BUCKINGHAMSHIRE
NEW UNIVERSITY
EST. 1891

School of Creative and Digital Industries

COM4008 Programming Concepts: PR1

Module Title:	Programming Concepts	**Module Code:**	COM4008
Assignment No/Title:	PR1	**Assessment Weighting:**	70%
Submission Date:	**Friday 3rd November 2023** by 14:00 Module Week 5	**Feedback Date:**	+ 3 Weeks
Module Tutor:	Justin Luker	**Degree/Foundation:**	Foundation and Degree
Student ID:	22304094	**Student Name:**	Vanesa Efremova
Course:	BSc. (Hons) Computer Science		
Group Name:	Play Crazy		
GitHub Repository Link:	https://github.com/aminakxo/play.crazy		

Table of Contents:

The Aim of your Game (1 page maximum)

This section is an introduction to the game that you and your group have made. It is important that you can articulate the game idea. You should specifically state the style and genre of your game, the background game story for the level(s) you have developed, the objective for the game player(s). You should also reflect on your individual contribution to creating the game idea and how you interacted with the other group members.

Write your reflection here:

For this project we have created this simple game which we decided to call it Rattle Snake Rush. The concept of this game is about a snake on the hunt for food to be exact apples. The objective for the game is for the player to reach the highest score that he can while the snake is continuously moving on the play area under the direction of the player. Once the snake reaches and eats the apple, the player's points are increasing, and so does the length of the snake. The background game is also revolving around the hunt for the apple while the snake is facing some difficulties and challenges like avoiding the surrounding walls and its own tail which means that the player needs to make bold moves to protect the snake form touching the tail. The player can control the movements of the snake by using the arrows buttons on its keyboard choosing sides like up, down, left, or right. If the snake touches the walls or its own tail, then the game will end, and the player will lose This game belongs in the genre of sub-action video games style. As a team we interacted and communicated with each other in the campus and through our WhatsApp group in which we constantly interacted with each other disused ideas and goals that we had set for the game also we were making sure that we stick to our plan and complete everything in the given time interval. Together we discussed our separate roles in the group and which part of the creation we will be separately focusing on. We divided the workload among ourselves in a way that we had two main coders who are creating the roots of our game and coding it, we had one person who was creating the sounds and the graphics of our game, then another person who was organising and managing the team also documenting the process of the game and the last team mate was also helping with documenting the process along creating the presentation for the game meanwhile suggesting and implying different ideas for the game and ways to improve it.

ENTER YOUR FULL NAME AND YOUR STUDENT ID NUMBER HERE Friday 3rd November 2023 by 14:00

The Game Requirements (1 page maximum)

In this section you need to list the requirements of your game and put these at the top of the section as a Bullet List (please refer to the Space Invader demo from the lecture materials). This list should be the same for everyone in the group.

Below this list you need to provide a written reflection on what your contribution was to the establishment of each requirement within the group. Reflect on why these requirements were selected by the group and consider if these were the most appropriate requirements for the game overall. Reflect on if and how the requirements changed over the development process. Would you do something different if you had the opportunity to do the assignment again?

Write your requirements list and reflection here:

Creating the design of the game: This was quite straightforward for everyone since the main coder of our group with the support of the other members decided to create a game with an action genre which later was finalised as a snake game. Together we looked at assorted designs that were suggested and presented by our main coders. We did change the designs over time because as coding the game the coders became more confident and experimental towards the visuals, so we upgraded them and developed a better option.

Background: We downloaded an image from Google for the background which was a picture of a grass. We did this because it was more convenient for us to implement this picture as beginners, and it was not as much time consuming. We used the "pygame.image.load" function to upload background image.

Music: We downloaded background music from Google and uploaded in our game using the "pygame.mixer.init()" function. This was appropriate for the game as it was remarkably simple.

Apple: We downloaded an image of an apple from Google and uploaded it onto our game using the "pygame.image.load" function. This made it easier for us to create the apple instead of having to do it manually. This was all simply agreed by the group since it was considerably basic to accomplish.

Creation of the snake: We downloaded a block image from Google and then we used pygame.image.load function to upload the image and then we increased the length of the snake using the "self.length" function. We agreed on these choices for the game as we all are beginners and not so experienced in this filed yet, so we were trying to get the best products using simple tools to help us build our game and increase our knowledge.

For the next project we would make most of the things differently as we will have more experience everyone will be engaged more in the coding and, we would create something more difficult and challenging which will make us push our limits.

Reflection on Your Role in the Group (1 page maximum)

In this section your need to reflect on what your role was in the group and how you established that role. You need to discuss how the group divided up the work between the group members and your contribution to this process. How did you communicate the objectives of your role to the rest of the group e.g. how did you go about organising your commits via GitHub and letting the other group members know what you had been working on. You need to reflect on the following questions: What was the significance of your contribution to the game design? What was the significance of contribution to the development of the game code? Can you discuss specific code that you worked on or other game elements that you created? How would you evaluate your role in the group?

Write your reflection here:

My role in the group was suggesting ideas for the visuals, the code, the sound track also making sure we are competing all the requirements as well as creating a presentation to present the game that we created along documenting the process of creation, development, and progress. We divided our workload correctly into fileds that everyone was feeling most confident in. We had two coders, one audio and graphics creator, one organiser and documentation creator and me as innovator, presentation creator and part of the documentation creation. However, all of us were working together on the game as we were constantly advising with each other and suggesting improvements. We mostly communicated on campus and on WhatsApp to share our ideas, time manage everything and stick to the plan, although all our work was uploaded to GitHub daily as it was the right place to gather everything together and discus it, even though it was a bit challenging for us at first because we none of us had any previous experience of using it, we managed to learn how to use it. My biggest contribution in the game was communicating with everyone as I had to make sure I had everything documented so it can be presented in our presentation also even though I was not the initial coder of this game I was still suggesting ideas of improvement and development to my teammates. I specifically went through every change of our game so I can capture the development of it and finalise it in the presentation. I really liked my team because all of us are really working hard and nonstop to create this project in the best way we can also I really liked that we were advising each other even though we had separate roles. Further on I would evaluate my role in the group by engaging more in the creation part of the game as the graphics and the sounds which will help me engage and challenge myself to try something that I'm not confident in and something that I have not done before.

Reflection on the Game Development Process (1 page maximum)

In this section you need to reflect on how the group set about developing the actual game code and any game assets (sound FX, music, graphics, menus). If you are new to programming, how did you find the process of learning, whilst working on a game? If you already had some programming experience, how did you challenge yourself to learn more whilst work on the game? How did your group divide up the programming tasks? What problems did you encounter and have to solve programmatically? Were there any problems that you didn't manage to solve? What would you add to your game if you could continue to develop it further? Did you have any issues with synchronising your code edits in GitHub? How did you manage code comments and the structure of your code? How did you use classes within your code? What parts of your code were the most challenging to develop and why?

Write your reflection here:

For this project we developed mostly the code by changing the design of the snake and its extension. We started with a quite simple design which was the in our comfort zone but furthermore we realised that we needed something more advanced and detailed. The process itself was challenging as we are beginners in python and this change was requiring more knowledge and dedication. We created this development when we swapped our square shaped snake, which was not giving the initial simulation of a snake and that did not have the right structure. We developed this by creating a snake in a different shape and we added more details to make it looks more realistic and to initiates the character of the snake clearly. As a group we all divided our tasks equality separating them on sections in which we felt most confident in, meanwhile we were continuously supporting each other through the whole process and communicated the changes that we wanted to be made. To evaluate this if we had more time, we would add an introduction page to the game which would make the game more professional and advanced, we will also create more levels to be completed so the game can be challenging and fun, we would add variety of sounds that will represent the different levels as well as much more graphics. However, for this code we had three classes on which our game was focused on and the most challenging part was the creation and the design of the snake, we added comments very frequently along the creation of this game which was very helpful for the members of the group who were not so strong in coding field, because for them it was easier to understand the code better and to get more knowledge about it. All the documents that we created we uploaded them to the GitHub which was our main program for sharing our work and providing feedback as well as advising each other on the things that we could have changed and had done differently.

References

Under this section, please reference any sources that you used to help you in the game design and development. These might be references to game tutorials online, or royalty free game assets, or online or textbook code teaching/learning resources etc. Please use the Harvard referencing format (see online guide to referencing in Harvard at: https://www.citethemrightonline.com/ (look for institution login and use your Uni login to access under the Bucks New University subscription).

Write your references here:

Free sound effects • Uppbeat. https://uppbeat.io/sfx?rt=ppc_google_search_sfx_uk&utm_source=google&utm_medium=cpc&utm_campaign=search_sfx_uk&utm_content=659094315778&utm_term=free%20digital%20sound%20effects&gad=1&gclid=CjwKCAjwkY2qBhBDEiwAoQXK5VFY0OjvGCa74Wtbn9u0zGJ1OJrQzaJZEQAS7kXF3AZhuxK-r-dc-RoCTZ0QAvD_BwE. (Accessed: 1 November 2023)

Red Apple Fruits PNG Transparent Images Clipart Icons - Transparent Background Apple Clipart (2019) Nohat. https://nohat.cc/f/red-apple-fruits-png-transparent-images-clipart-icons-transparent-background-apple-clipart/m2K9A0m2A0K9b1A0-201907231217.html. (Accessed: 30 November 2023).

Snake game with PyGame | Replit Docs (2023). https://docs.replit.com/tutorials/python/build-snake-with-pygame. (Accessed: 23 October 2023).

https://www.google.com/search?q=dark%20green%20image%20for%20game&tbm=isch&tbs=rimg:CVYI2oE-ugTCYesncsXRkBCesgIVCgIIABAAKAE6BAgBEAFAAVW-g1I_1wAIA2AIA4AIA&hl=en&sa=X&ved=0CBoQuIIBahcKEwig7avypKWCAxUAAAAAHQAAAAAQDg&biw=1257&bih=598. (Accessed: 1 November 2023).

Appendix A: Code Snippets

Under this section, please list the code snippets that you specifically wrote and worked on during the development of the game.

Copy your specific code snippets here:

Our code snippets are as follows:

```
import pygame

from pygame.locals import *

import time

import random

size = 40
```

Appendix B: Game Code

Under this section, please put a copy of the game code. Use subheadings to indicate code from different files and indicate the folder/file structure that you used. We should be able to get to your code online via your link to your GitHub repository, but it is useful to have your code offline as well. This section will be the same for everyone in the group.

Copy and paste your full game code here:

```
# import is used to iclude path of a class or the entire package in program
import pygame


# pygame.locals used to include constants in game.
from pygame.locals import *
import time
import random


size = 20



# Classes provides template for creating objects, which can bind code into data.
class apple:
    # Python def keyword is used to define a function.
    def __init__(self, parent_screen):
        # Self represents the instance of the class.
        self.parent_screen = parent_screen
        # pygame.image.load function call is used to load images.
        self.image = pygame.image.load("resources/app.png").convert()
        self.x = size * 3
        self.y = size * 3

    # draw() function is used to draw image on a specific point.
    def draw(self):
        self.parent_screen.blit(self.image, (self.x, self.x))
        # display.flip() is important everytime when we make any changes on display.
        pygame.display.flip()

    # move() function is used to move image in different sections of screen.
    # random module is used to generate random numbers.
    def move(self):
        self.x = random.randint(1, 14) * size
        self.y = random.randint(1, 19) * size


class Snake:
    def __init__(self, parent_screen, length):
        self.parent_screen = parent_screen
        self.block = pygame.image.load("resources/downmouth.gif").convert()
        self.direction = "down"
        self.body = pygame.image.load("resources/body.png")
        self.length = length
        self.x = [20] * length
        self.y = [20] * length

    def move_up(self):
        if self.direction != "down":
            self.direction = "up"
            self.block = pygame.image.load("resources/upmouth.gif")

    def move_down(self):
        if self.direction != "up":
            self.direction = "down"
            self.block = pygame.image.load("resources/downmouth.gif")

    def move_left(self):
        if self.direction != "right":
            self.direction = "left"
            self.block = pygame.image.load("resources/leftmouth.gif")

    def move_right(self):
        if self.direction != "left":
            self.direction = "right"
            self.block = pygame.image.load("resources/rightmouth.gif")

    def walk(self):
        # self.prev_x = self.x[0]  # Save the previous head position
        # self.prev_y = self.y[0]


        for i in range(self.length - 1, 0, -1):
            self.x[i] = self.x[i - 1]
            self.y[i] = self.y[i - 1]


        if self.direction == "up":
            self.y[0] -= size
        elif self.direction == "down":
            self.y[0] += size
        elif self.direction == "right":
            self.x[0] += size
        elif self.direction == "left":
            self.x[0] -= size


        self.draw()


    def draw(self):
        # Draw the head
        self.parent_screen.blit(self.block, (self.x[0], self.y[0]))


        # Draw the body segments
        for i in range(1, self.length):
            self.parent_screen.blit(self.body, (self.x[i], self.y[i]))
        pygame.display.flip()

    def increase_length(self):
        self.length += 1
        self.x.append(1)
        self.y.append(1)


class Game:
    def __init__(self):
        pygame.init()
        pygame.mixer.init()
        self.play_backgroud_music()
        self.surface = pygame.display.set_mode((800, 600))
        self.snake = Snake(self.surface, 1)
        self.snake.draw()
        self.apple = apple(self.surface)
        self.apple.draw()

    def render_background(self):
        bg = pygame.image.load("resources/background-image.png")
        self.surface.blit(bg, (0, 0))
    def is_collision(self, x1, y1, x2, y2):
        if x1 >= x2 and x1 < x2 + size:
            if y1 >= y2 and y1 < y2 + size:
                return True
        return False

    def display_score(self):
        font = pygame.font.SysFont("Arial", 30)
        # score is set to snake length - 1 as length start with 1 as snake head.
        score = font.render(
            f"Score: {self.snake.length * 10 -10}", True, (200, 200, 200)
        )
        self.surface.blit(score, (650, 10))

    def show_game_over(self):
        self.render_background()
        font = pygame.font.SysFont("Arial", 30)
        line1 = font.render(
            # score is set to snake length - 1 as length start with 1 as snake head.
            f"Game Over! Score : {self.snake.length * 10-10}",
            True,
            (200, 200, 200),
        )
        self.surface.blit(line1, (200, 300))
        line2 = font.render(
            f"To play again press Enter!",
            True,
            (200, 200, 200),
        )
        self.surface.blit(line2, (200, 350))
        pygame.display.flip()
        pygame.mixer.music.pause()

    def reset(self):
        self.snake = Snake(self.surface, 1)
        self.apple = apple(self.surface)

    def play_backgroud_music(self):
        pygame.mixer.music.load("resources/bg.music.mp3")
        pygame.mixer.music.play(-1, 0)

    def play_sound(self, sound):
        sound = pygame.mixer.Sound(f"resources/{sound}.mp3")
        pygame.mixer.Sound.play(sound)

    def play(self):
        self.render_background()
```

```
self.snake.walk()
self.apple.draw()
self.display_score()
pygame.display.flip()


# snake collidig with apple
if self.is_collision(
    self.snake.x[0], self.snake.y[0], self.apple.x, self.apple.x
):
    self.play_sound("crunch.1")
    self.snake.increase_length()
    self.apple.move()


# snake colliding with itself
for i in range(2, self.snake.length):
    if self.is_collision(
        self.snake.x[0], self.snake.y[0], self.snake.x[i], self.snake.y[i]
    ):
        self.play_sound("game-over")
        raise "Collision Occured"
# snake colliding with boundries of window
if not (0 <= self.snake.x[0] <= 800 and 0 <= self.snake.y[0] <= 600):
    self.play_sound("game-over")
    raise "Hit the boundry error"
    def run(self):
        running = True
        pause = False

        while running:
            for event in pygame.event.get():
                if event.type == KEYDOWN:
                    if event.key == K_ESCAPE:
                        running = False
                    if event.key == K_RETURN:
                        pygame.mixer.music.unpause()
                        pause = False
                    if not pause:
                        if event.key == K_UP:
                            self.snake.move_up()
                        if event.key == K_DOWN:
                            self.snake.move_down()
                        if event.key == K_LEFT:
                            self.snake.move_left()
                        if event.key == K_RIGHT:
                            self.snake.move_right()

                elif event.type == QUIT:
                    running = False

            try:
                if not pause:
                    self.play()
            except Exception as e:
                self.show_game_over()
                pause = True
                self.reset()

            time.sleep(0.1)


if __name__ == "__main__":
    game = Game()
    game.run()
```