<Great Instructor Races>

“You will RACE but will you ACE???” © (N)\*everLasting

Design Document   
(Homework No.2)

Project team: (N)\*everLasting

Instructor: Dr. Araz Yusubov

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

|  |  |
| --- | --- |
| Version date | Version information |
|  | Initial draft |
| <Date> | <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 % |
| Ilyas Karimov |  | 25% |
| Sevil Jafarova |  | 25% |
| Ismayil Bagirov |  | 25% |
| Aytan Alakbarova |  | 25% |

# Table of Contents

<Automatically generate here using Microsoft® Word menu References→Table of Contents>

<This document describes how game objects behave, controlled and properties they have. This is often referred to as the “mechanics” of the game. This documentation is primarily concerned with the game itself. This part of the document is meant to be modular, meaning that you could have several different Design documents attached to the Concept Document.>

# Introduction

Menu:

When the game is launched the player will see the following buttons

* **Start New Game** (with options of easy, medium and hard levels)
* **Top 10 Board**

Synopsis:

One of the Professors of ADA (Araz Yusubov,Emin Alasgarov, Gulmammad Gulmammadov,Samir Rustamov) race with other cars in the roads of ADA University to stay alive until time of the game elapses.

Game Play:

In GIR one car is chosen (indecisive whether there will be other car options) and races with other cars to have a high score in a small amount of time. There will be related boosters which are explained right below if they are chosen:

**Gulmammad Gulmammadov**

skill:

· Using gravity which attacks other cars to the ground to limit their movement of the cars.

**Emin Alasgarov**

skill:

· Encrypts the car, making it untouchable which frees him from obstacles on the road.

**Samir Rustamov**

skill:

· Uses car speed booster to overtake the opponents’ cars and adds x10 seconds to time.

**Araz Yusubov**

skill:

· Turns the car into a bus no 77 and adds x2 Score.

These all features will last maximumly 10 seconds.

The game begins and the player races with other cars, the other cars are managed by the computer. The player’s aim is to get a high score

· **Objectives**: get a high score within limited time.

· **Outcomes:** getting a high score and seeing your score in Top10 board.

· **Uncertainty**: collecting bonus boosters, random elements which have different features.

· **Rules and Structure**: skills of the personages (in the boosters) will be applied as stated above, overcoming obstacles along the road.

**Outline:**

**Game Options - choosing many boosters of the 4 personages on the road and the level of the game.**

**Game Elements - personages, cars, obstacles, bonus boosters, ADA University.**

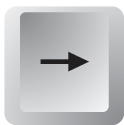
**Game Levels – easy, medium, hard levels in the game.**

*Why is all this fun?*

Most of the multiplayer racing games have the problem of **player balancing** where skilled players get bored and weak players get frustrated (Cechanowicz et al., see References [2]). To solve this problem, we add challenges with random probabilities of appearance during the game. Moreover, the game has unique storyline where personages represent professors of the university and the dean (Dr. Araz Yusubov) waits them on the finish line motivating them for new challenges.

Player Control:

Players can press ← and → keys to move to the right and to the left

Game Over (Winning & Losing):

The player wins if it does not collide with opponents’ cars and gets a particular score. If the player collides with an opponent's’ car then, it loses the game.

## Game matrix

|  |  |
| --- | --- |
| **Player/Antagonistic Element** | **Game Property** |
| Player | Car that races with opponents during the game |
| Opponent car X | Antagonistic Element   * If collides with player’s car destroys it |
| Stone | Antagonistic Element   * If player’s car collides with stone, the score of the player decreases |

## Game Flow Chart

State

## 

## Determined action

Action

Input

## Optional action

Decision

Play the game’s main scene

Start

Get input from the user (can be 🡨 or 🡪 keyboard)

Game Over

Has the player collided with opponent’s car?

Is time elapsed?

YES

Collect boosters

NO NO

Collision with stones

# Player Elements

<The player elements section lists all the elements that are directly related to the player or serve to benefit of the player. Devise two sets of names for player elements. One set is a generic name (or code) and the other is its game name. Describe the terminology that you use to describe the player’s properties.

This is a good place to interact with a graphic designer to ensure the game graphics match the game names. Graphics that will be seen during game play should be exhibited here. Multi-player issues should also be mentioned here.>

## Player Definition

<Use the player definition section to make quick descriptions that define the player.

Below is a suggested list of player definitions:

* Default (Status): Player is given a car with specifications (color,size and etc) which cannot be changed during the game. Car moves at a constant speed that differs from the hardness of the game chosen: easy, medium and hard and the speed of the game is slow, medium and fast respectively.
* Actions: Player can go to the left and right positions through the columns. One time pressing on the left/right keys should position the car to the nearby left/right columns.
* Information (Status): What information about the game is available for the player?
* Default Properties: Player begins the game with the score of 0 and time of 3:00(three minutes)
* Winning: The player doesn’t have any winning actions, however, if the player can survive and get the highest score, this might me considd
* Losing: The player loses the game if the player’s car collides with any other cars on the lanes. The player has also the opportunity of loss if the given time ends.>

## Player Properties

<Make a list within the player properties section that defines the properties for each player. Player properties can be affected by player’s action or interaction with other game elements. Define the properties and how they affect the player’s current game. A suggested list of player definitions may include:

* Health
* Weapons
* Actions
* Etc.

Each property should mention a feedback as a result of the property changing.>

## Player Rewards (Power-ups & Pick-ups)

<Within the player rewards section, make a list of all objects that affect the player in a positive way (i.e. health replenished). Define these objects by describing what affect they cause and how the player can use the object.>

# User Interface (UI)

< This is where a description of the user’s control of the game can be placed. It is also recommended to think about which buttons on a device would be best suited for the game. Consider what the worst layout is, then ask yourself if your UI is it still playable?

A visual representation can be added, where we relate the physical controls to the actions in the game. When designing the UI, it may be valuable to research quality control and user interface (UI) design information.>

## Heads up Display (HUD)

<The HUD section is where a description of any graphics that will represent information during game play should be described. A visual representation (mock-up screenshot) here would be useful. This is another good place to seek the advice or collaboration of a graphic designer.>

## Player View

<A screen shot is very necessary in the player view section. It is also beneficial to include a definition of how the camera moves for the player. Finally, a (mock-up) overview of the level relative to the screen size will help create a perspective of a levels size compared to what is actually seen.

# Antagonistic Elements

<This is where a list of antagonistic (i.e. enemies, opponent) objects should be listed with graphics and written description. Describe the terminology that you used to describe antagonistic properties. Devise two sets of names for player elements. One set is a generic name (or code) and the other is its game name. This is another good place to collaborate with a graphic designer to ensure the game graphics match the game titles, names, and descriptors. >

## Antagonistic Definitions

<This where a description goes of what makes an antagonistic element.>

## Antagonistic Properties

<This is a list of properties that antagonistic elements have in common.

## Antagonistic List

<This is where a list of all the antagonistic elements goes.>

## Artificial Intelligence (AI)

<This is where visuals and written description(s) of the antagonistic element’s behaviors. These should be labeled in such a way that they can be used in level design without having to describe them again. Devise generic names for repetitive behaviors. This is how an AI action could be deconstructed:

* Normal State: What is the object doing if it has not come in contact with the player?
* Detection State: What does it take for this object to detect the player?
* Reaction State: What does the object do as an action after passing the reaction state?
* End State: What happens to the object after player has reacted correctly or incorrectly to object?

>

# Global Game Elements

<In this section, it is important to describe the boundaries, neutral objects, camera views and scale of the world. Neutral game world objects can be things like a static background, objects that do not interact with the player or antagonistic elements.>

## The Story

<This is where the story can be described in detail. A story board can be used to tie in graphics to the text. This can later be used for splash screen concepts.>

## The Story Copy

<A shorter version of the story (the in game version) should also be written here. This is where the script for in game characters or story information during the cut scenes would be placed. This category does not always pertain to the current Game Design.>

## Concept Art

<Sketches that are used for the concept can go into this section as visual reference. In the case of a brand, certain creative restrictions should be noted here. This is a good place to collaborate with a graphic designer to ensure game graphics match game names.>

## Level Design

<This is where information pertaining to level design and visuals of the level design goes. Level design can best be shown as a flow chart. Use generic names to create level design.>

## Level Copy

<This is where the script for in game characters or story information during the cut scenes would be placed.>

## Audio & Sound F/X

<This is where game audio and Sound F/X should be listed, first with generic names and then described. This section also includes deciding if you will use a device’s vibration ring mode.>

# Game Architecture

<The game architecture section is best produced using a flow chart to represent the overall game. Be sure to identify (i.e. name, number) each screen.

* Title Screen
* Option Screens
* Game Modes
* End Screens

>

## Game Architecture Overview

<The splash screens or video clips need to be in accordance to game story and style. If cut scenes use video then story boards should be created.

Menus should be designed with the most important options easily accessible. Be aware how many clicks it takes to accomplish a task. The game Instructions should be written so that the player understands how to play the game. Mock-ups should be made so that the game programmers get the correct layout of the menu. It is a good idea to mention and describe the high score screen in this section.>

## Architecture Copy

<All text from the game can be compiled here. Review the Game Architecture Overview section.>

## How to Play Copy

<This section will organize the game copy. The game copy includes information for the player, clearly describing how to play the game.>

# References

[2] J. C., C. G., & S. B. (2014, October19). Improving Player Balancing in Racing Games. Retrieved from <http://hciweb.usask.ca/uploads/347-p47-cechanowicz.pdf>