Snake 3D

Technical Design Document

*Version 1.0*

*Omar Saeed*

*Website: secondwarrior.com*

*Mobile: +201008569433*

# Table of Contents

[Table of Contents](#_f3hbr6p0ak37)

[Design History](#_e5akww2oi216)

[Game Overview](#_j2kld1ana13b)

[Game Summary](#_i566fs1eqqtn)

[Target Platform](#_w0h9hbmhvcr)

[Development Overview](#_h7psffrwssbg)

[Development Team](#_xxk0bak9cyj9)

[Development Software](#_drxstdfky6pr)

[External Code](#_3wx81gnjfbw1)

[Game Mechanics](#_hm65yr3rs85w)

[Main Technical Requirements](#_c2kb94gs6u8i)

[Architecture](#_yxdrr8vse4bb)

[Game Flow](#_93tph0q2eust)

[Graphics](#_dal8viau5j83)

[Audio](#_oc7d279r9ymu)

[Physics](#_w9h7t0xxygzf)

[Game Objects and Logic](#_jm2i9mtxtpro)

[User Interface](#_rvrktgcok4ku)

[Main Menu](#_qc8tfbuzct1k)

[In Game HUD](#_9sbf9gvn1b4w)

# 

# 

# Design History

Version 1.0: First produced version of the game.

# 

# Game Overview

--------------------------------------------------------------------------------------------------------------------

# Game Summary

As a snake trying to grow as you can thought eating fruits, you are also trying to avoid hazards like hitting a wall, an obstacle, or eating a bomb, Developed upon the mechanics of the legacy snake mobile game.

## Target Platform

As a casual game, it mainly target android and ios devices, but it also will be exported to PCs.

# Development Overview

--------------------------------------------------------------------------------------------------------------------

# Development Team

[Omar Saeed](http://github.com/Omar47i/)

Game Design and implementation, UI Design, Animations, Sounds, AI, pretty everything,

# Development Software

Project is developed using Unity engine, Photoshop is used for UI image editing, Version control will be handled by [Git](http://git-scm.com/) with the repository hosted by [GitHub](http://github.com/).

# External Code

[iTween](https://www.assetstore.unity3d.com/en/#!/content/84) free unity plugin has been used to implement simple animations for some game elements e.g. fruit spawning, bomb spawning, and snake’s bite.

The game has been made from scratch, though i used 3D assets from my previous projects and other online resources only as a proof of concept so please don’t reuse in any form (free or commercial).

Object pooling is used from a previous project, a lot of examples and scripts is available online for object pooling.

# Game Mechanics

--------------------------------------------------------------------------------------------------------------------

# Main Technical Requirements

The main technical requirements of the game is the difficulty progression algorithm, Most other aspects of the game are not so intensive: growing a snake, randomize direction on start, pickups (fruit/bomb) spawning, Collision with borders - obstacles - and bombs, player score, player best score, saving and loading player’s data (score - best score), pooling frequently created and destructed elements (snak nodes, fruits), sounds management, managing animations for sliding in and out in title panel and HUD, animations using iTween engine, etc..

# Architecture

There are two main modules in the game. These are: Home module and gameplay.

The home module hosts the title panel animations, settings (sounds and music toggle), and play button.

The gameplay module hosts every thing related to the gameplay.

## 

# Game Flow

The game has only one flow which is the **Main Game**, But two levels are implemented with different art.

When game starts the title panel animation slides in, displaying the game title and snake’s icon, with the ability to click/tap on the snake’s icon to start the game.

When you start the game the title panel animation slides out and the HUD slides in.

HUD contain information like current score, best score, pause button, sounds toggle option, music toggle option, and snake icon.

You start with the snake containing only three nodes: head, middle node, and tail.

The start direction is randomized then snake reposition its nodes based on the new direction, A new fruit is spawned at random direction inside the game playground area with respect to snake and obstacles position.

Snake movement is handled using keyboard arrows and mobile swipe for the four directions, when the snake eats a fruit it grows one extra node at the tail and a new fruit is spawned in a new position.

After two fruits spawning, the DifficultyProgression.cs will start to generate the bombs based on a random time, the bomb will stay alive for a specified random time then it depletes.

Colliding with a bomb or a wall will result in losing the game, when you loses a comparison between your current score and your last best score is held to register the best score in PlayerPrefs.

When you loses game resets without reloading as all the variable elements listen to a Reset event to reset its values and then user go back to the title menu in a reverse animation.

# Graphics

As the game targets mobile device, All 3d models are picked with respect to poly count, most of the shaders are mobile optimized shaders though standard unity shaders are avoided due to their expensive cost on mobile GPUs, all game textures are compressed to reduce final build size and memory usage especially on mobile devices, reference resolution for the game is 16/9 though the game can be played on 4/3 devices.

# Audio

Game has main menu and gameplay music, music stops only when player loses, though gameplay module has sound effects.

# Physics

Game uses kinematic rigid bodies to simulate physics collisions like snake/bomb collision, snake/wall collision, snake/self collision, snake/fruit collision, and snake/obstacle collision.

# Game Objects and Logic

**Snake**

Parent game object of the snake’s head and the rest of the snake’s body, this object is responsible for creating extra nodes when colliding with fruits, though the collision system is decentralized as every object colliding object with the snake informs the snake about the collision not the other way as this is cleaner and organized, the rest of the snake’s body checks for collision only with the head, Snake gameObject also responsible for clearing snake’s nodes upon reset, snake can eat a fruit to grow, and can die by colliding with walls, obstacles, and bombs.

**Fruit**

Created at random locations and can be eaten by the snake, it gorws the snake length by one node.

**Bomb**

Created at random locations and can be eaten by the snake, results in player’s losing the game.

**Border**

Defining the game playground area, colliding with them will result in player losing the game.

**Obstacle**

Placed in game playground area, colliding with them will result in player lossing the game.

**Hit VFX**

A special effect that is created when colliding with game borders, obstacles, and self-collision.

**Scoring Text**

As a rewarding effect the scoring text is created after eating a fruit, a special scoring text is created when beating the last best score.

# User Interface

--------------------------------------------------------------------------------------------------------------------

# Main Menu

The main menu contains the title of the game associated with a snake icon as a button to be clicked/tapped for entering the gameplay module.

# In Game HUD

The game HUD contains:

* Current score text.
* Best score text.
* Snake icon.
* Pause button.
* Sound effects toggle.
* Music toggle.