moreover, in damage state, collision with wrecking ball can be indicated. This ball can damage player (taxi which is driven by ‘Aganatiq’) in every hit.
  + Animation Frames in ‘antagonistic elements’ section are other cars which causes sometimes overlap with main character “Aganatiq” ’s car.
* Global Elements
  + In “Aganatiq” game, background as a part of global elements can include background picture which is city view, main buildings of Baku, street lights, objects such as stars, rain, dust and so on. Additionally, texture of the game is 2D city view and tile is ground.
  + As mentioned before, Font Type of this game is Arial font.

# Audio Content

<This is the section for organizing the audio content. It is very important to communicate with the audio designer before and while the audio content is being developed.

* General
  + File Size Restrictions
  + File Format Type
  + File Quality Type
* Player Elements
  + Type of Sound f/x
  + Device Vibration
* Antagonistic Elements
  + Type of Sound f/x
  + Device Vibration
* Global Elements
  + Ambient Music
* Splash Screens
  + Ambient Music
* Menus
  + Type of Sound f/x

>

# Programming Content

The objective of this section is to try to organize and modulate how the game should be.

* General
  + Requirements

The requirements for the games include the following:

* Operating System

OS for our game will include Windows, Android and IOS because these systems are the most famous ones in today’s world.

* Processor Speed

For the Windows operating system the requirement for processor will be Intel Core I3. For the Android the requirement is Android 5.0+. For the IOS operating system the requirement will be IOS 7.0+.

* Memory

Our game doesn’t need much memory to run it, so the memory (RAM) will be minimum 1GB RAM for the Windows. For the Android OS the requirement for memory is 1GB RAM, so it is same with IOS.

* Graphics card

Any graphics card will support our game because our game doesn’t take much memory on the system.

* Hard disk space

It will not take much space on the OS, so the maximum size will be 100MBs.

* + File Size Restrictions

Our game doesn’t require high storage, so for the final version of the game will be maximum 100MB which means the maximum size of one image is 5MB (if we will have 20 elements in our game).

* + File Format Type

The file format size will include PNG, MP3, C# and other ones.

* + Specify Coding Conventions

If we specify the coding conventions

* + Language/Device Restrictions

The main language will be English language, but we will also add other languages like Azerbaijani and Russian too. The idea is that we will add buttons for changing the languages in the game and if player chooses, for example English, the scene will change into scene that is in English.

* + Screen Type (Small, Medium, Large)

We will create our game for both PCs and laptops which make the game more powerful in a sense that more player will have opportunity to play and have it. For the PCs the screen type will be large; however, the player will be able to choose different resolutions for running the game. However, for the mobile phones we will modify the resolution that most of the phone can open the game in small screen.

* Player Elements
  + Go to the destination on time

We will have lots of elements that player will include in himself. The first element is that on screen (phone or PC) the player will be able to press the “accelerate” and “brake” buttons while driving the car. We will also add other elements like sirens and lights of the car in order to make it look much better and make player enjoy the gameplay.

* Antagonistic Elements
  + Slow down the player

First antagonistic element in our games will be the obstacles like ball, that are connected to one point and try to hit our player in order to make difficulties for him. This is made by using Joint 2D in Unity 5. When the collision happens between car and ball there will be the function which is OnCollisionEnter2D that will detect the collision and will decrease the health of player. We will add several AI elements like this in order to make the game more beautiful and harder.

Our second antagonistic element will be the police officer named “Sergeant Balayev” who takes care of the roads of Baku. If the player will go beyond the speed limit, our police officer will stop our player. We will also use OnCollosionEnter2D method in order to detect the collision.

* Global Elements
  + Sprites and lights

Global elements will include background sky sprite, street lights, buildings and other objects which doesn’t affect to the player and antagonistic elements. Global elements will be mainly used to make the game look better in a way with graphics, lightnings and other types of elements.

* Splash Screens
  + Fade In, Fade Out

The first splash screen will be Unity manual one which shows the logo of Unity. However, we will also add our custom splash screen which will come after Unity’s one. We will do it by adding new scene which will have “Loading” component and after some seconds the scene will change into MainMenu scene.

* Menus

In our game we will have several menus in the game. The first one will include splash screen which will show the logo of our game, after that the main menu screen will show up. In MainMenu screen the player will be able to choose what options he/she wants. According to the choice of player the new scene will pop up which will be the scene that player will be able to play game itself. For every level we will add new scenes into the game.

* + Load Level
  + Play Game, Options, Quit Game

## Code Structure

These section gives information about the functions we used in our game.

**OnCollisionEnter2D** checks if the collision happened or not. If the collision happened the car will get damaged by several objects like ball and other obstacles. We also used this function for our AI which is police officer and if player goes near the police the collision will happen and there will be options that pop-up.

**LateUpdate** function will be used for following the player and will be attached to the main camera.

**Click** function will check if the button is pressed or not. If the button is pressed on the screen/keyboard, the actions like siren sound and lights will be on.

**ClickRestart** function will be used if the player’s health is zero and he wants to restart the game.

**ClickQuit** function will be used if the player wants to quit the game.

**Bonus:** Get extra **10%** points for including Interaction Matrices[[1]](#footnote-2) and/or Class Diagrams.>

Game

-Sprite

+AudioSource()

Main Menu

-Sprite

+PlayGame()

+Options()

+Quit()

Splash Menu

-Sprite

+FadeIn()

+FadeOut()

GlobalElements

-Sprite

-Lights

+AudioSource()

AI

-Sprite

+OnCollisionEnter()

Player

-Sprite

+Lights()

+Move()

+Siren()

StreetLights

-Collider

+OnCollisionEnter()

Buildings

Ball

-Physics

-Collider

-OnCollisionEnter()

# References

The following game did not require drastic use of references to build up a clear idea of the game design and dynamics. Therefore, the following two references were enough to conclude a general idea on the product.

Reference 1: Hill Climb Racing, *by Fingersoft,* <https://play.google.com/store/apps/details?id=com.fingersoft.hillclimb&hl=en>

Reference 2: Taxi Sim 2016, *by Ovidiu Pop,*

<https://play.google.com/store/apps/details?id=com.ovilex.taxisim2016>

1. Interaction matrix is a spreadsheet listing game objects on sides, and interactions that can occur between them during the game at intersections of rows and columns. [↑](#footnote-ref-2)