

Long Nok, Chan

c010313m@student.staffs.ac.uk

**[Momo Life Simulator]**

Technical Design Document

Contents

[Project Introduction 3](#_Toc190760439)

[Project Goals 3](#_Toc190760440)

[Challenges and Risks 3](#_Toc190760441)

[Hardware Requirements 3](#_Toc190760442)

[Platforms 3](#_Toc190760443)

[Target Platform 3](#_Toc190760444)

[Engine Specific Specifications and Limitations 3](#_Toc190760445)

[Engine Summary 4](#_Toc190760446)

[Systems and Diagrams 4](#_Toc190760447)

[Game Loop 4](#_Toc190760448)

[Information Architecture 4](#_Toc190760449)

[Main Menu and Pause Panel 4](#_Toc190760450)

[Main level Game HUD 4](#_Toc190760451)

[Game Over Scene 4](#_Toc190760452)

[Wireframe 4](#_Toc190760453)

[Main Menu 4](#_Toc190760454)

[Loading Scene 4](#_Toc190760455)

[Game HUD 4](#_Toc190760456)

[Flowchart 4](#_Toc190760457)

[Main Menu 4](#_Toc190760458)

[Character Movement and Variable Update 4](#_Toc190760459)

[Button Interaction 4](#_Toc190760460)

[Game end 4](#_Toc190760461)

[Pause Panel 4](#_Toc190760462)

[Optimisation and Profiling 4](#_Toc190760463)

[Profiling Systems 4](#_Toc190760464)

[Profiling Graphics 4](#_Toc190760465)

[Profiling Network/Multiplayer (If Applicable) 4](#_Toc190760466)

[Coding Standards 4](#_Toc190760467)

[Programming Standards 4](#_Toc190760468)

[Style Guide 5](#_Toc190760469)

[Commenting Rules 5](#_Toc190760470)

[Production Overview 5](#_Toc190760471)

[Moscow 5](#_Toc190760472)

[Timeline 6](#_Toc190760473)

[Budgeting 6](#_Toc190760474)

# Project Introduction

This project is to create a prototype game in the game category of “Life Sim” that reaches a playable and polished demo quality. It requires to have:

* A built version of the prototype game could be played.
* A demo video to demonstrate prototype game features.
* An project version of the prototype

## Project Goals

The brief goal of this project is to create a Life Sim prototype with the following features:

* A controllable character that players can interact with
* The character contains a range of requirements and stats that could be controlled and responded to by players
* Stats diverse change as different commands execute to increase gameplay. (One action command could affect other stats at the same time)
* Set game time to create a game-loop to keep players entertained and explore interest

## Challenges and Risks

The challenge and risk of this project would:

* Be able to balance the stats change as players interact with the character. Not overpower or too weak.
* Stats change could reach different game endings at the end
* Set the approved game time for the current prototype game, too long would cause the player to lose interest and too short would cause the prototype to lack of challenge, which will affect gameplay.
* Simple and straight forward gameplay might cause players to easily lose interest

## Hardware Requirements

This prototype development would be using Unreal Engine game engine.

# Platforms

## Target Platform

This prototype platform would be playable on PC only. Currently, no further plan to launch on other platforms.

## Engine Specific Specifications and Limitations

For PC users, the recommended Hardware requirements would be:

|  |  |
| --- | --- |
| Operating System | Windows 10 64-bit version 1909 revision .1350 or higher, or versions 2004 and 20H2 revision .789 or higher |
| Processor | Quad-core Intel or AMD, 2.5 GHz or faster |
| Memory | 32 GB RAM |
| Graphics RAM | 8 GB or more |
| Graphics Card | DirectX 11 or 12 compatible graphics card with the latest drivers |

## Engine Summary

This project uses Unreal Engine 5.4 Version and no plugin is needed.

# Systems and Diagrams

## Game Loop

A diagram of a game development

AI-generated content may be incorrect.

## Information Architecture

### Main Menu and Pause Panel

A screenshot of a computer

AI-generated content may be incorrect.

### Main level Game HUD

A diagram of a company

AI-generated content may be incorrect.

### Game Over Scene

A screen shot of a computer

AI-generated content may be incorrect.

## Wireframe

### Main Menu

A screenshot of a computer

AI-generated content may be incorrect.

### Loading Scene

A screenshot of a computer

AI-generated content may be incorrect.

### Game HUD

A screenshot of a computer

AI-generated content may be incorrect.

## Flowchart

### Main Menu

* Main Menu is a new level in the prototype
* When player start game, it enter a loading scene with game control map
* Credit panel show and hidden as player trigger credit option

A diagram of a flowchart

AI-generated content may be incorrect.

### Character Movement and Variable Update

* Character movement relate to clicking on map level with navigation mesh volume
* Character stats update on HUD and get player notice when ever variable is on Max or below a set float.

A diagram of a character

AI-generated content may be incorrect. A diagram of a character

AI-generated content may be incorrect.

### Button Interaction

* The prototype contain 2 sets of buttons, one is controlling character’s needs like saturation, hygiene etc.
* Another one is gaining character skills like creativity, intelligence etc.
* Sound effect and animation feedback respond to players of button function firing (cooldown progress bar and sound effect) and completed (button animation and sound effect)

A diagram of a flowchart

AI-generated content may be incorrect. A diagram of a flowchart

AI-generated content may be incorrect.

### Game end

* The game ends 2 two scenarios, one reflects on the character’s needs variable, and another reflects on the character’s stats
* When the character’s needs don’t get fulfil, the player will enter bad endings
* When game time reaches day 3 plus the character fulfils the requirement stats variable, the player will enter good endings.

A diagram of a company

AI-generated content may be incorrect.

### Pause Panel

* Pause panel trigger with button in HUD or esc command

A diagram of a game

AI-generated content may be incorrect.

# Optimisation and Profiling

## Profiling Systems

To check and monitor prototype functionality, QA test would be set up in the middle of the project development to collect feedback on gameplay and features. That would be:

* Does the prototype run properly?
* Is there any bug?
* Character movement and command responsive?
* Was the gameplay interesting and kept you interested during the play?
* Ask for suggestions to improve gameplay and game feel.
* Rate the prototype and should this prototype should be kept in the development process.

Based on the collected responses, perform changes and implement suggested game features to improve gameplay, feel and quality.

## Profiling Graphics

Graphic performance monitor by checking prototype FPS during playtest and development, aim to keep the FPS rate minimum 50 and aim to reach between 100-120 in high performance.

## Profiling Network/Multiplayer (If Applicable)

This prototype is designed to be a one-player game, therefore there would be no testing or plan to monitor or maintain multiplayer/ network features.

# Coding Standards

## Programming Standards

Assets will need to be stored in related folders for developers to look up.

* Store all assets into one folder and separate icon, image and font
* All widgets need to be stored in one folder and separated in base widget and custom widget (progress bar, button, Game ending scene etc.)
* Inside regular blueprints/ blueprint components etc., variables will need to be categories to increase ease of reading (cause some blueprints would require a lot of reference variable)

|  |  |
| --- | --- |
| **Asset** | **Naming** |
| Actor Blueprint | BP\_(Name) |
| Game Mode Blueprint | GM\_(Name) |
| Game State Blueprint | BP\_(Name)GameState |
| Blueprint Interface | BPI\_(Name) |
| Level | (LevelName) |
| Input Mapping Context | IMC\_(Name) |
| Input Action | IA\_(Name) |
| Structure | S\_(Name) |
| Widget Blueprint | WBP\_(Name) |
| Blueprint Component | BPC\_(Name) |

## Style Guide

**Coding style will be format by colour:**

|  |  |
| --- | --- |
| **Component Blueprint** | |
|  | Event/ Function that is for initialising purposes |
|  | Custom event blueprint |
|  | Code explanation purpose |
| **Widget Blueprint** | |
|  | Event/ Function that is for initialising purposes |
|  | Code explanation purpose |
|  | Suggestion/ reminder for future development/ improvement |
|  | Event/Function on setting widget component (Progress bar/ Button Click/ Text/ Animation etc.) |
| **Actor Blueprint** | |
|  | Code explanation purpose |
|  | Suggestion/ reminder for future development/ improvement |

## Commenting Rules

Commenting on the prototypes requires grouping blueprint nodes into group to tidy for reading. Title required in comments to display the group blueprint, for some detail complex blueprints, give detailed comments of describing what the blueprint do. Title and subtitle comments will need to active bubble zoom for developers to be aware the blueprint group, and also for some blueprint nodes that might need to be improved in future development.

# Production Overview

## Moscow

|  |  |
| --- | --- |
| Must | * Basic character interaction​ * Input/ interactive command (either mouse only or keyboard input)​ * Pawn stat |
| Should | * UI element on character’s requirements (hunger/ emotion etc.)​ * Pause Menu​ for player to pause game or quit game * Game time cycle (End game) |
| Could | * Inventory and shop​ * Currency * AI character movement when player not controlling pawn​ * Sound/ Music​ * Effect (eating/ emotion low hint)​ * Skill and learning system * Game ending scene base on character’s stats |
| Won’t | * Extra Game level * Extra Mini Game​ * Environment change (day and night)​ * Advance animation (eating/ emotion low hint) |

## Timeline

A diagram of a work flow

AI-generated content may be incorrect.

## Budgeting

This project’s manpower would be separated into 3 sections. First will be in charge of gameplay development, which would focus on the base system of character variables, input and control, and implement functions that create game loops, to make the game playable. That will be:

* Increase and decrease variable
* Player input and control
* Collision actor
* Gamemode/ Gamestate

Second will be in charge of UI/UX creation and implementation, which focus on widget creation and data linking:

* Set up game HUD
* Character attribute variable
* Interactable command button with function controlling
* Game ending scene
* Pause game panel
* Main Menu and credit panel
* Variable link and update

The final will be the polishing of the prototype, which focuses on implementing custom assets, mesh, sound effects, responsive feedback polish etc. to increase the game feel:

* Implement sound and animation feedback to in game widgets
* Custom widget asset to match with game theme and colour.