ES207 SOFTWARE ASSIGNMENT

THE CONCEPT

To Successfully meet the requirements listed in the project brief, I conceptualised creating a 2-stage multiplication game for primary school children

Stage 1(60 seconds)

1. 60 second Timer Starts
2. User Selects Difficulty
3. Presented with a math problem
4. User answers Question
5. If correct, Ammos is generated
6. steps 3,4,5 repeated
7. Timer runs out

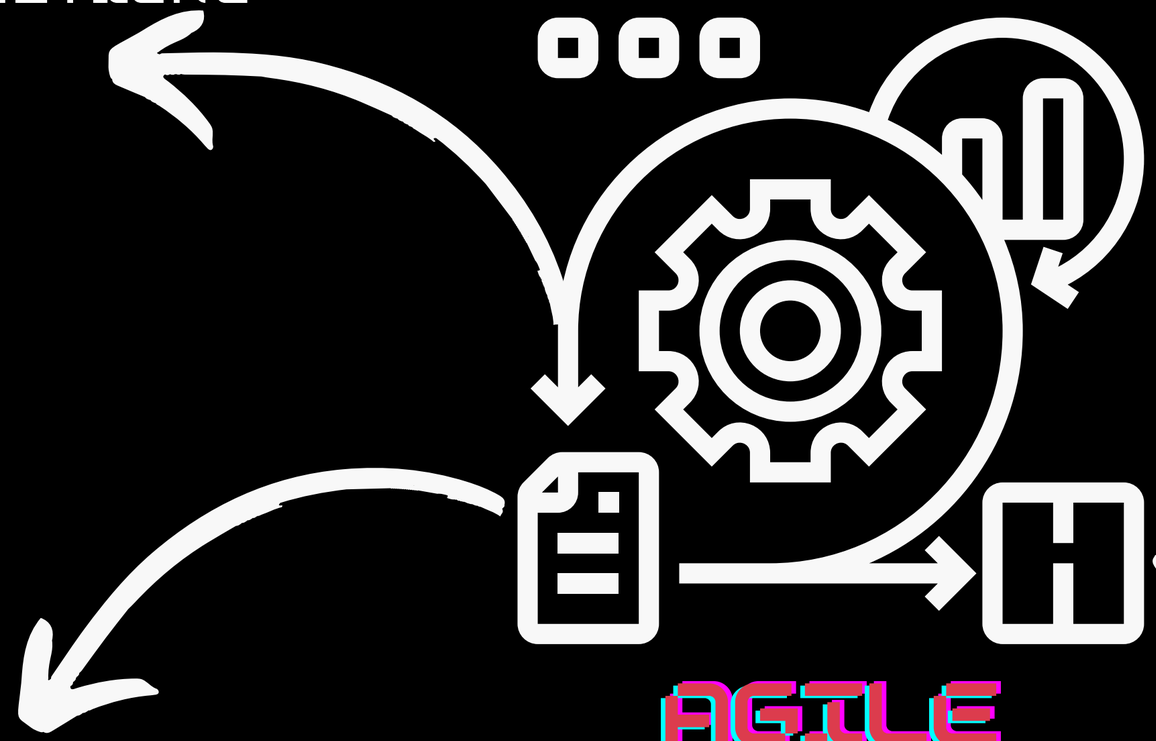
Stage 2(30 seconds)

1. Galactic attack under way
2. Users ammo level depends on stage 1
3. Able to exchange some ammunition for a galactic Bomb
4. shoot down incoming meteors
5. Ensure that Meteors remain in safe zone

DEVELOPMENT APPROACH

INDIVIDUALS AND INTERACTIONS

WORKING SOFTWARE



AGILE

CUSTOMER COLLABORATION

RESPONDING TO CHANGE

PRODUCT BACKLOG

User story	Priority(1-5)	Time Estimation(1-5)
As a user, I am shown descriptions of how the game works with rules so I understand how to play	5	1
As a user, I have the option to retry stage 1 or proceed onto stage 2 of the game	3	1
As a user, I am able to select the diffiiculty of the problems	*5*	*4*
As a user, I am able to select to play with or without sound	1	1
As a user, I am able to skip questions that are too difficult	3	2
As a user, I am able to play the game again if I wish	4	1
As a user, I am able to press a button to exhchange my bullets for bombs providing me with more options	2	5
As a user, I am shown the time remaining throughout the game	5	4
As a user, I am able to clearly see the ammunition I have generated in game	5	2
As a user, I am able to identify when the attempt is close to finishing as time remaining will become red	2	1
As a user, I can observe the total number of bombs and laser bullets I have remaining	2	3
As a user, I can clearly see the critical zone which I must protect	3	1

User story	Priority(1-5)	Time Estimation(1-5)
As a user, I am able to see the number of lives that I have remaining	2	3
As a user, I am given a location where I can type in my answer	5	1
As a user, I am able to utilise a range of push buttons and key presses to play the game	5	5
As a user, I want new questions to be generated upon correctly answering a previous problem	5	3
As a user, I am able to toggle and switch difficulty throughout the duration of the game	5	3
As a user, I am able to skip questions that are too difficult	5	2
As a user, I can directly see my missiiles being shot towards the target	3	4
As a user, I can see the asteroids being destroyed when they are blasted with lasers	2	4
As a user, I want to see whether my answer is correct or incorrect	1	1
As a user, I can easily understand that this is a space shooting game	3	3
As a user, I can see and control the spacecraft within the game	1	3

SCRUM SPINTS

Upon developing a product backlog of activities, I then categorized the following tasks into a set of scrum sprints which are:

Basic UI

- Ensuring that the game is intuitive and easy to follow
- Creating Buttons to press such as return, retry, proceed etc.

Game Constraints

- Timer
- Alternating Difficuty in the game
- Number of lives within the game
- Considering negative marking

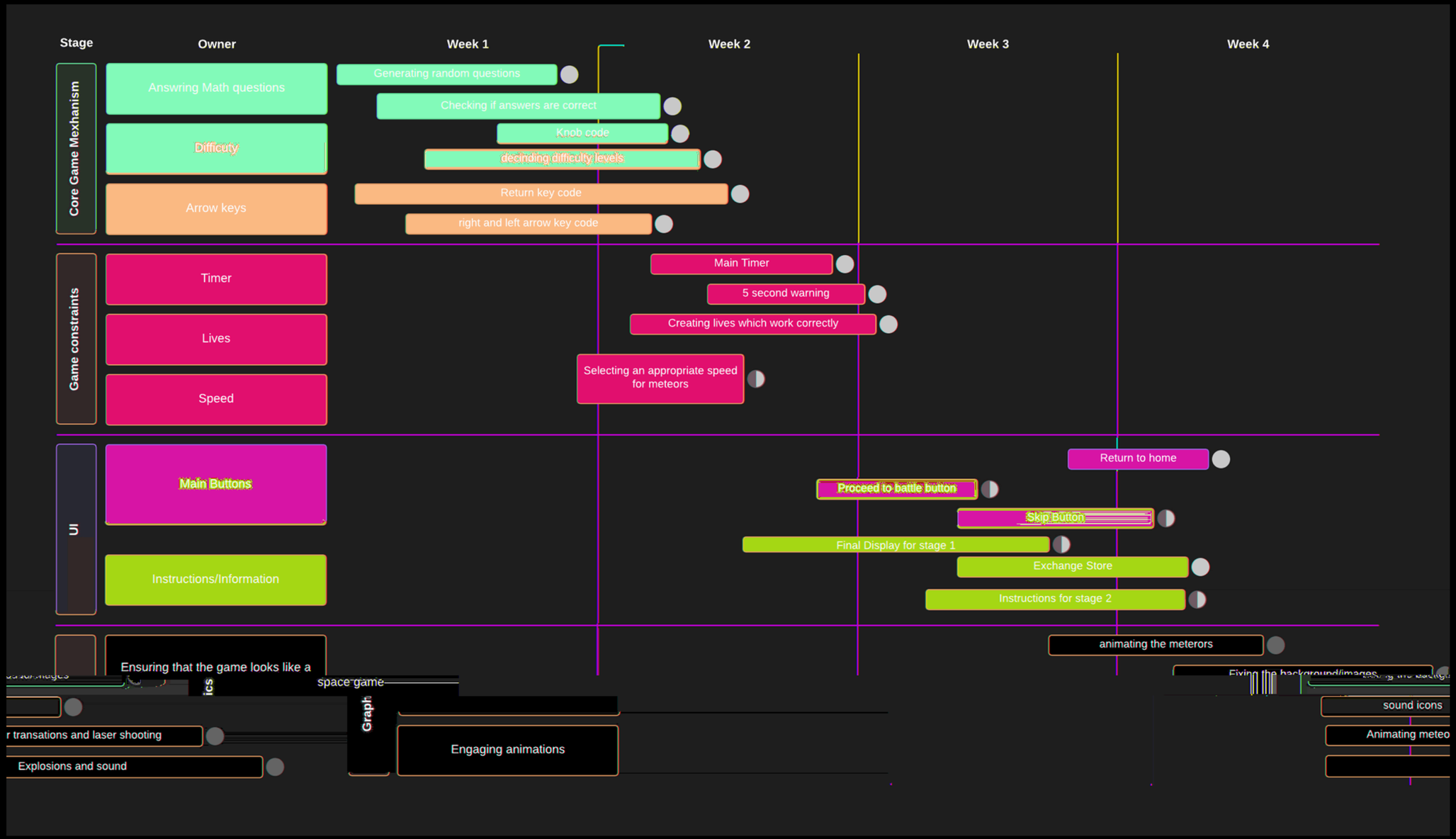
Game Mechanism

- This section focuses on the key way in which game is played:
- Being able to press keys to play game(ie=.e arrow and enter buttons)
- Being able to toggle the difficulty

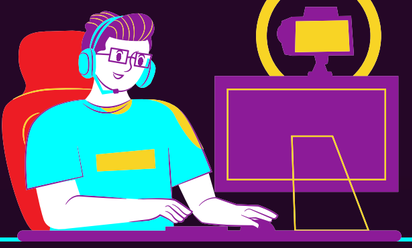
Graphics

- Ensuring that the game looks like a space game
- animating meteors and meteor explosion
- creating a display which is easy to understand

GANTT CHART


Stage
Owner
Week 1
Week 2
Week 3
Week 4
Core Game Mexhanism
Answering Math questions
Generating random questions
Checking if answers are correct
Difficuty
Knob code
decinding difficulty levels
Arrow keys
Return key code
right and left arrow key code
Game constraints
Timer
Main Timer
5 second warning
Lives
Creating lives which work correctly
Speed
Selecting an appropriate speed for meteors
UI
Main Buttons
Return to home
Proceed to battle button
Skip Button
Final Display for stage 1
Instructions/Information
Exchange Store
Instructions for stage 2
animating the meterors
Ensuring that the game looks like a space game
Graph
Engaging animations
sound icons
r transations and laser shooting
Explosions and sound

DEMO



KEY CODE(STAGE 1)

- TIMER
- ANSWER CHECKING
- ADDITIONAL FEATURES

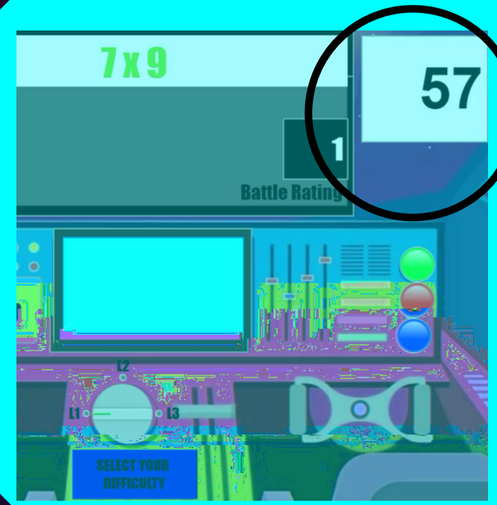
TIMER

```
function StartButtonPushed(app, event)
timelimit=60;
timesup=0;
timerstart=tic;
while~timesup
    pause(0.1 );
    timepassed=toc(timerstart);
    app.Timer.Value=round(timelimit-timepassed);
    timesup=app.Timer.Value<=0;
if ~timesup
else
    app.CheckButton.Enable='off';
    app.Retry.Visible='on';
    app.Proceed.Visible='on';

    app.BulletsFinalCount.Visible='on';
    app.BulletsFinalCount.Value=app.Bulletcount.Value;
    app.FinalTextDisplay.Visible='on';
    app.FinalBattleRatingLabel.Visible='on';
    app.AnswerBox.Enable='off';
    app.randbluebox.Visible='on';
    app.randbluebox.Value=' ';
    app.Background.Visible='on';
    app.Battleratingmessage.Visible='on';
end
```

In order to create a timer, I utilised the '**tic**' and '**toc**' functions which are already available on matlab:

- 'tic' essentially starts the timer and takes record of the current time
- 'toc' records the time which has passed since the timer has started
- Thus by utilising the following functions I was able to successfully create and display a countdown timer



ADDITIONAL FEATURES

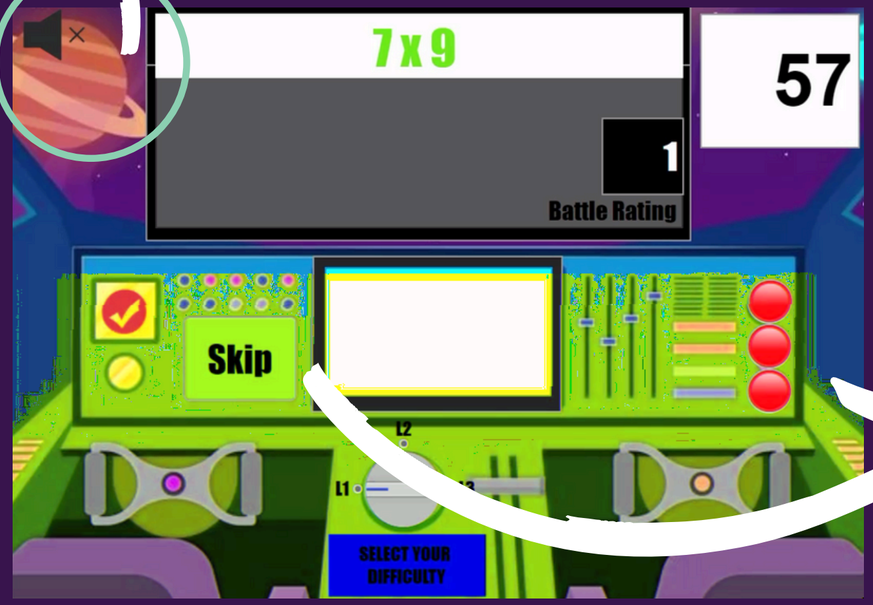
SOUND

```
function SoundOnImageClicked(app, event)
    app.SoundOn.Visible=['off']
    app.NoSound.Visible='on'
    clear sound
end

% Image clicked function: NoSound
function NoSoundImageClicked(app, event)
    app.SoundOn.Visible=['on']
    app.NoSound.Visible='off'
    [y,Fs]=audioread("techno-future-drone-main-9724.mp3");
    sound(y,Fs)

end
```

By utilizing the image clicked callback available on matlab,I was able to toggle the music on and off throughout



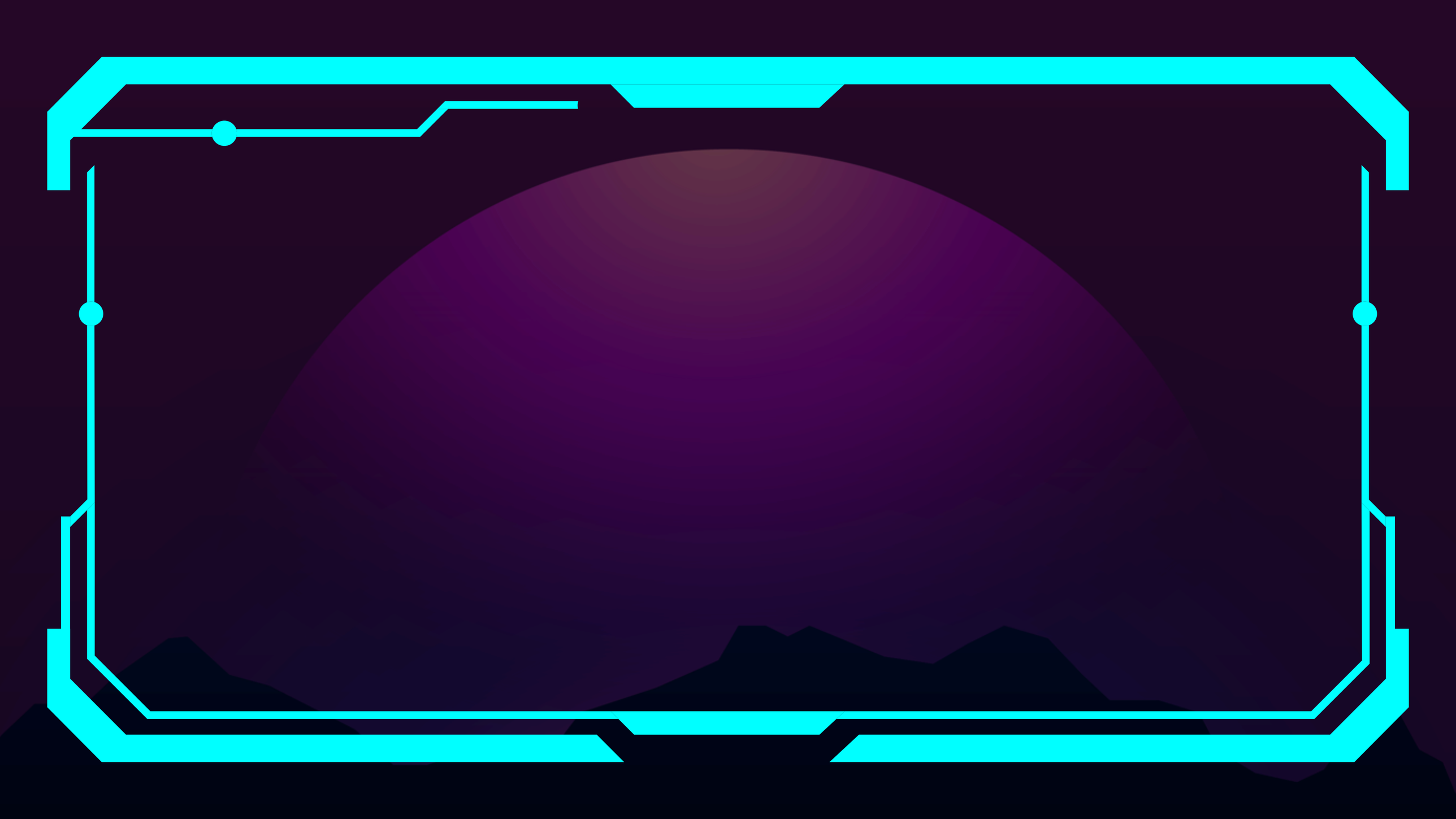
SKIP

```
function SkipButtonPushed(app, event)
if app.Knob.Value=='L1'
            x=randi(9);
            y=randi(9);
            z=num2str(x);
            q=num2str(y);
            app.ans=x.*y;
            qtext=[z ' x ' q ' ']
            app.x8Label.Text=qtext
elseif app.Knob.Value=='L2'
    x=randi(14);
            y=randi(14);
            app.ans=x.*y;
            z=num2str(x);
            q=num2str(y);
            qtext=[z ' x ' q ' ']
            app.x8Label.Text=qtext
else
    x=randi(20);
            y=randi(20);
             z=num2str(x);
            q=num2str(y);
            app.ans=x.*y;
            qtext=[z ' x ' q ' ']
                app.x8Label.Text=qtext
end
figure(app.UIFigure)
        end
    end
```

This code is essentially identical to question generatiion

KEY CODE(STAGE 2)

- DATA TRANSFER
- TIMER/METEOR ATTACK
- KEY PRESSING
- LIVES



KEY PRESSING

```
function UIFigureKeyPress(app, event)
if app.Timer.Value>=0.5&app.BulletsEditField.Value>=1;

    key = event.Key;
laserup=[0 75 0 0];
cutoff=[470 0 0 0];
cannonposition=app.Spaceship.Position;
location=cannonposition(1);
switch key
    case 'rightarrow'

        app.Spaceship.Position=app.Spaceship.Position+[200 0 0 0];
        %app.laser.Position=app.laser.Position+[200 0 0 0]
        app.laser.Position=app.Spaceship.Position;
        if location>600;
            app.Spaceship.Position=[50 25 100 100];
            app.laser.Position=[50 25 100 100];

        end
    case 'leftarrow'

        app.Spaceship.Position=app.Spaceship.Position-[200 0 0 0];
        %app.laser.Position=app.laser.Position-[200 0 0 0]
        app.laser.Position=app.Spaceship.Position;
        if location<80
        app.Spaceship.Position=[650 25 100 100];
        app.laser.Position=[650 25 100 100];
    end
```

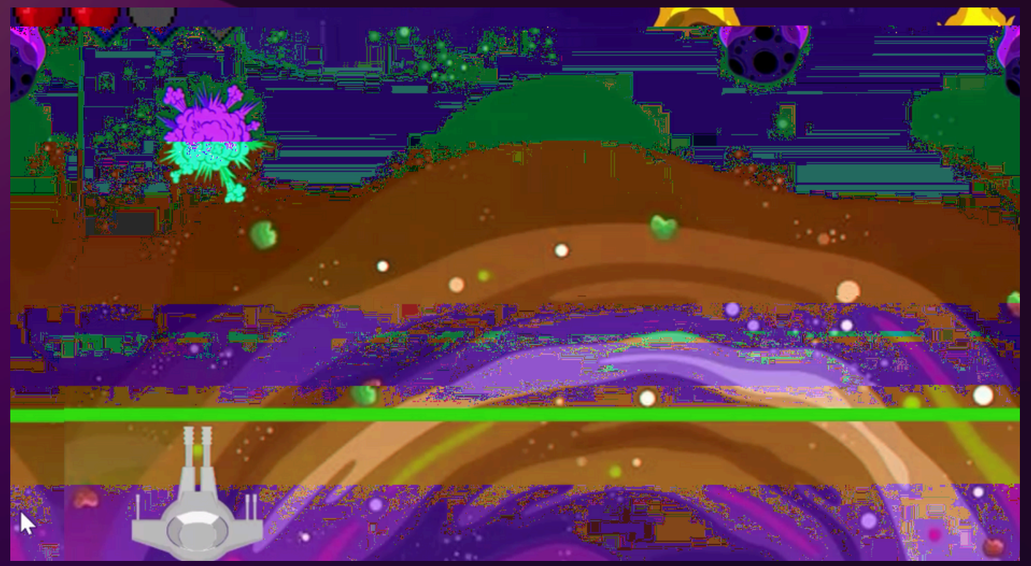
In the following sections, i used the key press functions in order to control the spaceship. Much of the code is moving images based on certain actions. In the case of shooting lasers, The code begins by comparing which row the spaceship is on and the height of the correspinding meteor

```
function UIFigureKeyPress(app, event)
case 'return'
app.laser.Position(1)==app.Fireball1.Position(1)&app.laser.Position(2)<=app.Fireball1.Position(2)
   app.BulletsEditField.Value=app.BulletsEditField.Value-1
   app.laser.Visible='on'
   app.laser.Position=app.laser.Position+laserup;
   pause(0.03)
   app.laser.Position=app.Fireball1.Position-[0 30 0 0];
   pause(0.03)
   app.laser.Position=app.Fireball1.Position-[0 5 0 0];
pause(0.03)
app.laser.Position=app.Fireball1.Position;
pause(0.03)
[y,Fs]=audioread("blaster-2-81267.mp3");
sound(y,Fs)
app.Fireball1.Visible='off';
app.explo1.Position=app.Fireball1.Position;
app.explo1.Visible='on';
app.laser.Visible='off';

pause(0.1)
app.explo1.Visible='off';
app.Fireball1.Position=[50 500 100 100];
app.Fireball1.Visible='on';
app.laser.Position=app.Spaceship.Position
```

TIMER/METEOR ATTACK

```
function STARTButtonPushed2(app, event)
timelimit=30;
timesup=0;
y=[0 1 0 0];
timerstart=tic;
while ~timesup && app.deadnesscounter.Value<3
pause(0.035);
timepassed=toc(timerstart);
app.Timer.Value=round(timelimit-timepassed);
    app.Fireball1.Position=app.Fireball1.Position-[0 5 0 0];
    app.Fireball2.Position=app.Fireball2.Position-[0 5 0 0];
    app.Fireball3.Position=app.Fireball3.Position-[0 5 0 0];
    app.Fireball4.Position=app.Fireball4.Position-[0 5 0 0]
timesup=app.Timer.Value<=0;
```



The code for this is very similar to stage 1, however I also added some code to simulate the meteors going down. Originally, I was having errors as I was attempting to run 2 separate for loops however, I later discovered that matlab was single threaded so it wasn't possible to run 2 loops. To further improve, I believe there is some software available on MATLAB that can directly resolve this rather than putting both of them in the same loop.

THANK YOU