



Battleship Ξ xtreme

GROUP-9

Basil Issac

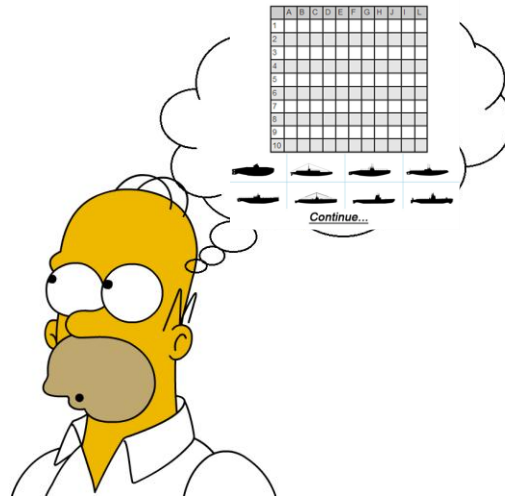
Shubham Jain

Kanishka Garg

Bradley Robinson

Brief Description:

- This game is a 2-D board game based on the conventional Battleship game, but with new features and strategies.



Starting the Game

- The user starts the program through his or her web browser.
- A menu will welcome the player along with asking for his or her information, like name, age and country for playing online multiplayer and keeping high score.
- After entering their information, another menu will appear prompting them to select “Player vs. Computer” or “Player vs. Player”.
 - If “Player vs Player”
 - Player will select to send invitation to another player to join the game.
 - Two grids will be created for each player. The lower grid will contain the player’s ships and other resources along with red “X”s for opponents hits, and the top grid will track the player’s moves against his or her opponent with red “X”s for player’s HITs and blue “O”s for player’s MISS.
 - If “Player vs Computer”
 - Player will start game with AI
 - AI will set up its board

	A	B	C	D	E	F	G	H	J	I	L
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											



Continue...

Rules:

- Players take turns
- Purchasing can be done ONLY on turn
- Use weapons accordingly
- **Victory:** The first player to sink all opposing ships(excluding Ore Refinery) wins

Opponent

	A	B	C	D	E	F	G	H	J	I	L
1											
2											
3											
4											
5				O							
6											
7											
8											
9	X										
10											

Notifications

HIT at [A,9]

Your Turn

MISS at [D,5]

Waiting for Opponent...

Player

	A	B	C	D	E	F	G	H	J	I	L
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											

Inventory



Cash

1000\$ +50\$

Select Weapon

Make Attack

<Weapon Selected>

Buy Weapon

How to make \$\$\$

- In the beginning of the game, each player is given a certain amount of money.
- A player is able to generate additional revenue through two different ways.
 - The first way a player can generate revenue is through an “Ore Refinery”.
 - Each player will start of with one
 - Same as ship (Can get hit and sunk), but cannot be last piece
 - The second method to generate revenue is to HIT the opponent’s ships
 - If opponent ship is sunk, attacker receives more cash depending on ship sunk

How to spend the \$\$\$

- The revenue generated from the previous methods can be used to make purchases on attack weapons & Defense mechanisms:
 - Attack: Giga Bomb, Mega Bomb, Homing missiles, Radar
 - Defense: Skip Turn, MoveShip, Shield
 - Each attack weapon or defensive method will cost a certain price according to its usefulness. Each weapon will have a different use, which will be described later.
 - Cost: GigaBomb>MegaBomb>HomingMissile>Radar



Bomb

Price = 100 \$



Homing Missile

Price = 200 \$



Radar

Price = 300 \$



Cash Remaining = 500 \$

Cancel

Purchase

STORE

Requirements

Functional Requirements:

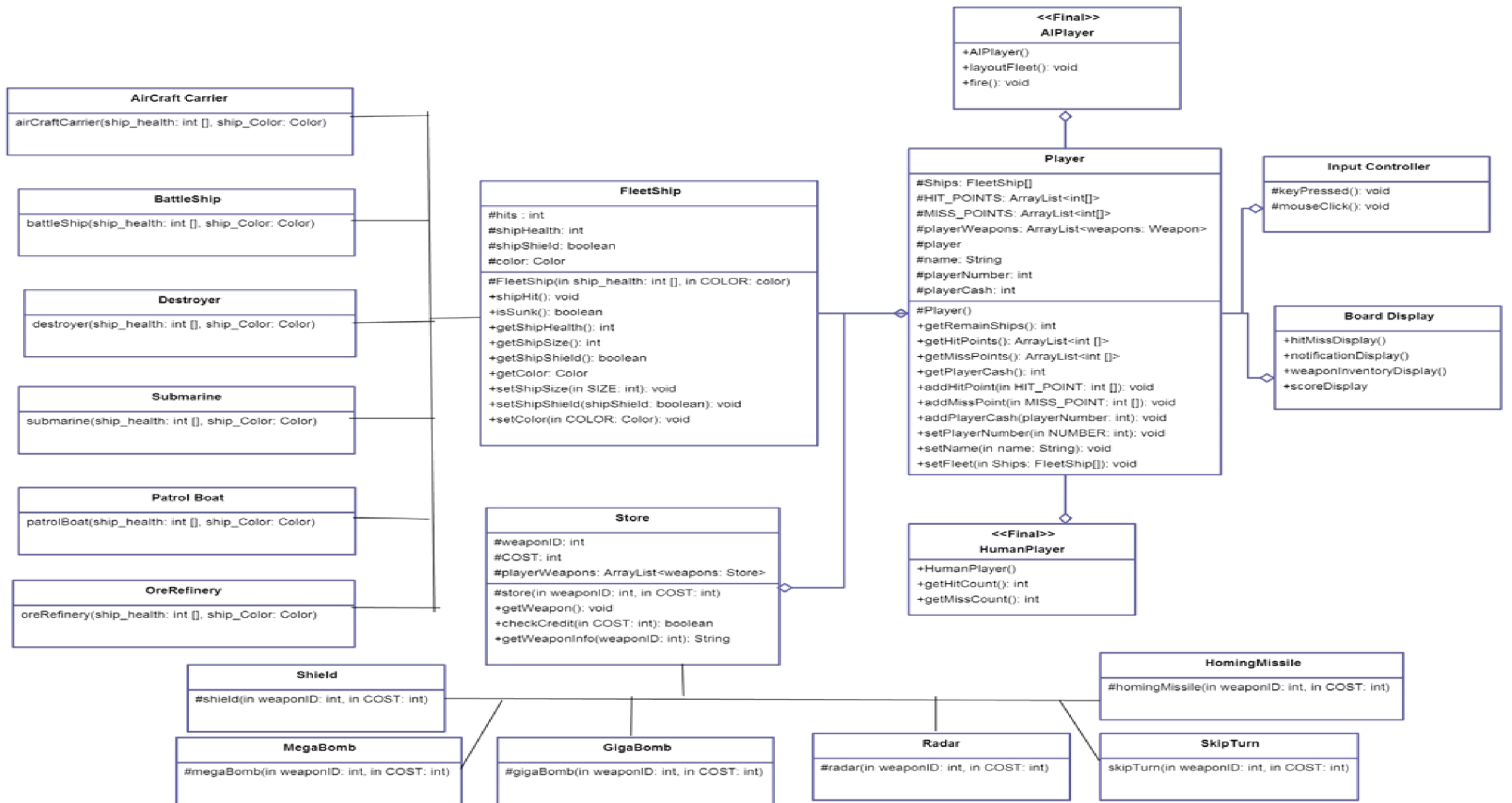
- Input through Keyboard and Mouse
- Support for Player vs. Player and Player vs. CPU
- Stable AI.
- Database keeping track of players information, scores, friends list, etc.

Nonfunctional Requirements

- Speed. Everything should run fast and smooth.
- Stable servers capable of handling a large load of users.
- Kid friendly environment.



Object Model



Design Goals

- ***Efficiency:***

- The game is to run at least 30 frames per second at all times in order to provide smoothness in the movements of the game's objects.

- ***Reliability:***

- System will be bug free and will not crash from unexpected inputs.

- ***Usability:***

- Easiness of use is a must. Anyone should be able to effectively play the game after 5 minutes of game play.

- ***Extensibility:***

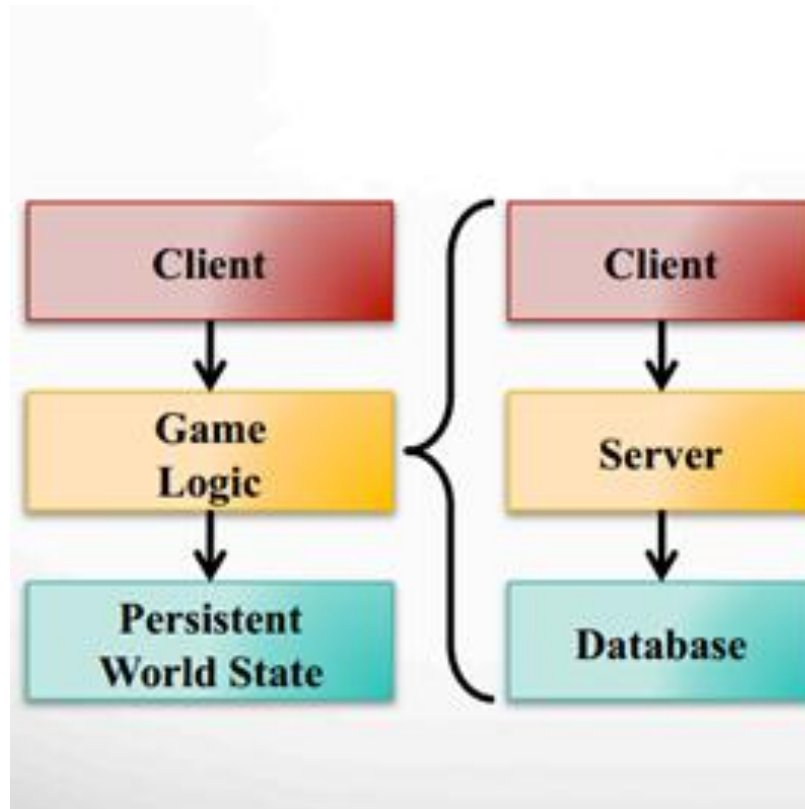
- In the initial release, game should be able to run with at least 1000 players playing at once without problems arising.

- ***Adaptability:***

- Java is the preferred programming language for Battleship Extreme as it offers more range in the amount of operating systems supporting the game.

Software Architecture (3-Tier)

- Presentation Tier
- Application logic tier
- Data Tier



- **Client**
 - Graphics and animation
 - Maybe world physics
 - No other game logic (why?)
- **Game Logic**
 - Determines success v. challenges
 - Manages NPC behavior
 - Coordinates multiple players
- **Persistent World State**
 - Stores state of game objects
 - Logs changes over time

The AI

- This game cannot have a computer that completely makes random attacks.
- Will be able to make “smarter” moves throughout the game with gathering more information.
- The AI must be able to
 - Set up the board
 - Strategically make attacks(Know HIT and MISS areas)
 - Using Probability
 - Purchase weapons (Spend Money)
 - Manage resources and Earn money

At the end of the day

- “3-D” Thinking
- Integrated with Social Media
- Community of players
- Recognition of achievements (High Scores and Medals)
- Enjoyable