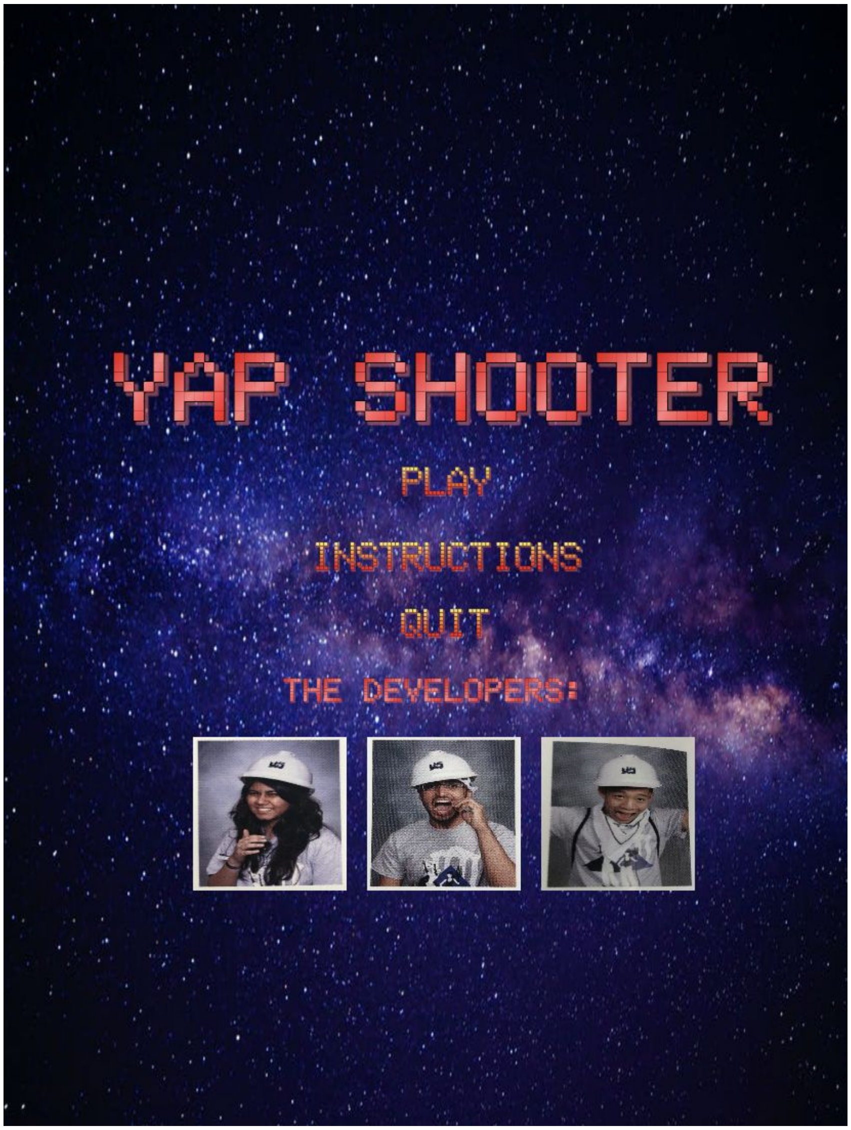


By: **Y**usra Irfan, **A**bdulaziz Chalya, **P**hillip Truong

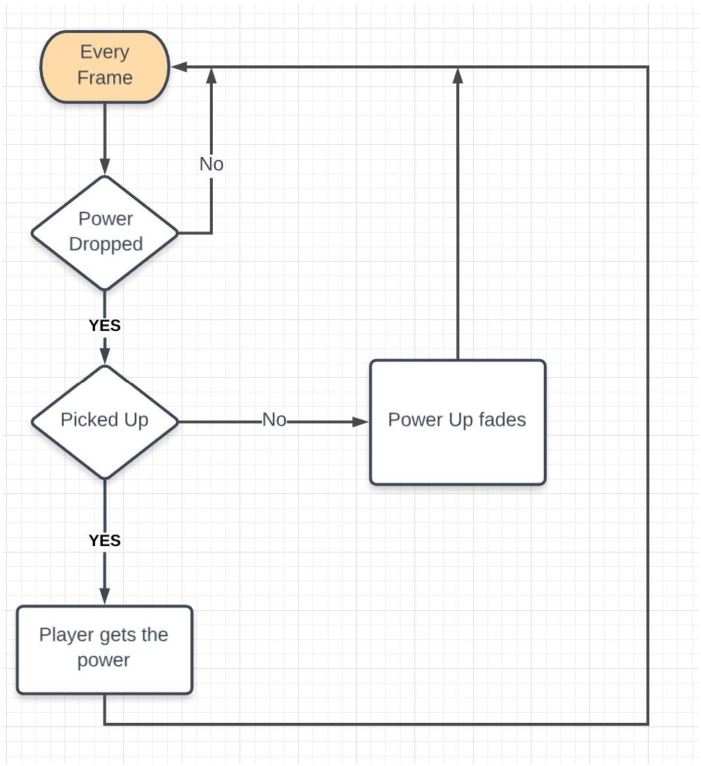
Features:

- Power Ups
- Levels
- Weapon that destroys all enemies - Nuke
- Game and sound effects
- Meteor
- Menu + Instructions
- Enemy Damage - Colour Flashing

Power Ups (3-Types):

- Using a 3D text object we created letters to indicate what type of power up (ie. S for shield, M for movement speed, and A for attack speed).
  - Added a cube child that rotates and moves.
  - Colours were also adjusted for to reflect the power up projectile colour.
- Enemies randomly drop power ups based on a percentage outlined within enemy class.
- Power ups are instantiated when an enemy dies within the enemy class.
- Power ups can be picked up by the hero (only the hero). This is outlined within the hero script.
- Shield
  - Can have up to 4 levels of shield maximum
  - Shield block hits from either enemies or meteorite
- Movement Speed
  - Increases hero speed for faster movements
  - No maximum because game gets faster and harder when hero moves faster
- Attack Speed
  - Increases the attack speed of all guns (gun & blaster) or in our game the delay between shots is lowered
  - The minimal delay between shots is 0.15 seconds (will only go down till 0.15 seconds)

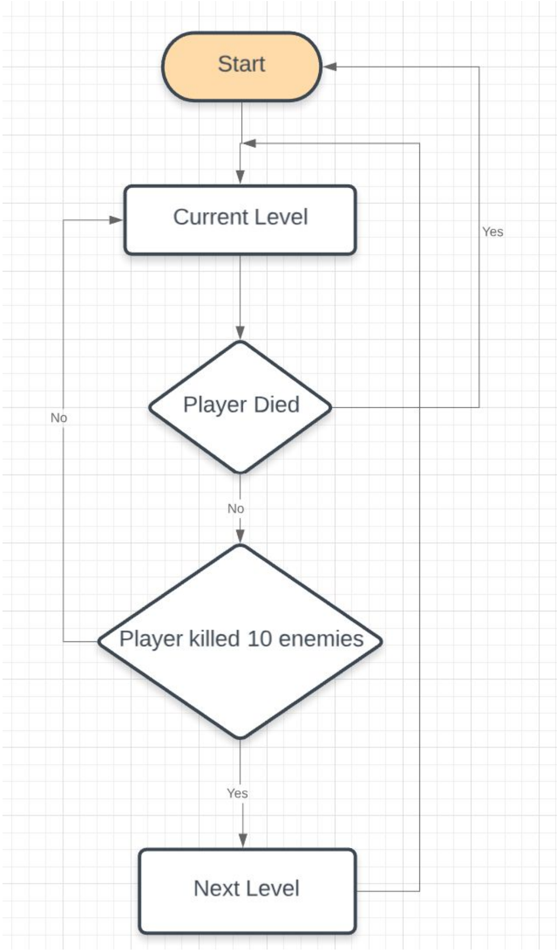
Flow Diagram - Power Ups:



Difficulty Levels:

- After destroying every 10 enemies, the game enters the next level
- Leveling up increases the speed of the enemies and spawn rate
- This process is never ending, so essentially the game is played until it becomes too fast for the player, akin to tetris
- Also works in conjunction with the meteorite feature
- The key components are the Main and Enemy scripts
- The Enemy script keeps track of the enemies destroyed, and calls the nextLevel method in Main after reaching 10 enemies, it also increases the speed of enemies by 30%
- The nextLevel method in Main increments the level, plays an audio clip, increases spawn per second by 25%, and calls the ShowLevelText method
- The ShowLevelText instantiates the Text prefab which displays the new level the player has reached
- The Text prefab automatically fades away in 2.5 seconds after being created because of methods in the prefab’s Awake method

Flow Diagram - Levels:

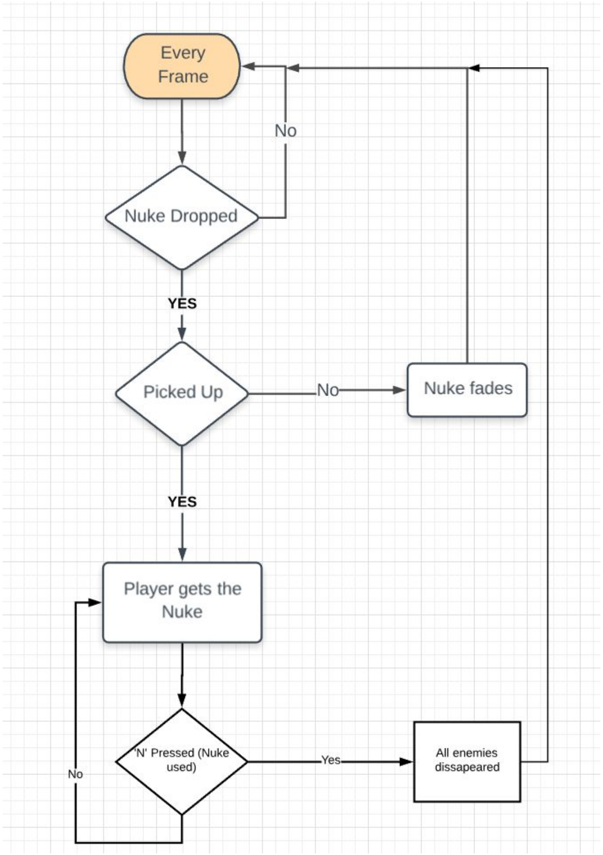


Gun that destroys all enemies on-screen - Nuke:

- This gun is very rare and only accessible by a rare enemy drop (10% chance as a drop)
- The ship can only carry one of these guns at a time
- The use of the gun instantly wipes out every enemy on the screen
- The gun ignores meteorites if they're present on the screen
- There's visual and audio feedback when the gun is used
- To use the gun, it's special key has to be pressed, the 'N' key, not spacebar
- The weapon is dropped as a power up
- When the weapon is picked up, it sets the activeNuke bool as true in the Hero script
- Anytime the key 'X' is pressed, launchNuke is called in Hero
- launchNuke only does something if activeNuke is true, otherwise it doesn't do anything
- When activeNuke is true, and launchNuke is called, all GameObjects that have an "Enemy" or "Enemy_2" tag are instantly destroyed
- The nukeBlastImage method in Main is called to give the player visual feedback
- A "BOOM" sound is also played

- After completing the nuke, activeNuke is set to false
- The nukeBlastImage method Instantiates a white image on top of the screen which fades away in 1 second

Flow Diagram - Nuke:



Game Sounds and Sound Effects:

Added audio clips and audio sources to assets and linked them to the scripts and objects. Properly connecting the sounds and effects to certain actions to enhance the game.

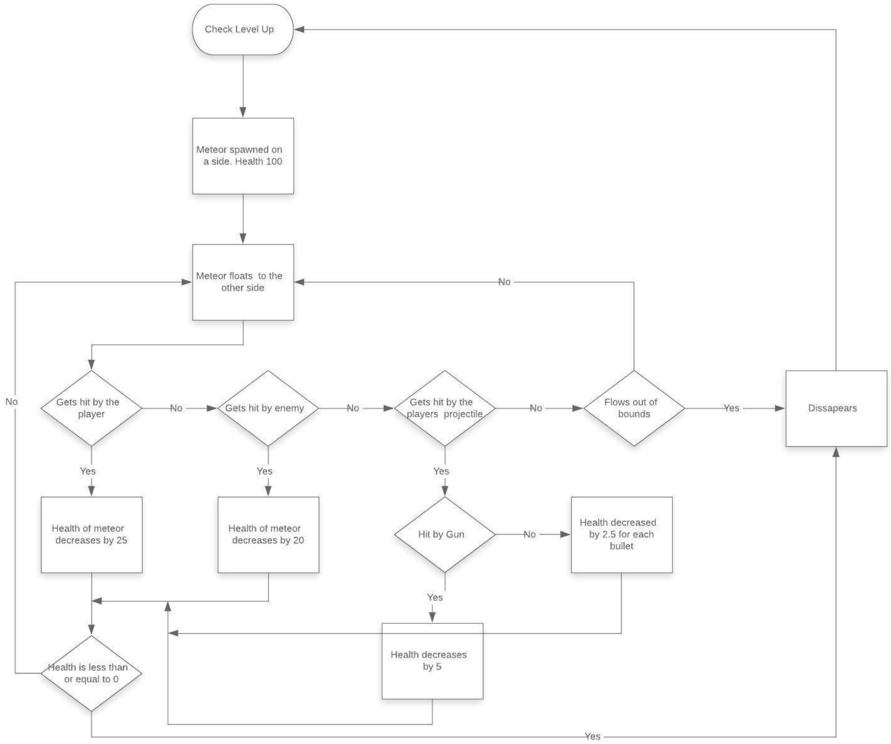
- Background music for both Menu and Game Scene
  - Attached to the main scripts and played within the start functions
- Hero losing shield.
  - Activated when the hero loses a shield level or dies
- Hero receiving power up
  - Activated when hero absorbs a power up
- Level increase (level up)
  - Activated when the level increases
- Enemy death
  - Activated when an enemy dies
- Gun weapon shooting sound
  - Activated when the gun is fired
- Blaster weapon shooting sound

    - Activated when the blaster is fired
- Nuke weapon sound
    - Activated when the nuke is fired

Meteorite:

- 3D sphere object that looks and acts like a meteor
- Blocks enemies for hero, taking damage and eventually getting destroyed
- Acts as an obstacle and will damage the hero if hero touches the meteor
- A new meteor is randomly spawned on a side of the screen when a new level begins.
- The meteor prefab is instantiated by the nextLevel method in Main
- The max and min possible position on the y axis are given by BoundsCheck
- The position on the x axis is either the left side or right side of the screen
- The spawn on left or right side is determined by a random boolean in MeteorScript
- MeteorScript is attached to the meteor prefab and manages movement
- Based on which side of the screen the meteor spawned, the script will choose the correct side of movement accordingly
- BoundsCheck is added to the meteor prefab as well, so the meteor is destroyed as it leaves the screen

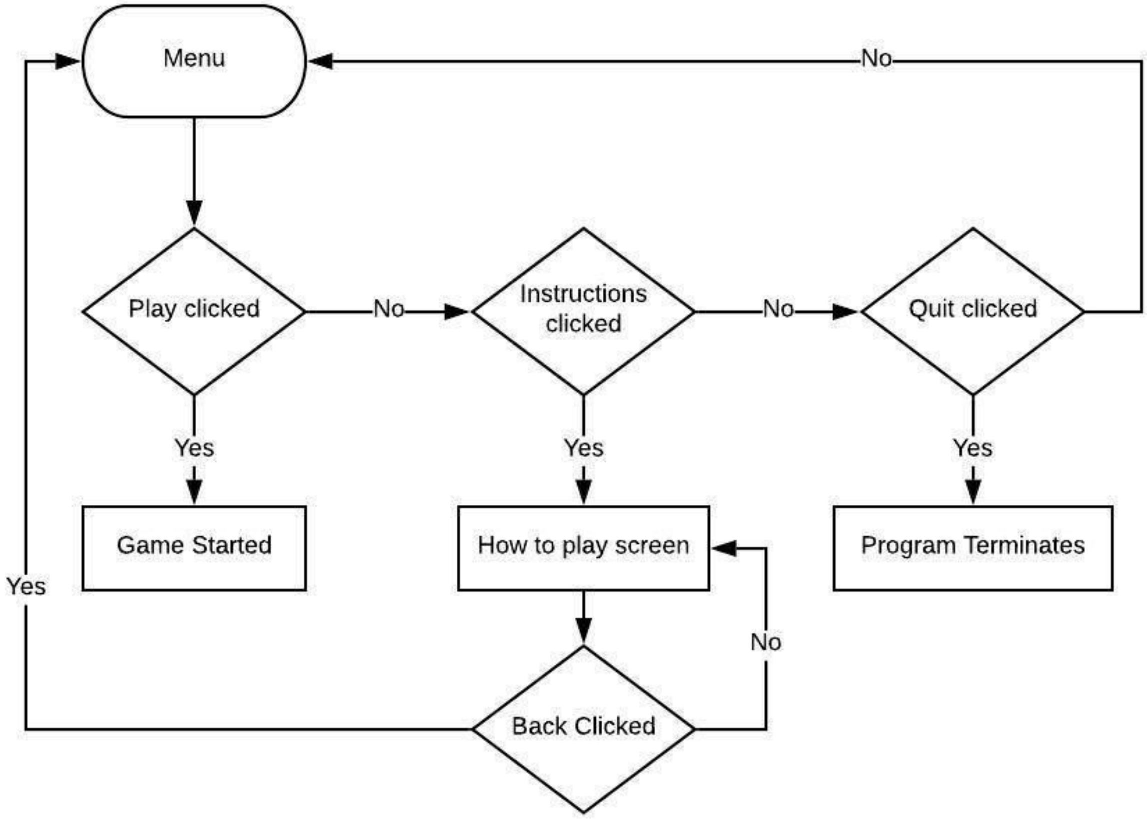
Flow Diagram - Meteorite:


Check Level Up
Meteor spawned on a side. Health 100
Meteor floats to the other side
No
No
Gets hit by the player
No
Gets hit by enemy
No
Gets hit by the players projectile
No
Flows out of bounds
Yes
Dissapears
Yes
Yes
Yes
Health of meteor decreases by 25
Health of meteor decreases by 20
Hit by Gun
No
Health decreased by 2.5 for each bullet
Yes
Health decreases by 5
Health is less than or equal to 0
Yes

Menu:

- Simple menu with galaxy background.
- Includes TextMeshPro assets to make text look nicer. Included gradients and other text effects.
- Menu includes buttons for play, instructions, and quit.
    - Play activates game scene.
    - Instructions deactivates menu in the canvas and activates the instruction menu.
        - Inside the instruction menu also has a back button that does the reverse to get back to main menu
    - Quit calls application.quit(). Also prints a debug message
- Menu also includes images of the team.
- Menu also includes nice background music.

Flow Diagram - Menu:

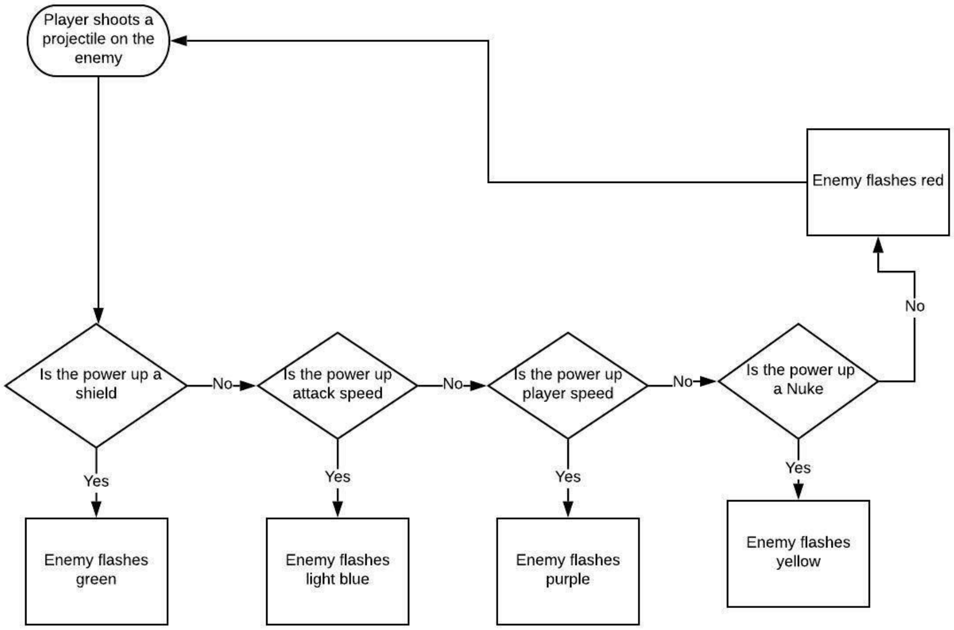


Colour Flashing:

- When an enemy is hit by a hero projectile, it flashes a colour depending on its power up state.
    - If it has no power-up, it flashes red by default
    - If it has a power-up, it flashes whatever type of power-up colour it has
- Basically allows the player to know if the enemy is going to drop a power-up or not, and specifically which one.
- Also works as a visual aid to count the number of times an enemy needs to be hit until it dies.

- When an Enemy ship collides with a HeroProjectile, ShowDamage is called
- The method goes through all the children of the Enemy ship, to access their materials
    - The array of children was initialized using the GetAllMaterials method in Utils
- Depending on what kind of drop, or no drop, the materials are set to that colour
- The showingDamage bool is set to true
- The Update method checks if showingDamage has been true for as long as damageDoneTime, if so, then it calls UnShowDamage
- UnShowDamage goes through all the children of the Enemy ship to change the materials back to their original colours.

Flow Diagram - Colour Flashing


Player shoots a projectile on the enemy
Enemy flashes red
No
Is the power up a shield
No
Is the power up attack speed
No
Is the power up player speed
No
Is the power up a Nuke
Yes
Yes
Yes
Yes
Enemy flashes green
Enemy flashes light blue
Enemy flashes purple
Enemy flashes yellow