Student ID : 1171101517	**Student Name:** Michelle Chai Mei Wei
Student ID : 1171100973	**Student Name:** Foo Fang Jee

Game Description and AI Concept

Game background

As defined in Wikipedia, shoot ‘em up games, also commonly known as shmup or STG, is a part of the shooting subgenre of video games in the action genre. One of the earliest games of this subgenre is Spacewar! dated way back in 1962. After that, in 1978, the sensational arcade game Space Invaders popularised and laid out the general template for the shoot ‘em up subgenre.

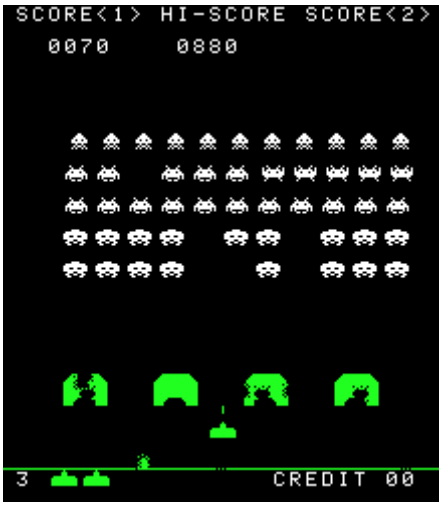

Figure 1: Gameplay of Space Invaders

This subgenre has multiple categories defined by its design elements, i.e. viewpoint and movement, such as rail shooters, scrolling shooters, bullet hell, and run and gun. Critics have different opinions as to what kind of design elements define the shoot ‘em up subgenre. However, the common elements of this subgenre are:

- Top-down perspective or side-view perspective
- Player must use ranged weapons to perform action
- Player’s avatar is usually a vehicle that is constantly under attack

- Player's goal is to shoot at anything that moves and destroy them
- Player may endure some damage before destruction
- Player need fast reaction and some kind of memorisation of the enemy's pattern

Game Proposal

Our proposed game is a shoot 'em up game with a top-down perspective, similar to Galaga, Ikaruga, Mushihimesama, Sky Force Reloaded. Our game theme is sci-fi and space. There are a few objects in this game: asteroids, turrets, enemies, boss, missile, and player's avatar.

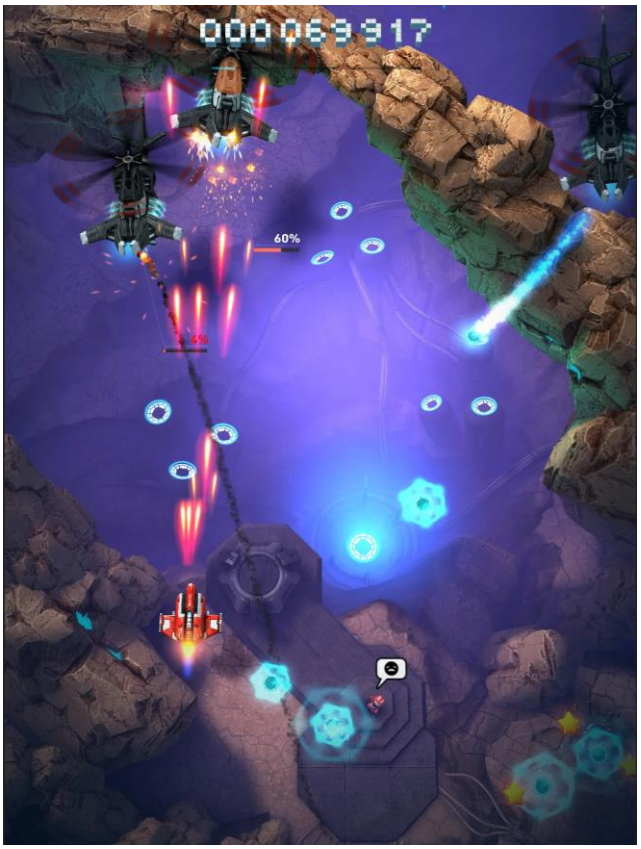

Figure 2: Gameplay of Space Force Reloaded

Win Condition

The player needs to progress through the stage without getting destroyed by the barrage of bullets from the enemies and defeat the boss at the end of the stage.

Lose Condition

The player loses all of their health bars.

Challenges

- Turrets are placed at specific positions. Turrets will attack the player once the player is within sight. Turrets can be destroyed after 10 hits and their output damage is 1 health bar per bullet.
- Asteroids will wander around on the screen. If the asteroid clashes with the player, the asteroid will be destroyed and the player loses 1 health bar. Players can destroy the asteroid by firing 1 bullet.
- Enemy Type 1 will move in a group. Enemy will attack the player once the player is within sight. Each enemy has 5 health bars. If the enemy clashes with the player, both of them will lose 1 health bar.
- Enemy Type 2 will move towards the target. Enemy will attack the player once the player is within sight. Each enemy has 7 health bars. If the enemy clashes with the player, both of them will lose 1 health bar.
- Boss is controlled by the decision-making AI, this means the boss will dodge the bullet from the player, attack the player at a suitable time and become stronger when the health bar reaches a certain amount. Boss has 50 health bars. If the boss clashes with the player, both of them will lose 1 health bar.

Rewards and Punishment

Rewards: There will be life pickup for the player to collect if the player can survive for a certain distance. The score will increase after the enemy is killed. Different enemies have different scores.

For punishment, if the player dies, they have to restart and start from the beginning of the level again.

AI Concept

1. Movement
    - Kinematic Seek

      Kinematic seek allows the game object to move towards a target. This movement algorithm is the same as the one discussed in class.
    - Dynamic Wander

      This movement algorithm allows objects to move randomly in different directions, thus changing the object's orientation.

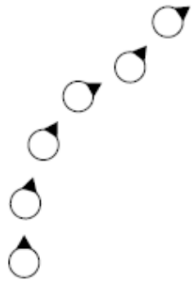

Figure 3: Movement of Dynamic Wander

According to Pandey, as compared to kinematic wandering, a few more parameters are needed for dynamic wandering, including max speed, max acceleration, wander radius, wander offset, and wander rate in order to generate a better and smoother movement. A wander point is set in a circle of wander radius, which is wander offset units forwards from the player. Wader rate refers to the frequency of orientation change. The object will align itself with the target and moves towards the direction of the target. Once the object reaches the boundaries (goes off screen), the object will be destroyed.

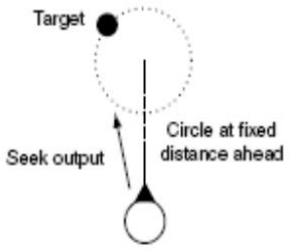

Figure 4: Dynamic Wander AI

- Pattern Movement

This movement algorithm allows objects to move in an organized manner. We plan to employ the classic Catmull-Rom spline curve to interpolate a smooth path between a few points, so that the game objects can move across the screen in a curved line.

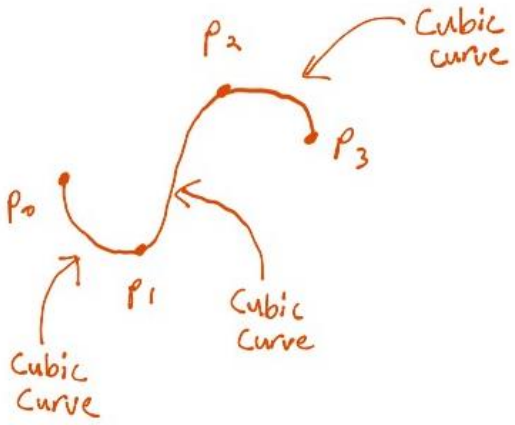

Figure 5: Catmull-Rom spline curve

2. Line of Sight

   Line of sight refers to what the game objects can see and how they should react to it. Bresenham's line scan algorithm draws an approximated straight line on the screen between two endpoints. Objects can exhibit a natural behaviour through visual limit, free sight, and Bresenham's algorithm, where lines are drawn between two objects to check whether they can see each other or not. Visual limit is the limitation of the distance of their sight, e.g. half of the window size. Free sight refers to an uncollided or unblocked line between the two objects.

   An easier way compared to drawing lines is to check whether the object is within the radius of another object through a simple calculation of their distance apart. However, this does not take into account whether there are other objects blocking their line of sight.

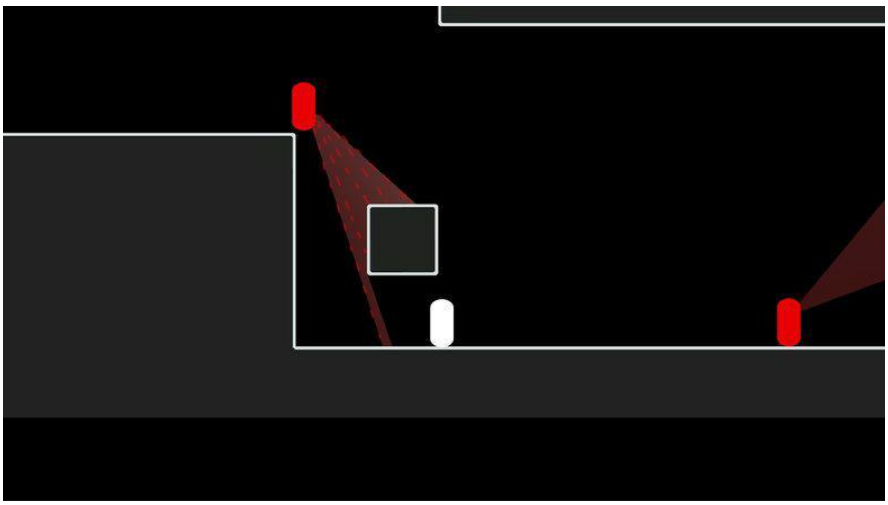

Figure 6: Line of sight (cone-shaped)

3. Pathfinding

   Pathfinding allows the game objects to move to another point using the shortest path. The common algorithm for pathfinding is Dijkstra's algorithm and A* algorithm. For this game, we plan to use A* algorithm for pathfinding. We choose to use A* algorithm is because this algorithm is optimal.

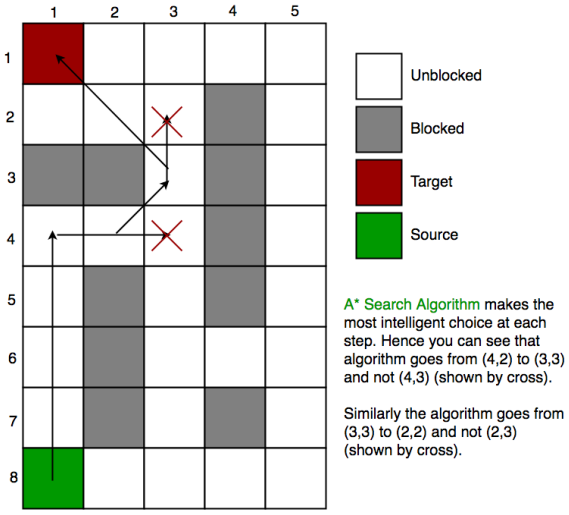

Figure 7: A algorithm*

4. Decision-making

   Decision-making algorithms allow game objects to change their behavior when there is a change of circumstances. For example, the ghosts in Pac-Man change their behavior from seeking to fleeing, once Pac-Man consumes a super pill. Finite state

machine is one of the oldest forms of game AI, but it is simple and it produces a great result when it comes to object behaviour.

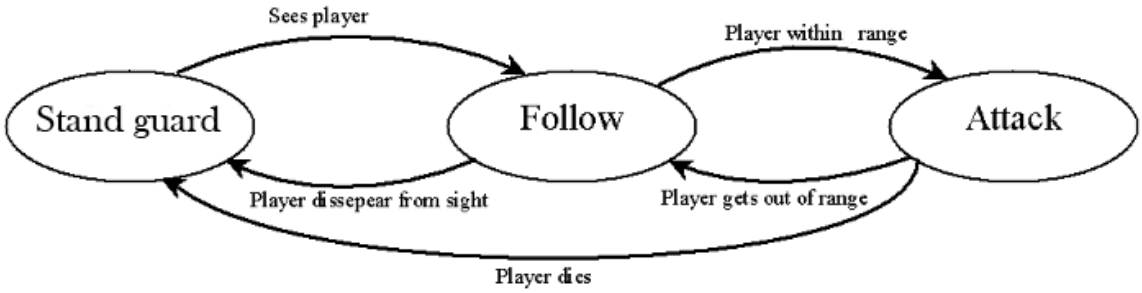

Figure 8: A simple finite state machine for turret

Assessment

The main strength of our AI concept is decision-making. This AI will be implemented on the boss (NPC) where the boss will do certain actions based on the player action and current situation. For example, if the player fires, the boss will move to the other side to dodge the bullet. If the player stands still, the boss will fire towards the player. Our boss will become stronger in terms of shooting speed if the health bar reaches a certain amount. For example, if the boss reaches 30 health bars, it will shoot faster and when the boss reaches 15 health bars, it will shoot even faster. The second strength of our AI concept is pathfinding. Pathfinding AI will be implemented on the missile where the missile will select the shortest path to chase the target and avoid all the other objects along the path.

In our opinion, this game is cool because it is a fast-paced shooting game as players need to dodge the bullets flying at them and shoot at the enemies. It will be challenging because it requires quick reactions and hand-eye coordination. Players who love challenges, shooting, and do not want to spend much time on video games would love this game. The player archetypes for this game will be the berserker as the player can run amok to shoot and destroy enemies in the game. Player also needs to survive and kill the boss in order to win the game.

The other NPCs in this game are turrets, asteroids, enemies, and the bosses. Line of sight and decision making are implemented on turrets where the turrets attack the player if the player comes within their sight. Kinematic wander will be implemented on the asteroids, where the asteroids will wander around on the screen. If the player bumps into them, their health bar will drop. There are two types of enemies. Pattern movement and line of sight will be implemented on Enemy Type 1. They will move in a group towards a direction without overlapping, shooting at the player once they are close enough. Kinematic seek and line of sight will be implemented on Enemy Type 2. This type of enemy will move towards the player and attack if the player is within the range. As for bosses, they are “smarter” than normal enemies since they have decision-making AI, where they can dodge attacks and behave differently in respect to their health bars.

Game Objects

Game Object	Health bar	Destroyable	Can shoot?	Bullet damage	Clash damage	Movement	Line of sight	Decision-making
Player's avatar	5	✓	✓ (bullet, missile)	1	1	Player controlled	Player's	✗
Missile	✗	✗	✗	3	3	Pathfinding	✗	✗
Asteroid	1	✓	✗	✗	1	Kinematic wander	✗	✗
Turret	10	✓	✓	1	1	Stationary	✓	✓
Enemy	5	✓	✓	1	1	Pattern movement	✓	✗
Enemy 2	7	✓	✓	1	1	Kinematic seek	✓	✗
Boss	50	✓	✓	1	1	Stationary, pattern movement	✓	✓

Table 1: Properties of Game Objects

Game Interface

This is the main menu for our game. Players can start playing the game after clicking on the "Start Game" button. Players also can quit the game by clicking the "Quit Game" button.

Figure 9: Main Menu

Figure 10: Gameplay Interface

Game Input

Keys	**In-game action**
↑	Move forward
↓	Move backward
←	Move left
→	Move right
Space bar	Fire bullet
Z	Fire missile

References:

1. https://en.wikipedia.org/wiki/Shoot_%27em_up
2. http://www.codenamepandey.com/movementalgo
3. https://gamedevelopment.tutsplus.com/tutorials/understanding-steering-behaviors-wander--gamedev-1624
4. http://www.cs.uu.nl/docs/vakken/b2ki/LastYear/Docs/Slides/motion-handouts.pdf
5. https://andrewhungblog.wordpress.com/2017/03/03/catmull-rom-splines-in-plain-english/
6. https://answers.unity.com/questions/46366/how-do-i-handle-pathfinding-in-shoot-em-ups.html
7. https://gamedev.stackexchange.com/questions/28204/ai-algorithm-for-avoiding-bullets-in-a-shoot-em-up-game
8. https://gamedevelopment.tutsplus.com/tutorials/build-a-stage3d-shoot-em-up-terrain-enemy-ai-and-level-data--active-11160
9. https://www.diva-portal.org/smash/get/diva2:4762/FULLTEXT01.pdf
10. https://en.wikipedia.org/wiki/Pathfinding
11. https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1063&context=itbj
12. https://stackoverflow.com/questions/54391261/pathfinding-to-a-moving-target
13. https://mukulkakroo.com/2019/01/30/ai-movement-algorithms/
14. https://www.geeksforgeeks.org/a-search-algorithm/