10.014: Computational Thinking for Design

1D Project

F09 – Team 9D

“SUTDent Life Simulator”

Student ID	Student Name
1007881	Annie Chan
1008162	Skipper Smahon
1008250	Alicia Choy
1008164	Hui Wan Sheng
1008204	Rebecca Tan
1008016	Matthew Phua

Description

Background of game:

Many SUTDents have trouble managing their time in school. Hence, we created a game with the objective of helping SUTDents improve their time management skills using python. We started off with the idea of creating a life simulator focused on time management with the aim of teaching players how to balance various aspects of life. Then, we started to branch out ideas which will help make our game more engaging and realistic.

Scenario:

As the latest cohort of incoming students embarks on their academic journey at SUTD, they are about to encounter the multifaceted challenges and array of responsibilities inherent to university life. At the same time, they are also discovering all the exciting new things they can do and experience while they're here. However, there are many trade-offs that must be made due to the limited resources available. As university students, one of the biggest limitations is the amount of time we have. Students at SUTD frequently grapple with the challenge of effective time management due to the constraints imposed by academic commitments. Hence, by introducing an interactive, choose-your-path simulation game, we immerse players in scenarios mirroring real-time dilemmas faced by university students, showcasing the different outcomes to the daily decisions that they make. By engaging in this interactive experience, students would not only grasp the significance of their decisions but also acquire valuable insights into crafting efficient time management strategies. Moreover, this gamified approach serves the dual purpose of addressing time management issues while engaging and captivating students through an entertaining and educational platform.

Why use our game:

Our game encourages SUTD students to think about who they want to be at university and how they handle their time. By playing, they get to reflect on what's important to them and how they manage their schedule. It's not just fun—it helps them figure out better ways to juggle their responsibilities. This game brings a bit of excitement to their everyday school life while getting them to think about their goals.

How is it played:

The main objective of the game is to complete all tasks and survive the SUTDent life with high statistics.

The player will have 5 statistics in the game. This includes social, knowledge, physical health, emotional health and money statistics.

These trackable statistics represent the player's progress in different areas such as academic performance (knowledge statistics), relationship status (social statistics) etc.

The higher the statistics the player acquires, the better game ending the player gets.

The game will last for 28 days. For the entire duration of the game, there is a list of tasks that SUTDents must do to graduate successfully. If the player does not complete the tasks in time, the player will be "expelled", signalling game over.

In the game, the player has a limited amount of time (hours) each day and a set of actions that he/she can choose to do with that time. Players must allocate time to different activities like academics, relationships, hobbies, work, etc. However, the players' choices throughout the game affects their time allocation and outcomes. For example, choosing to attend a party might improve social relationships (social statistics) but reduce available study time (knowledge statistics). Basically, different outcomes result from different decisions made. Good choices might improve certain statistics while poor choices might have negative consequences. Hence, the player will not only be required to complete all the necessary tasks, but also consider the best way to build a high statistic.

Besides that, there may also be random events that will be thrown at the player each day. These random events will affect the players statistics either positively or negatively and the player will have no control over the happening of these events. In addition, there is also a special random event which requires the player to guess a random number and would gain or lose points depending on the outcome.

Main features:

Statistics: Social, knowledge, physical health, emotional health and money statistics. This helps to visually quantify the different aspects of one's life and how different decisions affect them.

Varied endings: Different paths and choices lead to varied endings, demonstrating the consequences of time allocation, and keeps the game more challenging yet engaging as well.

Random events: This feature disrupt the players' routine which forces them to adapt and manage their time effectively.

Regular feedback on the consequences of choices made: Helps players understand the cause-and-effect relationship between time allocation and outcomes, which further helps to improve their decision-making skills on time allocation.

Simulating scenarios akin to real-life situations: Helps players learn to prioritise tasks and balance responsibilities.

Documentation

Import requirements:

- os, random, copy, time modules

Global Variables:

Variables	Description
exitvar	Used in the while loop of the main game body and allows the player to move between the menu pages of the game.
menuPage	The menu page displayed based on the option chosen in menu2() of the main game body
dayCount	Returns the day count of the player in the game
hours	Returns the hours per day the player has
hoursSpent	Returns the number of hours consumed by the activities
hwFailCount	Keeps track of the number of failed completions of the "SUTD Tasks"
sleep	Used as condition to allow the player to go to the next day
days	A list that contains the number of hours the user has based on the day. Monday – Friday = 6 hours Saturday – Sunday = 12 hours
twocounter	Keeps track of the number of times the player has selected the same option in the menu page at the start of the game *(menuPage == 0)*
stats_orig	The original stat list that the stats variable copies its values from. This is needed when the player restarts the game
stats	A list that keeps track of the amount of stat points the player has. Stats: Social, Knowledge, Health (Physical), Health (Mental).

SUTD Tasks:

There are 2 tasks per week that the player must complete. The player is required to attain the sufficient amount of "Knowledge" stat points on the deadline day to complete the required tasks.

The tasks will be checked through the taskFailTest() function at the end of the deadline day.

The player has only 3 chances for a failed submission of the tasks. If there is more than 3 failed attempts, then the game ends.

The tasks along with its requirements ('req'), deadline ('deadlineDay') and completion status ('status') are placed in a dictionary based on the week number.

Functions:

Function Name	Description
clearOutput()	Clears the previous outputs of the code
getDayNo()	It is used in menu page 2 to display the day (e.g., Monday, Tuesday, etc). The output of this function is called to get the keys of the "days" list above.
getHours()	It gets number of free hours according to the "day" list. This function is called at the start of every day.
gameExit()	It is used to exit the game when the game is over or allows the player to restart the game
deathcheck()	It ends the game when one of the stat points falls below 0
suddenDeath()	It displays the outcome of the player after the game ends based on which stat point falls below 0
taskFailTest()	Checks whether the player's "Knowledge" stat points fulfil the requirements of the "SUTD Tasks" based on the day count
guess_the_number()	Execute a small number guessing game that would be called in one of the random events.
done_Status()	This function calls the values of the "SUTD Tasks" dictionaries and returns the completion status of "SUTD Tasks" in menu2() option 6.
taskDisplay()	Displays which task is due on the day itself when called in menu2()
menu1()	This is the menu page displayed at the start of the game
menu2()	This is the main menu page displayed throughout the game which contains the activity options and miscellaneous actions for the player to choose, as well as displaying the day, and day count.
tasks()	This is the menu *(menu2() option 6)* that displays the "SUTD Tasks" along with its respective deadline, requirements, and completion status

Random Events:

Various single events will be called on random days throughout the game after the player chooses the sleep option (to end the day). These random events are functions which will affect the player's stat points by increasing or decreasing it though a short story or scenario. The random events could even include simple games like number guessing games. The probability of the player getting a random event is 65% each day.

Examples of random events:

```
================================================================================

Oh nooo you have a terrible headache!!!

Your mental health goes down by 5!

Press enter to continue:
```

Example of a random event output

```
def headverypain():
    print("================================================================================\n")
    print("Oh nooo you have a terrible headache!!!\n")
    print("Your mental health goes down by 5!\n")
    cont = input('Press enter to continue: ')
    mental[1] += -5
```

Example of code for random event

The name of the random events is placed in a dictionary as keys and is called at the start of the while loop when (menuPage == 1).

```
events = {
    1: 'papaDied',
    2: 'wrongSideOfBed',
    3: 'Bobsbirthday',
    4: 'physicsremedial',
    5: 'headverypain',
    6: 'adhocrun',
    7: 'glassesbroke',
    8: 'ifartinurface',
    9: 'ifartinurface',
    10: 'guessinggamez',
    11: 'python_in_ceiling',
    12: 'fall_into_drain',
    13: 'concussion',
    14: 'extranotes',
    15: 'newfriend'
    }
```

Random Events Dictionary

```
while menuPage == 1:
    try:
        hours = days[getDayNo()][1] - hoursSpent
        clearOutput()
        randNo =random.randint(0,100)
        if dayCount >= 1 and randNo > 35:
            randEvent = random.randint(1,len(events))
            function_name = events[randEvent]
            result = eval(function_name + "()")
```

How the random events are called

Game Running Code:

The game running code is made up of various while loops. The variable that determines which while loop to be executed is the menuPage variable, and it is changed through reassigning the value that is tied with the variable within the while loops by using input functions and fixed assigning i.e. (menuPage == 1).

Menu Page	Description
menuPage == 0	While menuPage == 0, this page is displayed at the start of the game. It calls the menu1() function and allows the player to start the game.
menuPage == 1	While menuPage == 1, this page is the main menu page that is displayed for the player. It calls the menu2() function and runs through the taskFailTest() and deathCheck() functions. The random events are also run at the start of the day. The player can choose from options 1 – 9 and the menuPage of the corresponding option chosen will be displayed.
menuPage == 'Study'	This page is displayed when the player chooses option 1 in menuPage == 1. The player is asked to input how many hours he/she wants to study for using the Free Time available. 'Knowledge' stat point will increase by 1 point for every hour spent studying.
menuPage == 'Friends'	This page is displayed when the player chooses option 2 in menuPage == 2. The player is asked to input how many hours he/she wants to spend time with friends using the Free Time available. 'Social' stat point will increase by 1 point for every hour spent.
menuPage == 'Exercise'	This page is displayed when the player chooses option 3 in menuPage == 2. The player is asked to input how many hours he/she wants to exercise using the Free Time available. 'Heath (Physical)' stat point will increase by 1 point for every hour spent.
menuPage == '5Row'	This page is displayed when the player chooses option 4 in menuPage == 2. The player is asked to input how many hours he/she wants to join Fifth Row activities using the Free Time available. 'Knowledge' and 'Social' stat point will increase by 0.5 points for every hour spent.
menuPage == 'Nap'	This page is displayed when the player chooses option 5 in menuPage == 2. The player is asked to input how many hours he/she wants to nap using the Free Time available. 'Health (Mental)' stat point will increase by 1 point for every hour spent.
menuPage == 'Tasks'	This page is displayed when the player chooses option 6 in menuPage == 2. It calls the tasks() function and displays the "SUTD Tasks" along with its respective deadline, requirements, and completion status.
menuPage == 'Statss'	This page is displayed when the player chooses option 7 in menuPage == 2. It displays the stat points of the player.
menuPage == 'Instructions'	This page is displayed when the player chooses option 8 in menuPage == 2. It displays the game instructions and details to the player.

Endings:

By the end of Day 28, the game ends and the player will get a summary of his/her stat points.

Function Name	Description
socialEnding()	Returns the ending description of the player based on the 'Social' stat points
knowEnding()	Returns the ending description of the player based on the 'Knowledge' stat points
phyEnding()	Returns the ending description of the player based on the 'Health (Physical)' stat points
emoEnding()	Returns the ending description of the player based on the 'Health (Mental)' stat points

Example of ending summary:

```
================================================================================
THANK YOU FOR PLAYING!
THE GAME HAS ENDED
================================================================================
DAMN you have a sexy and fit body !
You always have a great peace of mind
IT Girl / IT boy !! everybody wanna be you
otw to become valedictorian
================================================================================
You have reached the end of the game
Press enter to continue.
```

Example of an ending summary output

The ending descriptions are stored in a dictionary, and they are called using a function which compares the stats and a fixed range within if statements. And depending on the value of the stats, would call the description based on which if statement it satisfies.

```
knowledge_ending = {1: 'otw to become valedictorian', 2: 'graduated with honors', 3: 'graduated with average results ',
                    4:'barely graduated', 5:'dropped out of SUTD'}
def knowEnding():
    if knowledge[1]>100:
        return (knowledge_ending [1])
    elif knowledge[1]>=85:
        return (knowledge_ending [2])
    elif knowledge[1]>=50:
        return (knowledge_ending [3])
    elif knowledge[1]>=30:
        return (knowledge_ending [4])
    elif knowledge[1]>=0:
        return (knowledge_ending [5])
    else:
        return (None)
    pass
```

Code of dictionary and function for 'Knowledge' stats ending summary

```
elif option == 9:
    taskFailTest()
    dayCount += 1
    hours = getHours()
    hoursSpent = 0
    if dayCount == 28: # to get ending
        clearOutput()
        print("================================================================================")
        print('THANK YOU FOR PLAYING!')
        print('THE GAME HAS ENDED')
        print("================================================================================")
        print(phyEnding())
        print(emoEnding())
        print(socialEnding())
        print(knowEnding())
        print("================================================================================")
        print('You have reached the end of the game')
        print(input('Press enter to continue.'))
        gameExit()
```

Code for ending summary output