COMPUTER GRAPHICS AND ANIMATION
PROGRAMMING ASSIGNMENT 15
Sprite Animation – Neon Tiger

CGA IT-3

Arya Sena Wiryady / 001201900110
M. Ichsan Nur Iman / 001201900034
Panji Arlin Saputra / 001201900037

1. Introduction.

- What is the program about?
  - The program is about sprite animation game with characters namely Megaman and Neon Tiger which have some powers and abilities that can be controlled by the user.

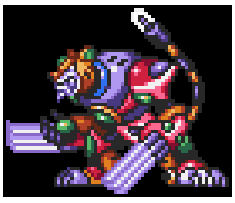
- In what language is the program implemented?
  - The Program is implemented using VB (Visual Basic) as the programming language.
  - Why did we select VB as the programming language?
    because it is easy to understand and learn, can be used to create windows application projects, can be created using Visual Studio IDE, and has several important tools and methods for creating sprite animation such as timer and the math method.

2. Basic theory.

- Explain sprite animation.
  - Sprite is a single graphic image that is incorporated from a larger scene for a frame in animation, sprite animation mean is an animation that consists of several smaller images (sprites) that can be manipulated separately from the rest of the scene, to make the animation moving.

- Explain how to put a sprite on a background.
  - To put a sprite on a background we need to create a “mask” of the sprite. Mask is a black contour on white background. To put the sprite on the background its recommended to change the sprite sheet background into black background.

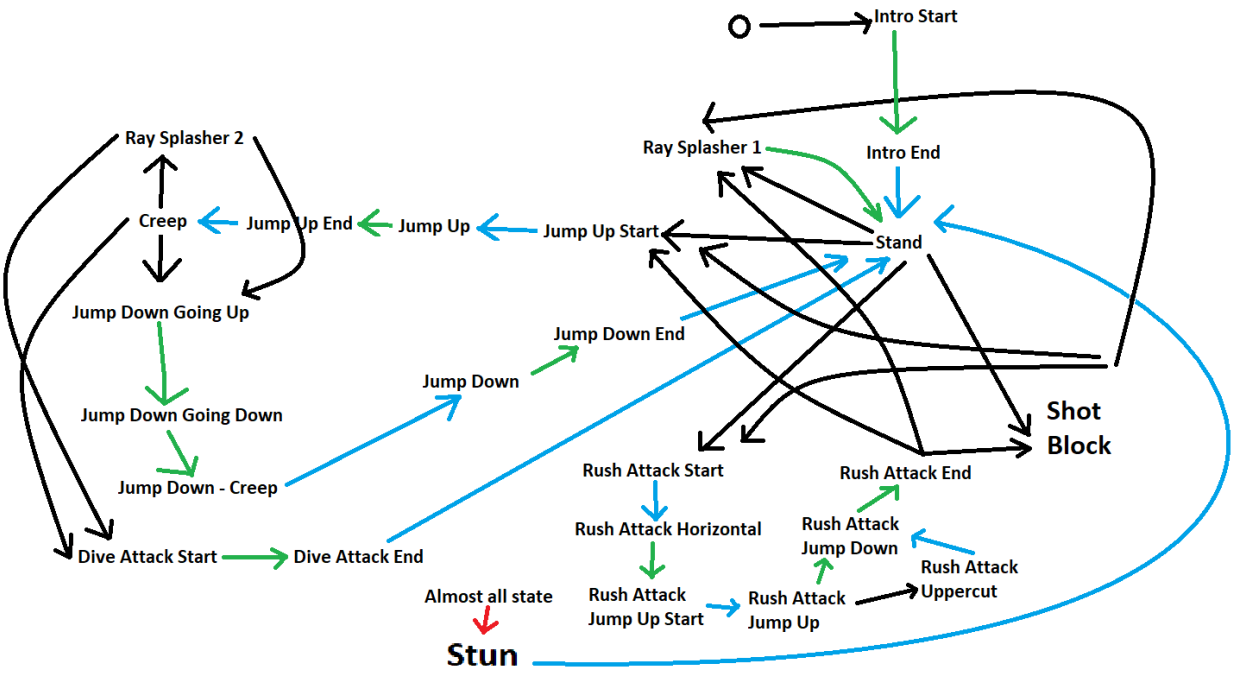

MASK


SPRITE

- Then perform an AND operation between the background and the sprites.

- Next, perform an OR operation between the background and the sprite.

- Explain character state diagram and state trasition table.
  - Neon tiger State Diagram

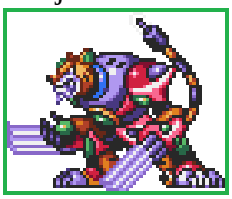


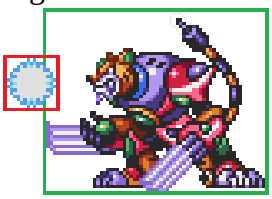
- Neon Tiger State Transition

Previous State	New State	Trigger
Intro Start	Intro End	NT lands on the ground
Intro End	Stand	Intro End animation is finished
Stand	Shot Block	NT decides to block shots
Stand	Ray Splasher 1	NT decides to do Ray Splasher 1
Stand	Jump Up Start	NT decides to do Jump Up Start
Stand	Rush Attack Start	NT decides to do Rush Attack Start
Stand	Stun	NT gets hit by Mega Man
Shot Block	Ray Splasher 1	NT decides to do Ray Splasher 1
Shot Block	Jump Up Start	NT decides to do Jump Up Start
Shot Block	Rush Attack Start	NT decides to do Rush Attack Start
Ray Splasher 1	Stand	NT goes to stand after a couple of shots
Ray Splasher 1	Stun	NT gets hit by Mega Man
Jump Up Start	Jump Up	Jump Up Start animation is finished
Jump Up Start	Stun	NT gets hit by Mega Man
Jump Up Start	Death	NT gets hit and HP = 0
Jump Up	Jump Up End	NT reaches a wall
Jump Up	Stun	NT gets hit by Mega Man
Jump Up End	Creep	Jump Up End animation is finished
Jump Up End	Stun	NT gets hit by Mega Man
Creep	Ray Splasher 2	NT decides to do Ray Splasher 2
Creep	Jump Down Going Up	NT decides to do Jump Down Going Up
Creep	Dive Attack Start	NT decides to do Dive Attack Start
Creep	Stun	NT gets hit by Mega Man
Ray Splasher 2	Jump Down Going Up	NT decides to do Jump Down Going Up
Ray Splasher 2	Dive Attack Start	NT decides to do Dive Attack Start
Ray Splasher 2	Stun	NT gets hit by Mega Man
Jump Down Going Up	Jump Down Going Down	NT reaches the ceiling
Jump Down Going Up	Stun	NT gets hit by Mega Man
Jump Down Going Down	Jump Down – Creep	NT reaches a wall
Jump Down Going Down	Stun	NT gets hit by Mega Man
Jump Down – Creep	Jump Down	Jump Down – Creep animation is finished
Jump Down – Creep	Stun	NT gets hit by Mega Man
Jump Down	Jump Down End	NT lands on the ground
Jump Down	Stun	NT gets hit by Mega Man
Jump Down End	Stand	Jump Down End animation is finished
Jump Down End	Stun	NT gets hit by Mega Man

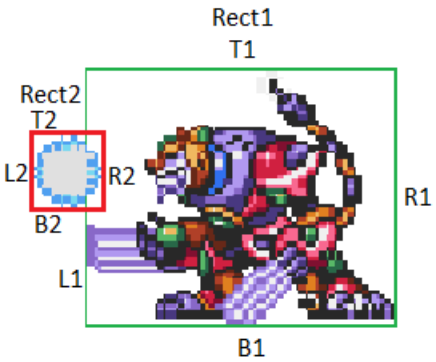
Dive Attack Start	Dive Attack End	NT lands on the ground
Dive Attack Start	Stun	NT gets hit by Mega Man
Dive Attack End	Stand	Dive Attack End animation is finished
Dive Attack End	Stun	NT gets hit by Mega Man
Rush Attack Start	Rush Attack Horizontal	Rush Attack Start animation is finished
Rush Attack Horizontal	Rush Attack Jump Up Start	NT reaches a wall
Rush Attack Jump Up Start	Rush Attack Jump Up	Rush Attack Jump Up Start animation is finished
Rush Attack Jump Up	Rush Attack Uppercut	NT decides to do Rush Attack Uppercut
Rush Attack Jump Up	Rush Attack Jump Down	NT reaches the ceiling
Rush Attack Uppercut	Rush Attack Jump Down	Rush Attack Uppercut animation is finished
Rush Attack Jump Down	Rush Attack End	NT lands on the ground
Rush Attack End	Shot Block	NT decides to do Shot Block
Rush Attack End	Ray Splasher 1	NT decides to do Ray Splasher 1
Rush Attack End	Jump Up Start	NT decides to Jump Up Start
Stun	Stand	NT lands on the ground or staggers back by 20 pixels from previous position

- Explain collision detection, especially the method of collision detection used in the program
  - Collision detection is a method or mechanism to detect if the there is a two or more objects collide or hit each other
  - Each object has its own area, in this program the area is in the form of rectangle



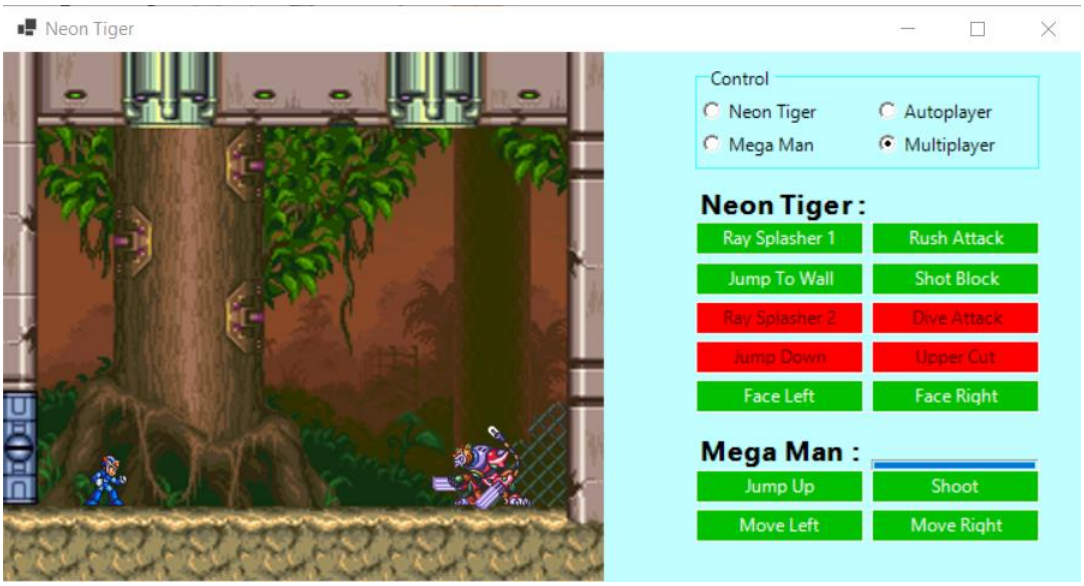


  - Collision occurs if two areas overlap each other



3. Implementation

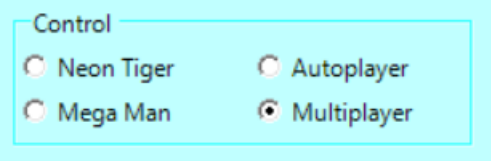
- Explain the main interface of the program.



This is the main interface of this program, as we can see, on the left side is the area where the animation is displayed while on the right is the control area where the user can control the powers and abilities of the characters.

- Explain every feature in the program and how to use them.

This is an animation area where every movement of each character is displayed here, Neon Tiger has the ability to hang on the wall so we provide a background with a wall inside.



This is the control area, here the user can determine what characters will be controlled by selecting the radio button. If Neon Tiger is selected, the user can only control the powers and abilities of the Neon Tiger while Megaman will move automatically, and vice versa if Megaman is selected, the user can only control the powers and abilities of Megaman while Neon Tiger will move automatically. If Autoplayer is selected both characters will move automatically and vice versa if Multiplayer is selected the user can control all the powers and abilities of both characters.

This is an area where just by cliking the button, the powers and abilities of the characters can be controlled. The green button indicates that the button is enable and vice versa the red button indicates that the button is disable.

4. Design

- Explain the main data structures (if any) used in the program.
  - The main data structure in this program is array. Array is a variable that is able to store multiple data of the same data type.

  - Explain how the background and sprite data are represented in the program.
    - All sprite characters are formed in the form of a sprite sheet. Because the sprite sheet and background are presented in a file in bitmap format, so we assign each pixel point of the image. The background and sprite areas are represented as rectangular areas so we need pixel coordinates from the left, top, right, bottom, and center. Afterwards each pixel coordinate data is assigned to array.

- Explain the main/global variables used in the program.
  - There are several global variables used in the main class (Class Form1) of this program

```
Dim bmp As Bitmap
Dim Bg, Bg1, Img As CImage
```

These global variable are used to provide the background from bitmap file into the program

```
Dim SpriteMap As CImage
Dim SpriteMask As CImage
```

These global variable are used to provide the character sprite into the program

```
Dim ListChar As New List(Of CCharacter)
```

This global variable is used to store all data of each character that will be displayed in the program

```
'NeonTiger
Dim NT As CCharNeonTiger
Dim NTIntroStart, NTIntroEnd, NTStand, NTShotBlock, NTRaySplasher1, NTJumpUpStart As CArrFrame
Dim NTJumpUp, NTJumpUpEnd, NTCreep, NTRaySplasher2, NTJumpDownGoingUp As CArrFrame
Dim NTJumpDownGoingDown, NTJumpDownCreep, NTJumpDown, NTJumpDownEnd As CArrFrame
Dim NTDiveAttackStart, NTDiveAttackEnd, NTRushAttackStart, NTRushAttackHorizontal As CArrFrame
Dim NTRushAttackJumpUpStart, NTRushAttackJumpUp, NTRushAttackUpperCut As CArrFrame
Dim NTRushAttackJumpDown, NTRushAttackEnd, NTUpStun, NTDownStun As CArrFrame

'Neon Tiger Projectile
Dim NTP As CCharNeonTigerProjectile
Dim ProjRaySplasher As CArrFrame

'Mega Man
Dim MM As CCharMegaMan
Dim MMIntro, MMStand, MMMove, MMJumpUpStart, MMJumpUp, MMJumpDown, MMJumpDownEnd As CArrFrame
Dim MMShoot, MMJumpShoot, MMJumpMove, MMStun, MMRecallStart, MMRecallEnd As CArrFrame

'Mega Man Projectile
Dim MMP As CCharMegaManProjectile
Dim ProjMegaManShoot As CArrFrame
```

These global variable is used to initialize data sprites, there are many variables in initializing sprites because each state contains different animated sprites

- Explain the character state diagram and the state trasition table for Neon Tiger.
  - Explain each state in the state diagram.
    - Intro Start is automatically displayed first when the program is running, NT jump down from the top to the ground then Intro End is displayed NT will perform several poses then move to Stand state.

    - In Stand state, NT can decide to move to several states such as Ray Splasher 1, Shot Block, Rush Attack Start, or Jump Up Start.

    - When the Shot Block state is displayed, NT raises its claws and NT can deflect all attacks from the enemy, NT cannot do Shot Block twice in a row so after the animation is complete NT must move to a different power or ability.

    - In Ray Splasher 1 state, NT shoots five projectiles in succession to different directions.

- When NT does Rush Attack, NT's skin color changes and NT becomes invincible, Rush Attack consists of several states such as Rush Attack Start (NT changes its skin color), Rush Attack Horizontal (NT attacks horizontally), Rush Attack Jump Up Start (NT prepares to attack vertically), Rush Attack Jump Up (NT jumps up), Rush Attack Uppercut (NT performs an uppercut attack in the air), Rush Attack Jump Down (NT returns to the ground), and Rush Attack End (Rush Attack completed and NT changes its skin color back to normal), NT cannot do Rush Attack twice in a row so after the animation is complete NT has to move to a different power or ability.

- In the Jump Up Start state, NT prepares to jump to the wall in front of him, then continues to the Jump Up state where NT does oblique jump towards the wall, when it gets to the wall the Jump Up End state is displayed, then after that it moves to the Creep state where this state NT hangs on the wall.

- When the Creep state is displayed, NT can decide for further moves such as Ray Splasher 2, Dive Attack Start, or Jump Down Going Up.

- NT shoots five projectiles in a row at Ray Splasher 2 state to different directions, NT cannot do Ray Splasher 2 twice in a row so after animation is complete NT must move to a different power or ability.

- NT jumps down when Dive Attack Start then when landing on the ground, NT moves to Dive Attack End state where in this state NT attacks nearby enemy.

- In the Jump Down Going Up state, NT jumps up when then when NT touches the ceiling, NT jumps down towards the wall in front of him (Jump Down Going Down), after that NT hangs on the wall for a while (Jump Down - Creep) then jumps down again (Jump Down) then landing on the ground (Jump Up End).

- In all states except Rush Attack and Shot Block, when NT receives an attack, NT moves to the Stun state NT will fall to the ground when NT receives an attack in the air and NT will stagger back when NT receives an attack on the ground.

- The sprites used for the animation of each action (include the duration of each frame).
  - Intro Start

1 1 → Duration of each frame

- Intro End

1 1 5 1 1 1 1 1 1

2 8

- Stand

10

- Shot Block

1 1 4 10 2

- Ray Splasher 1

1 1 1 1 1

1 1 1 1

4 2 2 1

4 2 2 1

4 2 2 1

4 2 2 1

- Ray Splasher 2

1 1 1 1 1

1 1 1 1 1

4 2 2 1

4 2 2 1

4 2 2 1

4 2 2 1

- Ray Splasher 1&2 (Projectile)

1 1 1

- Jump Up Start

2

- Jump Up & Jump Down going up

1 1

- Jump up End

2

- Creep & Jump Down – Creep

10

- Jump Down & Jump Down Going Down

1 1

- Jump Down End

2

- Dive Attack Start

1 1

- Dive Attack End

1 2 2 1 1

- Rush Attack Start

1 1 1 1 1 1 1

1 1 1 1 1 1 1

- Rush Attack Jump Up Start

2

- Rush Attack Jump Up

1 1

- Rush Attack Uppercut

2 2

- Rush Attack Horizontal

1

- Rush Attack Jump Down

1 1

- Rush Attack End

1 1 1 1 1

- Stun

When getting hit on the ground:

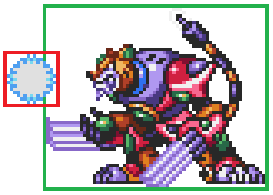
1 1 1

When getting hit on the air:

1 1

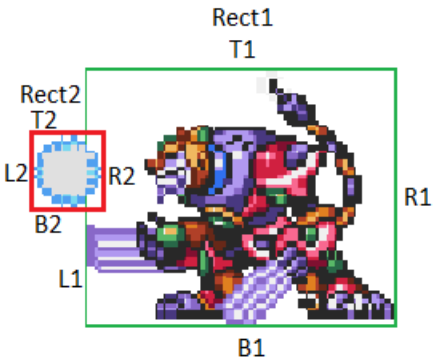
- Explain how the collision detection works in the program.
  - Each object has its own area, in this program the area is in the form of rectangle

  - Collision occurs if two areas overlap each other



  - Pseudocode that we use in the program:

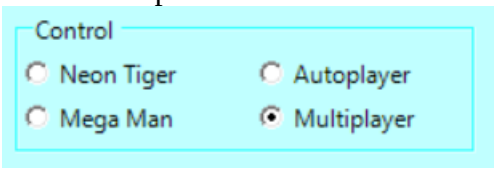
```
Rect1 = [L1, T1, R1, B1]              → Initializing the coordinate point of sprite
Rect2 = [L2, T2, R2, B2]              → Initializing the coordinate point of sprite
CollisionDetection(Rect1, Rect2){
    condition1, condition2 = False
    If (L2 > L1 AND L2 < R1) OR (R2 > L1 AND R2 < R1) OR (L2 < L1 AND R2 > R1){
            condition1 = True         → Checking left and right point overlap each other or not
    }
    If (T2 > T1 AND T2 < B1) OR (B2 > T1 AND B2 < B1) OR (T2 < T1 AND B2 > B1){
            condition2 = True         → Checking top and bottom point overlap each other or not
    }
    Return condition1 AND condition2
}
```



- Explain the pseudocode for each time tick.

```
TimerTick(){
    PictureBox.Refresh()                    → Refreshing the PictureBox
    If NeonTiger.CurrentState = Shoot{
        CreateNeonTigerProjectile()         → Creating projectile when Neon Tiger shooting
    }
    If Megaman.CurrentState = Shoot{
        CreateMegamanProjectile()           → Creating projectile when Megaman shooting
    }
    ListChar1 As CCharacter                 → Declaring ListChar1
    For each CC in ListChar{                →Looping all the data inside
        If CC.Destroy = False{              → To remove the projectile data that have to be removed
            If CollisionDetection(NeonTiger, MegamanAttack){
                NeonTiger.CurrentState = Stun  → When Neon Tiger under attack
            }
            If CollisionDetection(Megaman, NeonTigerAttack){
                Megaman.CurrentState = Death → When Megaman under attack
            }
            ListChar1.Add(CC)               → Storing the new data to ListChar1
        }
    }
    ListChar = ListChar1                    → Assigning new data to global variable
    DisplayImg()                            → Displaying new image to PictureBox
    UserButtonControl()                     → To control the user button controller
}
```

- Explain how the bonuses (if done) are implemented.
  - Add more options for the user to control the character



  - If Neon Tiger is selected, the user can only control the powers and abilities of the Neon Tiger while Megaman will move automatically.
  - If Megaman is selected, the user can only control the powers and abilities of Megaman while Neon Tiger will move automatically.
  - If Autoplayer is selected both characters will move automatically.

- If Multiplayer is selected the user can control all the powers and abilities of both characters.

➢ We provide button controller to make it easier for the user to control the character

- Green button indicates that the button is enable
- Red button indicates that the button is disable

5 Evaluation

- Evaluate the following test cases:
- Include screenshots of each test case.
- Explain whether each case is successful.
  - Starting the program (the intro animation). (SUCCESS)

  - Neon Tiger jumping to a wall. (SUCCESS)

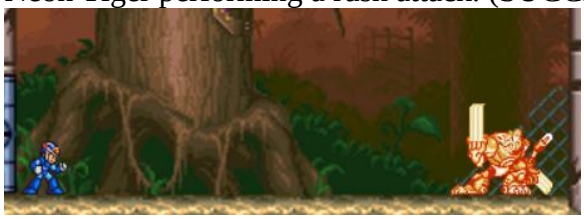
- Neon Tiger hanging on a wall. (SUCCESS)

- Neon Tiger shooting Ray Splashers (while standing and hanging on a wall). (SUCCESS)

- Neon Tiger jumping up after hanging on a wall. (SUCCESS)

- Neon Tiger diving from a wall. (SUCCESS)

- Neon Tiger performing a rush attack. (SUCCESS)



- Neon Tiger blocking Megaman’s projectiles. (SUCCESS)

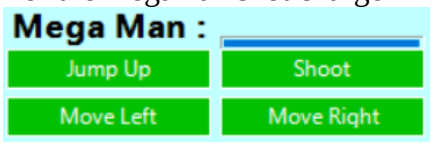
- The dummy getting hit by Neon Tiger’s attacks. (SUCCESS)

- Also perform a test case for all the bonus features you implemented.
  - Megaman shot is deflected by the Neon Tiger shot block. (SUCCESS)

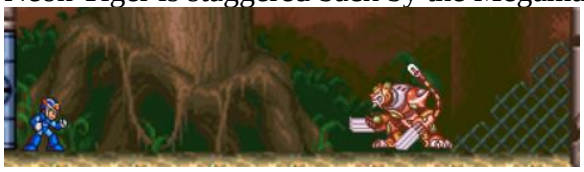
- Neon Tiger performing rush attack uppercut. (SUCCESS)

- Megaman shoots a projectile. (SUCCESS)

After Megaman shoots, his shot needs to be charged for 1 second, after 1 second he can do the next shot. We provide an indicator for the megaman shot charge



- Neon Tiger is staggered back by the Megaman's shot. (SUCCESS)



- Megaman jumps up. (SUCCESS)

- Megaman moves right. (SUCCESS)

- Megaman moves left. (SUCCESS)

- Megaman getting electrocuted when hit by lightning. (SUCCESS)

- Megaman getting destroyed. (SUCCESS)

- Megaman respawns at random location. (SUCCESS)

- Megaman is invincible when he respawns. (SUCCESS)

6 Work log.

- Record the date and time of every moment you work on this assignment and job description of each member at each session.

Date	Activity / Progress	Personnel Involved
April 17th, 2021	Constructing state diagram	Panji
April 18th, 2021	Creating state transition	Ichsan
April 19th, 2021	Arranging animation sprite	Arya
April 20th, 2021	Presenting our projects in class	Panji, Ichsan, Arya
April 24th, 2021	Starting to make the final report	Arya
April 28th, 2021	Starting to build the program, providing the background	Panji, Arya
April 29th, 2021	Fixing the motion of the character	Panji, Ichsan
April 30th, 2021	Creating projectile, dummy, and powers and abilities for the characters	Panji

May 8th, 2021	Creating user control	Panji
May 10th, 2021	Fixing the user control	Panji
May 11th, 2021	Creating collision detection	Panji
May 12th, 2021	Fixing the user interface sprites	Panji
May 13th, 2021	Finishing the program	Panji
May 14th, 2021	Continuing the report	Panji, Arya, Ichsan
May 15th, 2021	Continuing the report	Panji
May 16th, 2021	Finishing the final report	Panji

- Write a summary of the implementation of each requirement given in the first page. For each requirement, explain whether that requirement is fully implemented, partially implemented, or not implemented at all.
  - Starting the program (the intro animation).
    ✓ Fully implemented
  - Neon Tiger jumping to a wall.
    ✓ Fully implemented
  - Neon Tiger hanging on a wall.
    ✓ Fully implemented
  - Neon Tiger shooting Ray Splashers (while standing and hanging on a wall).
    ✓ Fully implemented
  - Neon Tiger jumping up after hanging on a wall.
    ✓ Fully implemented
  - Neon Tiger diving from a wall.
    ✓ Fully implemented
  - Neon Tiger performing a rush attack.
    ✓ Fully implemented
  - Neon Tiger blocking Megaman's projectiles (if implemented).
    ✓ Fully implemented
  - The dummy getting hit by Neon Tiger's attacks.
    ✓ Fully implemented
  - The bonus features
    ✓ Fully implemented

7 Conclusion and remarks.

- Does the program work as expected?
  - Yes, we believe so.

- If some parts of the program do not work as expected, explain why.
  - All parts of the program work as expected.

- What are your comments about this assignment?
  - Making program is fun but creating report is tiring.

Reference :

- https://docs.microsoft.com/en-us/dotnet/visual-basic/
- Class power point files
- Extra class file
- Class and extra class lecturing