CAP 3032 - Final Project Design

Jean-Pierre David
Maximiliano Ronda

Galactic Blast

Project Analysis:

Images and Sound files Included

Images used for this game are custom-made with pixel-art styling as choice. The background is large and serves as the world. With a space based game the images are largely representative of a galactic action game. The soundbites used are representative of what would occur in such science-fiction even though sound is made by vibrations through air and there is no air in space, thus space.

Sound actions include firing, travel, and game music.

Image files (all image files included are .png format)

Asteroid1
Asteroid2
background1
background2
MainMenu
SecondMenu
ThirdMenu
ExitMenu
PauseMenu
gameManual
pauseIcon
scoreIcon
gameOverIcon
planet0
planet1
planet2
planet3
planet4
planet5
planet6
stealthShip
stealthShipWithFire
tinyShip
shipSprite
Character

Sound files

asteroidcrumble.wav
asteroidcrumble2.wav
bigshot.wav
chargeShot.wav
enemyLaser.wav
enemyShot.wav
engineThrust.wav
MainTheme.mp3
primaryShot.wav
recharged.wav
space-hunter1.mp3
space-hunter2.mp3

User Interface Design and Implementation

User interface will utilize keyboard and mouse controls. The mouse will guide the direction of the ship during gameplay. Keyboard actions will be used mainly for gameplay and menu selection.

Game Design

In Space Warrior the player controls a ship class that navigates through a map, asteroids are generated at random and float through the environment at a reasonable speed. The player guides his/her ship by mouse and keyboard, thrusting in the direction to accelerate in. The goal is to destroy the asteroids and avoid the randomly generated aggressive targets. The hard level contains asteroids moving 3 times their speed.

Classes Included:

Asteroid

- Generated at start of level, when fired upon will break down into smaller asteroids
- Passive target
- Destroying adds to overall score

PlayerShip

- Has health
- Can obtain power-ups which will boost attacking capability
- Special attacks with periodic cooldowns
- Direction controlled by mouse
- Acceleration controlled by hitting D key repetitively
- Firing controlled by using F key

asteroidCluster

- Randomly generated asteroid
- Collision detection
- Images of asteroid and sounds are displayed here
- Checks when asteroid approaches in all direction

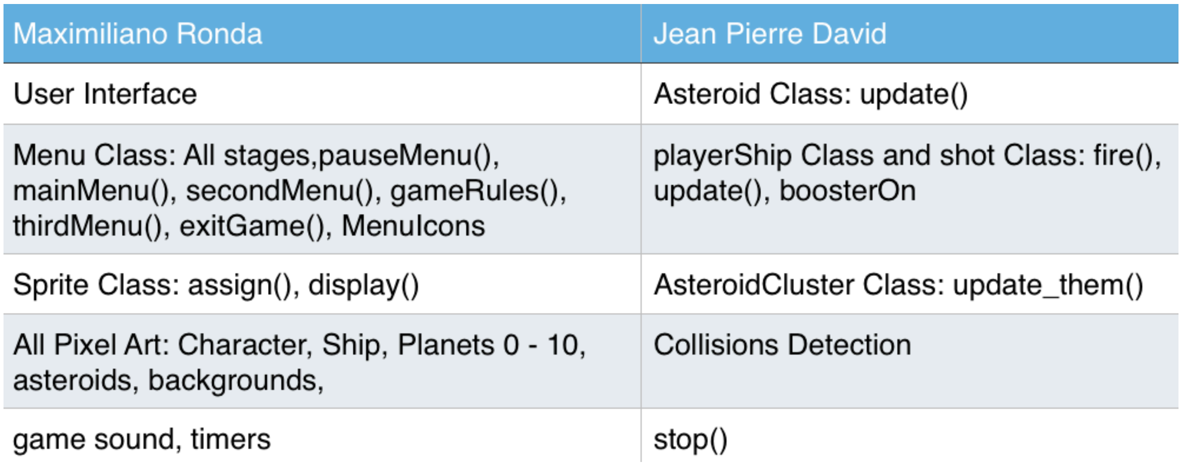
sprite

- Custom class to generate a ship sprite for main menu
- Gets number of images in sprite
- Uses the information in the image to loop through them and mimic movement
- The image uses is shipSprite and is the icon for menu selection

Menu/Interface

- Starts/Stops the game
- Maintains and displays score
- Displays and responds to interactive requests

Work Distribution



Maximiliano Ronda	Jean Pierre David
User Interface	Asteroid Class: update()
Menu Class: All stages,pauseMenu(), mainMenu(), secondMenu(), gameRules(), thirdMenu(), exitGame(), MenuIcons	playerShip Class and shot Class: fire(), update(), boosterOn
Sprite Class: assign(), display()	AsteroidCluster Class: update_them()
All Pixel Art: Character, Ship, Planets 0 - 10, asteroids, backgrounds,	Collisions Detection
game sound, timers	stop()

Difficulties

Limited difficulties encountered, most due to collision detection because asteroids are of different sizes. There were some nullPointerExceptions when entering the hard level but it was caused by Processing, not in our part. Overall, the project was smooth, we worked for several days and are happy with the results. Hope you enjoy!