Kauhanen Jani, Koivisto Eemeli, Lauronen Simo, Merilehto Iita, Niskala Milla

Duck Swipe

Technical Design Document

7.9.2022

Author(s):

Kauhanen Jani

Koivisto Eemeli

Lauronen Simo

Merilehto Iita

Niskala Milla

Version:

Unity 2021.3.9f1

Visual Studio 2019/2022

**Table of contents**

[1 Document change history 1](#_Toc81469034)

[2 Game/Software overview 2](#_Toc81469035)

[2.1 Game/Software summary 2](#_Toc81469036)

[2.2 Technical goals 2](#_Toc81469037)

[2.3 Tools 2](#_Toc81469038)

[2.4 Coding styles 2](#_Toc81469039)

[3 Class diagrams 4](#_Toc81469040)

[3.1 Class 1 4](#_Toc81469041)

[3.2 Class 2 4](#_Toc81469042)

[4 Software behavior 5](#_Toc81469043)

[4.1 Game / Software loop 5](#_Toc81469044)

# Document change history

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Date | Version | Comments |
| Eemeli Koivisto, Jani Kauhanen | 2.9.2022 | 0.0.1 | Document created |
| Matti Meikäläinen | 10.6.2003 | 0.0.2 | Document updated |
| Matti Meikäläinen | 13.6.2003 | 0.1.0 | Added pictures to document |
| Matti Meikäläinen | 15.6.2003 | 1.0.0 | Last version |

Table 1. Document change history

# Game/Software overview

## Game/Software summary

Duck swipe is a musical battle game where you have to correctly draw glyphs to attack. The theme of the game is to be lighthearted and cartoony. Game features should feel fluid and main game mechanics should be easy to use. Optimization is an important part of this, and it should be one of the top priorities when designing and developing.

Features from GDD: (TODO: muuta GDC:n mukaan)

* Drawing
* Rating the players drawing
* Battle mechanics:
  + Health
  + Attack turns
  + matching the music tempo to game tempo
  + combo meter
  + number of battles?
* Score and player resources
* Power ups
* Shop system
* UI

More in depth explanations of mechanics here.

## Technical goals

* What is the target device, Watch, Phone, Tablet, Smart Fridge, TV, VR headset, GooglePlay Instant, Cars…
* Target devices are recent low-range and older mid-range Android phones.
* What are the minimum and target version API levels,
* Minimum and target API level is 31, and target android version is Android 12.
* Supported Aspect Ratios
* Game supports 16:9 aspect ratio.
* Supported Input Devices: keyboard, mouse, gamepad, touch screen
* Supported input scheme is Touch screen.
* Network protocols used if any, TCP, UDP, Bluetooth, Bluetooth Low Energy, NFC, Wifi
* TCP / UDP, min 3G connection.
* Services: Ads, Analytics, IAPs, Billing, Asset delivery, in-app reviews,
* Game will have in game ads and analytics. (How many daily active users/active users, what data is collected)
* Multiple APKs used [Multiple APK support  |  Android Developers](https://developer.android.com/google/play/publishing/multiple-apks)
* If we have the need for it.
* Android App Bundles [About Android App Bundles  |  Android Developers](https://developer.android.com/guide/app-bundle)
* If we have the need for it.
* Sensors used: Camera, Motion, Position
* vibration
* Target Store, features used
* Target store is Google Play.

## Tools

Unity 2021.3.9f1

Visual studio 2019/2022

(UI työkalu)

Git bash, github desktop

KamIT versionhallinta

## Coding styles

* new coding styles and techniques

Pascal case:

* Namespaces
* Class names
* public Static fields
* Public variables
* Properties
* Method names
* Structs

Camel case:

* parameters

Exeptions

* Interfaces using IPascalCase
* private static variables using s\_prefix and camelCase, ex. s\_staticVar.
* private variables using \_camelCase
* Events using PascalCase and On prefix, ex. OnCharacterDeath
* Delegates using pacalCase and Callback suffix, ex. ProcessActionCallback
* coding standards
* Static variables on top of the class
* Comments on separate lines
* Braces on their own line

Text

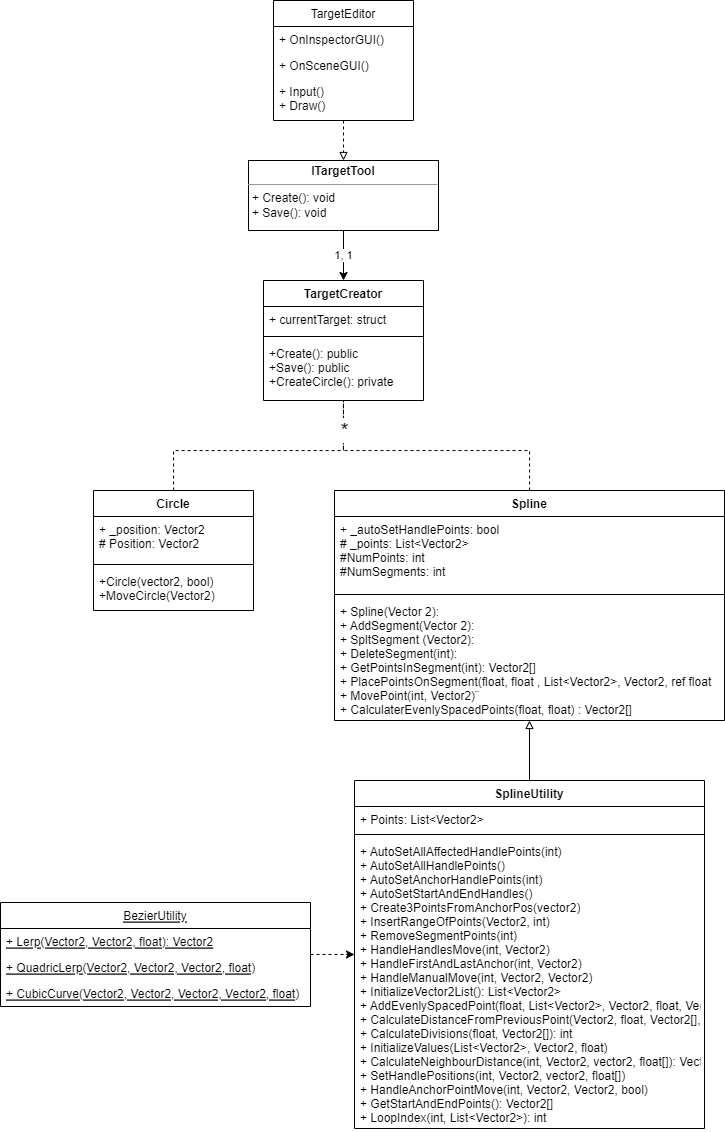
Description automatically generated

Figure Coding conventions example

# Class diagrams

Image (s) of UML diagrams. Write about the categories and how they work in the program

## Target creation



## Class 2

# Software behavior

Circles:

movement, Interactable,

Shapes:

input detection, pass/fail,

Health:

Turn handling:

UI:

Tutorial:

## Game / Software loop

* Add diagram of your game flow