

Kimberly Hansuwan 100752265
Seshawn Suresh 100752981

Game Engines Bonus Challenge Blog

Game Chosen: Galaga

Galaga is an arcade shooter game where the player controls the horizontal movement of a starship, shooting bullets to eliminate enemy ships in waves and destroying bosses in an endless cycle.



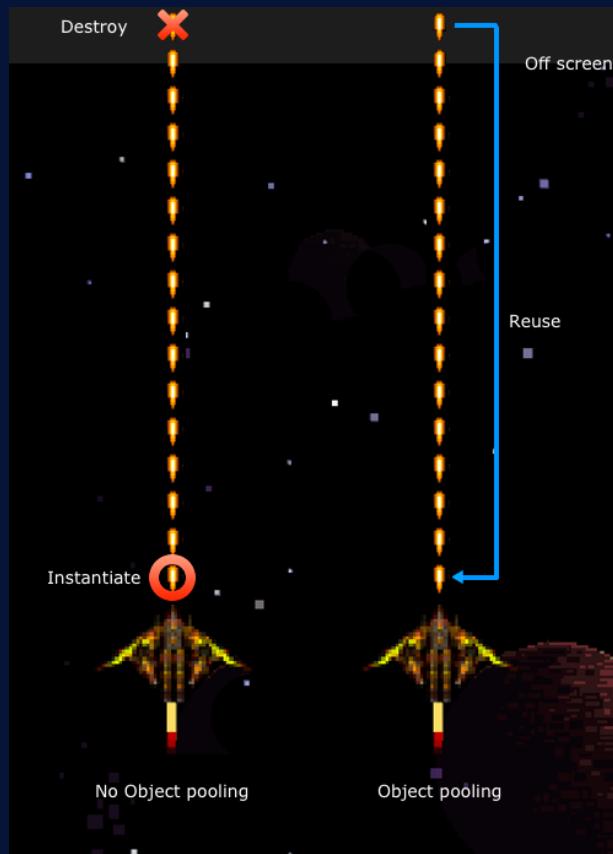
Still seeing Galaga remakes in arcade places such as Canada's Wonderland, Neb's Fun World brings back nostalgic memories. I used to play Galaga on my Gameboy and I loved the fact that it was very replayable. What I found difficult at the arcades were the button controls felt awkward and big compared to a Gameboy which was handheld and comfortable. Not to mention at the arcade places, the highscores wipes therefore, your name would never be there for years to come. It allows novice players to have a chance.



In the screenshot below, you can see the player is shooting out bullets from the starship. In theory, this means that this game is using object pooling. Object pooling in this game is to manage the amount of bullets on screen by reusing the bullets instead of storing empty destroyed objects.



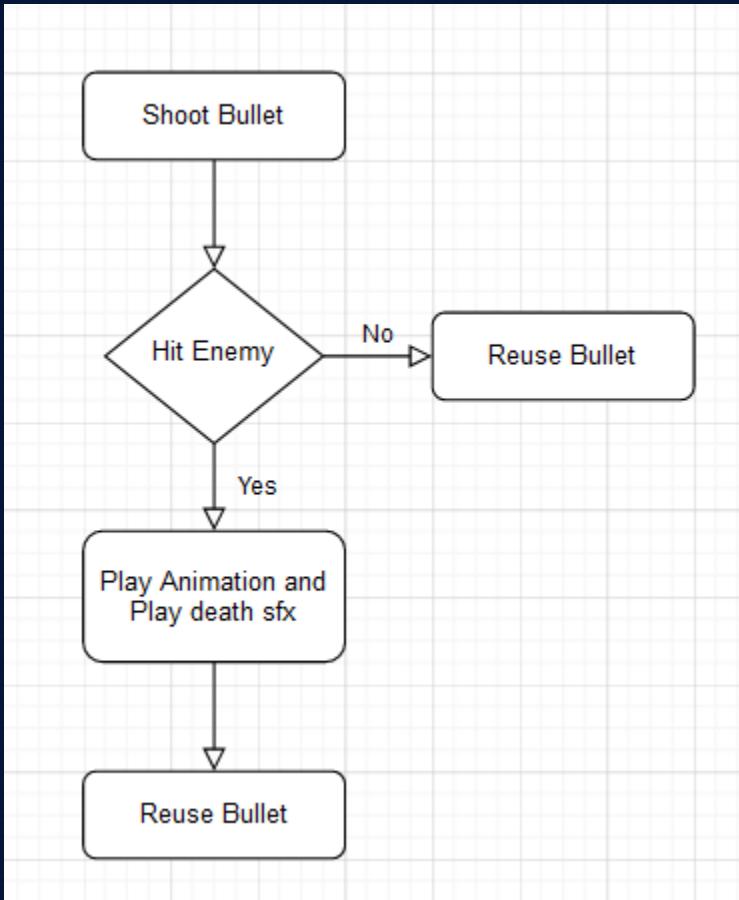
Here is an image that can visually describe what is happening.



Another design pattern that is apparent in Galaga is the observer pattern after destroying enemies. When a bullet collides with an enemy many things happen. There is a death animation, sound with the collision and the health of the enemy depleting to 0, increasing the player's score and then the game later deletes the enemy.

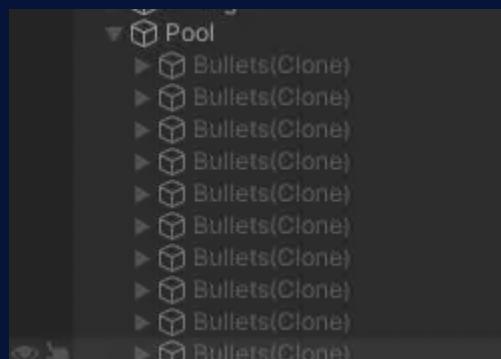


This is the object pool's flow chart:



The flow chart here shows when the player shoots a bullet what happens when it hits an enemy or if the bullet misses. There are different animations and processes that happen within the process of the bullets colliding or missing the enemy.

What we did to implement this model into the game was that we created a player that is able to shoot and when it collides with an enemy the bullet gets reused into the object pool in a queue.



GitHub: <https://github.com/kimberly-h/GameEnginesBonus>
YouTube: <https://youtu.be/Zz97GRRybwc>

Required blog parts

- Outline for the blog
- Introduction
 - 30 word video game description
 - 30 word description of your experience with the game, for example when did you play the game, which console, what did you love about, what did you dislike, what work well, what didn't, what improvements would you like to see.
- Dissection:
 - Choose a particular portion of the game and focus on identifying based on the knowledge you have gained so far a minimum of two game design patterns.
 - Dissect this game portion by including a description of what is happening supported by screenshots, animated gifs or a video.
- Explanation
 - Explain employing flow charts or diagrams such as UML what is happening in connection with the design patterns.
- Implementation (Part 2)
 - 30 word description of the assignment or low fidelity prototype being used for one design pattern
 - Design pattern demonstration
 - Design pattern explanation aligned with the chosen video game
 - 5 minute video summarizing the blog and live demonstration

If meet.google.com is sharing your screen. Stop sharing Hide

Bonus challenge worth 10 points of the total course grade Groups of 2 - timeframe: Class time

- Part 1 - five points:
 - Create a blog that analyzes a video game of your choosing by indicating what design patterns (at least 2) you identify in the game, how do you think these were implemented, and what improvements do you recommend.
 - Support your analysis with screenshots, animated gifs, and videos as necessary.
 - Additionally, use any diagrams to further explain your reasoning.
 - See this example on how to structure your blog:
<http://www.adriancourreges.com/blog/2015/11/02/gta-v-graphics-study/>
 - A clear explanation and analysis is required to get full marks
- Part 2 - five points:
 - Reuse any previous assignment or prototype a low fidelity, and I stress, low fidelity scene to demonstrate a design pattern similar to one analyzed in the game.
 - Explain how your implementation compares to that of the game you analyzed
 - A video demonstration summarizing the blog in 5 minutes
 - Clear explanations are required to get full marks.

If meet.google.com is sharing your screen. Stop sharing Hide