**Assignment 4: Adventure game architecture**

Version 1.

**Author**: Maryam Jafar 101079492.

**Date:** November 4, 2020.

**Scope:**

This document will explain design specifications for an online adventure game. It includes an introduction, the functional and non-functional requirements, the software architecture (Model View Controller), and class members for the design of the game.

**1.0 Introduction**:

The game is an online multiplayer that runs in persistently. It lets users create an account and login. The users are assigned groups of 3-5 and there are tasks for each group. The groups will be created by an admin user. The users have the option to either accept and perform the quests or simply wander around the game world and explore.

**2.0 Requirements**:

**2.1 Functional**:

2.1.1 Account

* + The game shall let the users register
  + The game shall register the admin
  + The game shall let the users log in

2.1.2 Admin

* + The game shall let admin create and name the groups
  + The game shall set a total min restriction of users per group to 3 and max to 5

2.1.3 Quests

* + The game shall present quests to admin
  + The game shall let admin choose quests
  + The game shall let admin assign quests to a group

2.1.4 Gameplay

* + The game shall present users with an option to chat with NPCs when approached
  + The game shall let users interact with certain objects in the environment
  + The game shall let the users collect objects
  + The game shall make the time pass
  + The game shall run NPC actions as time passes
  + The game shall present the user with choices

2.1.5 Database

* + The game shall store the account information in database for users
  + The game shall store the account information in database for admin

2.1.6 Render

* + The game shall render al the visual gameplay elements

2.1.7 Controls

* + The game shall let the user control the player by pressing keys
  + The game shall let the user pick weapon of choice hen fighting
  + The game shall let the user fight with their hands instead of weapon
  + The game shall let the user move the player by pressing keys
  + The game shall let the user choose desired interactions by pressing certain keys to select options

2.1.8 Animations

* + The game shall animate game objects

**2.2 Non functional:**

* The game shall connect to the internet
* The game shall provide user with ability to save progress in available slots

**3.0 Architecture (MVC):**

**3.1 Model**

**3.1.1 ofApp:**

* Data:
  + Game objects
  + Player and NPC objects
  + Keyboard array
  + Background images
* Code:
  + Setup()
  + Update()
  + Draw()
  + Event()

**3.1.2 NPC**:

* Data:
  + Position
  + ID
  + Name
  + Bool attack
  + Bool talk
  + Bool die
* Code:
  + NPCAttack()
    - * If attack is true :
      * Until die = true:
      * Attack player continuously by following them around and poking with sword
  + NPCTalk()
    - * While player doesn’t press KEY z to quit:
      * If Player x distance away, start conversation
      * Bool talk = true
  + NPCDie()

**3.1.3 Player**:

* Data:
  + Position
  + ID
  + Name
  + Bool talk
  + Bool die
  + Bool attack
  + Bool quitCovo
  + Group
* Code:
  + Attack()
    - * If attack is true:
      * Until die = true:
      * If KEY X pressed, attack with weapon
      * Else if KEY Y pressed, attack with hands
  + Talk()
    - * If NPC name = ‘...’
      * While quitConvo is not true:
      * Present response choices
  + Die()
  + chooseWeapon()
    - * Select weapon

**3.1.4 Quests**:

* Data:
  + ID
  + Name
  + Bool completed
  + Bool started
* Code:
  + DisplayQuestText()
  + CheckCompletion()

**3.1.5 Game**:

* Data:
  + Time
* Code:
  + Timer()
  + Main ()

**3.1.6 Animate**:

//from previous assignment examples

* Data
  + Frames array
  + lastFrame, firstFrame, numFrames, currentFrame
* Code
  + Draw and GetImage()
  + SetNumFrames()

**3.1.7 Weapon**:

* Data:
  + ID
  + Name
  + Bool inHand
  + Bool shoot
  + Bool slice
* Code:
  + Shoot()
  + Slice()

**3.1.8 Account**:

* Data:
  + Username
  + Password
* Code:
  + AdminLogin()
  + UserLogin()

**3.1.9 Admin**:

* Data:
  + Name
* Code:
  + Group()

**3.2 Controller**

**3.2.1 ofApp:**

* See 3.1.1

**3.2.2 Physics**:

//from previous assignment examples

* Data
  + groundLevels
* Code
  + ReadHeightMap()
  + GroundLevel()
  + MoveI()

**3.3 View**

**3.3.1 ofApp:**

* See 3.1.1

**3.3.2 Render:**

* Data:
  + none
* Code:
  + DrawTerrain()
  + DrawObjects()
  + DrawPlayer()
  + DrawNPC’s()