

CS4303 Game Design

Practical 2

120006063



Galactic Conquests
Dream. Build. Dominate.

12/10/2015

Devising an original video game using the Processing platform.

Table of Contents

[Title](#)

[Genre](#)

[Environment](#)

[Player\(s\)](#)

[Opponent\(s\)](#)

[Single-Player Mode:](#)

[Multi-player Mode:](#)

[Rules/Mechanics](#)

[Game Flow and Transitions](#)

[Screen Mockup](#)

[Screen Description](#)

[Transition](#)

[Goals](#)

[Game Loop\(s\)](#)

[In Context](#)

[Faster Than Light \(FTL\)](#)

[Planet Racer](#)

[Platform](#)

[Physics, Artificial Intelligence, Networking and Security](#)

[Physics](#)

[Artificial Intelligence](#)

[Networking](#)

[Security](#)

[Bibliography](#)

Title

The title “Galactic Conquests” was chosen for the following three reasons:

1. Provides an accurate summary of the main goal in the game (e.g. conquer the galaxy).
2. “Galactic Conquests” is short and memorable.
3. It is a commercially-viable brand name and the easily search-engine-optimisable domain galacticconquests.com was available. (Domain has been registered.)



Figure 1: Initial logo design for Galactic Conquests

In addition to the title, an initial logo design has been created for the game. The logo is both minimalist and simplistic consisting of three key elements: Planet image, Game Title and Game Tagline. The tagline “Dream. Build. Dominate.” attempts to excite potential players by communicating the essence of the game experience: **Dream** about new spaceships and galaxies, **Build** the most powerful and intriguing fleet and attempt to **Dominate** the universe.

Genre

The key gameplay action will involve navigating a 2D solar-system in a spaceship, attempting to out-maneuver and avoid moving target while shooting enemy ships in real-time.

The controlling of a ship from a third-person perspective (where the player’s view of the game-world moves as the player’s ship moves), suggests that the genre might be a **Third-Person shooter**. However, Third-Person shooters are typically 3D.

While the gameplay takes place in real-time, it does not involve resource micromanagement but instead focuses on battlefield tactics, thereby making the game less a **Real-Time Strategy** game (often abbreviated as RTS) and more a **Real-Time Tactics** game (or RTT).

The genre of gameplay would therefore be a cross between RTT and a Third-Person shooter game.

Environment

The main gameplay will be space battles which will take place in a 2D solar-system. A star will be centre and there will be orbiting entities such as asteroids, debris, planets and moons (as shown in figure 2).

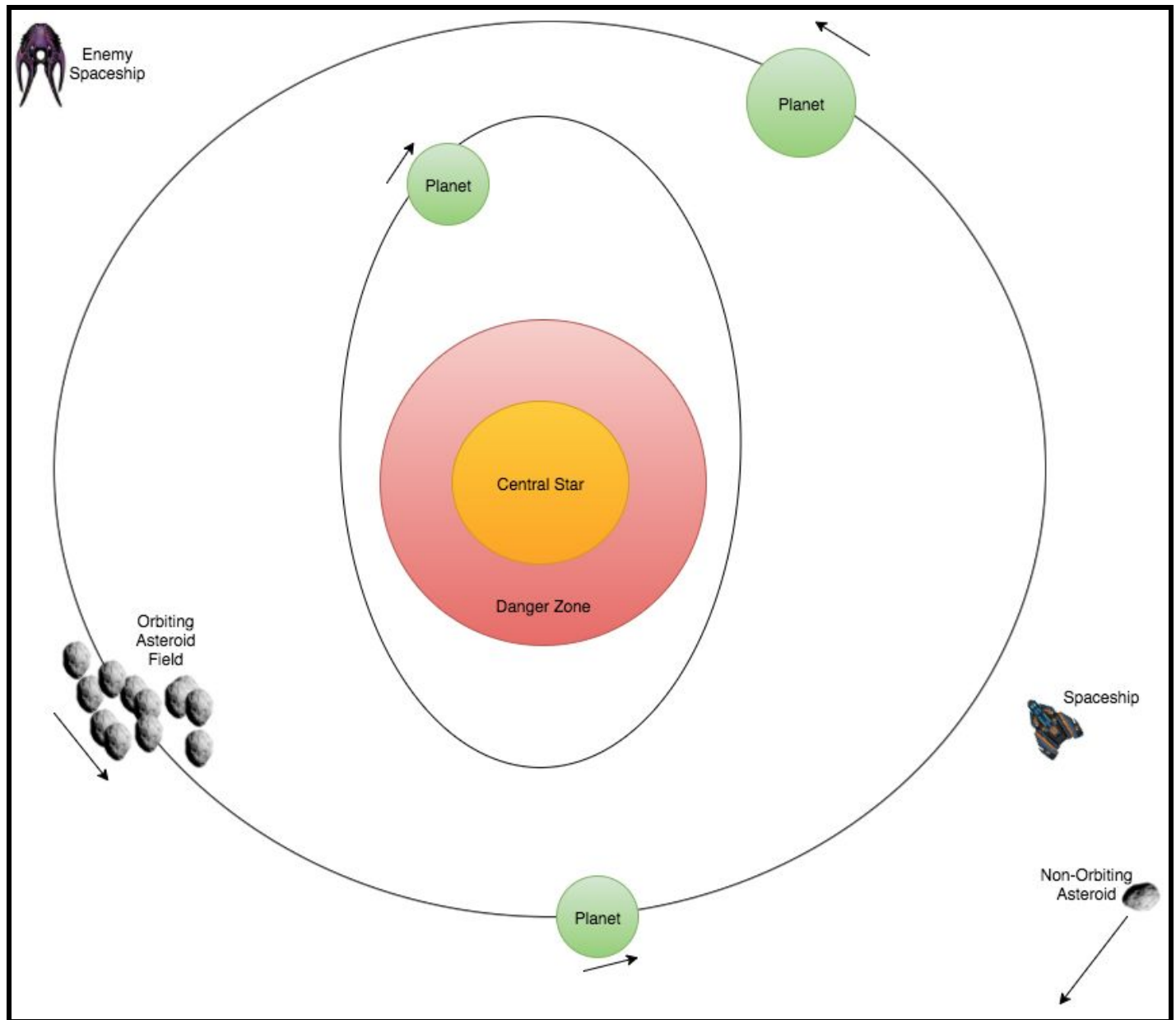


Figure 2: Rough outline of object placement in a Solar System. (Not to Scale)

Objects will be of unrealistic proportions in order to give a somewhat novel and fun feel. Planets will exert a gravitational pull on other objects. Moons will orbit planets and will exert much weaker gravitational pull than the planets.

Some asteroids will orbit, other asteroids will be affected by various gravitational pulls but will most likely pass straight through the solar system.

Objects which stray too close to the central star will be pulled into the star by its gravity and effectively destroyed. The area around the central star where the gravitational pull is exceptionally strong is denoted by the red "Danger Zone" in figure 2.

Player(s)

During space battle gameplay, each player will be the pilot of a spaceship. A player can own multiple ships in their personal fleet, however, each player must chose a single ship to control for any given battle.

A player must have a minimum of one ship is their fleet (otherwise the game is over). Below are a list of spaceships available for the player during the campaign:



Figure 3: Graphics of spaceships available to player. (Vector)

Opponent(s)

Single-Player Mode:

There will be 3 factions in the game: Villt, Qalz and Dol'eo.

In a given space battle, the player will typically have accepted a mission from one faction to fight another faction. The opponent will therefore be one or more spaceships controlled by the enemy faction.

Each faction will have their own classes of ships:

Villt Spaceship Classes:

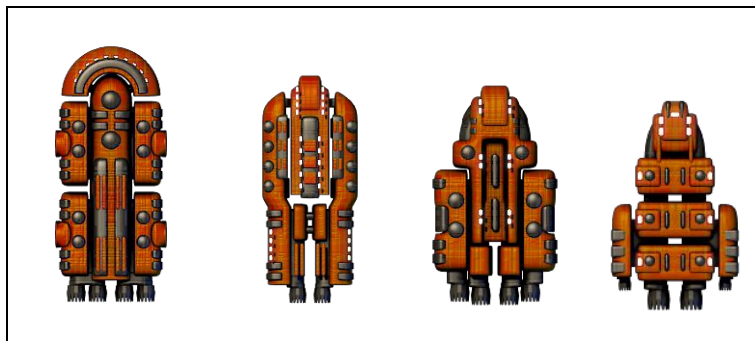


Figure 4: Graphics of Villt Spaceships. (Vector)

Qalz Spaceship Classes:

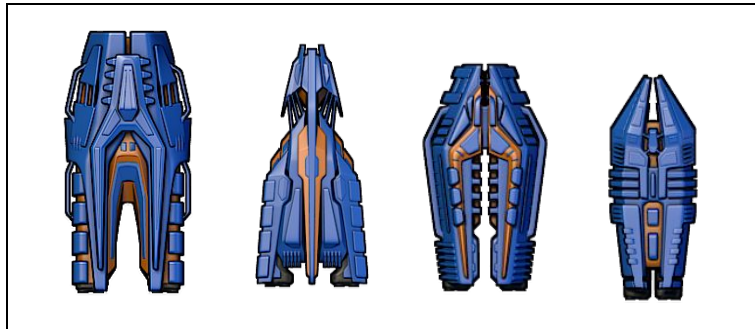


Figure 5: Graphics of Qalz Spaceships. (Vector)

Dol'eo Spaceship Classes:



Figure 6: Graphics of Dol'eo Spaceships. (Vector)

Multi-player Mode:

In multi-player mode, one player is the game host and sets up of the parameters for the game and the other others connect to this game. As multi-player is custom play, players can chose to pilot any ship of their choice (including any faction-specific ships). AI players can be added.

Opponents would therefore be represented by all the ships other than the player's ship.

A total maximum of 8 players will be allowed in any given game. This player limit thereby prevents maps becoming too crowded and networking overhead becoming too high.

Rules/Mechanics

A player may only change the game state if for the current screen, there exists an action which the player is allowed to perform and the player choses to perform the action. Available actions vary from screen to screen (see next section for more screen-specific details).


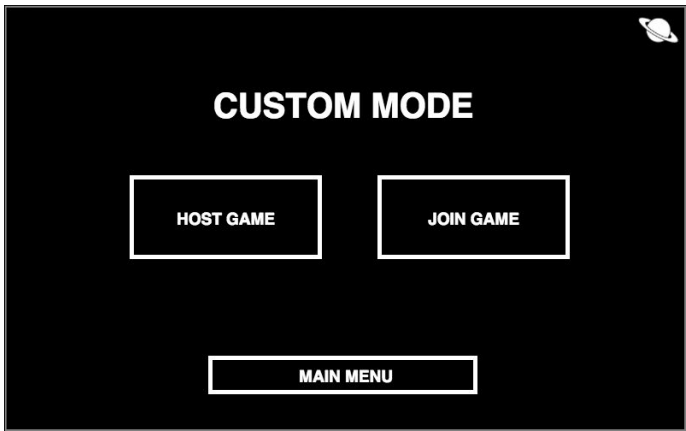
With the Fight Screen, updates to the game state are made every frame which includes updates to the environment, spaceships and other objects (such as missiles) and collision detection.

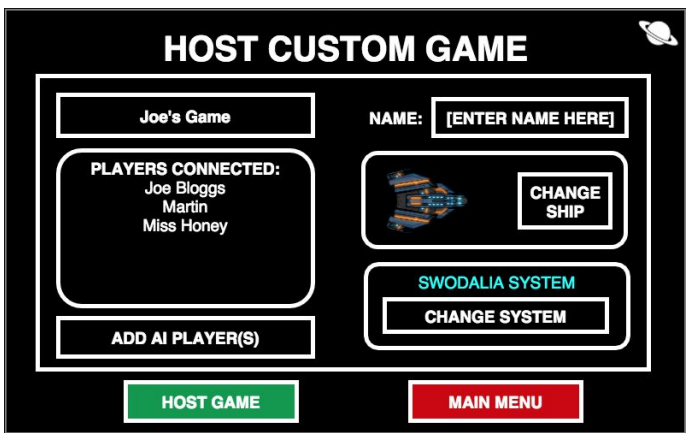
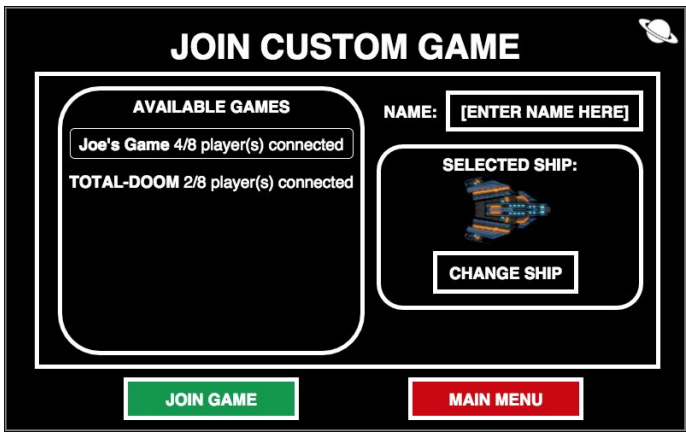

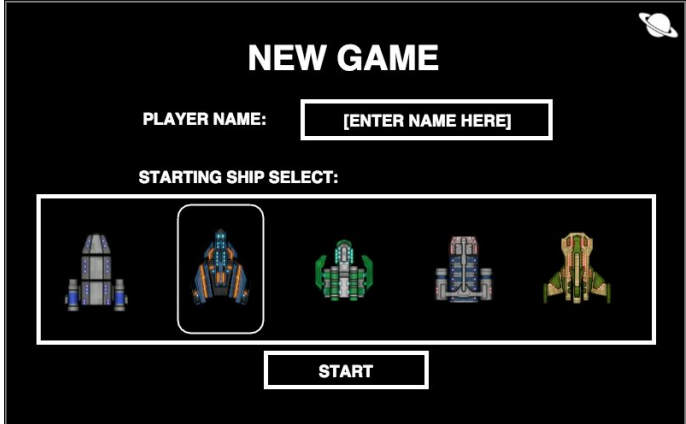
All objects will abide by the laws of physics and follow Newton’s First, Second and Third Laws.

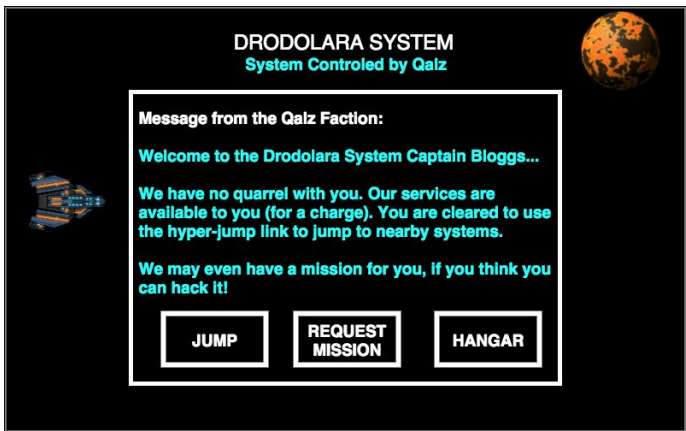
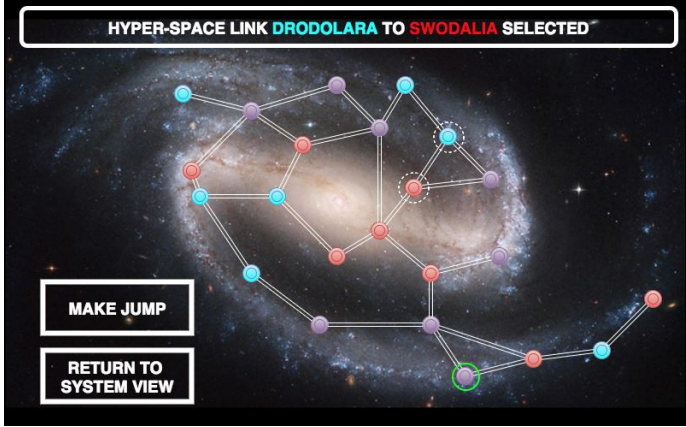
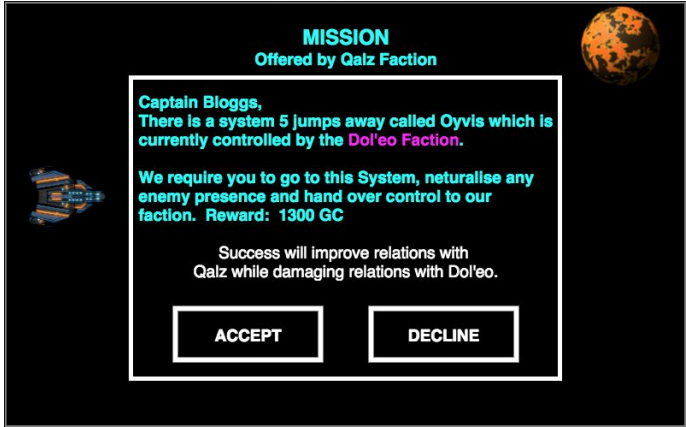
Spaceships have the ability to rotate, accelerate in the direction they're facing, fire missiles and get destroyed if their hull health falls to zero.

Game Flow and Transitions

Please note that the screen images are mockups and are only intended to visually communicate to role of each screen. Some pieces of information has been purposely omitted such as score, credits (called “Galactics Credits” or “GC” for short), current system location, reputation amongst each faction, current fleet, etc...

Screen Mockup	Screen Description	Transition
 The Main Menu Screen features a black background with a white planet icon at the top center. Below the icon, the text "Galactic Conquests" is displayed in a bold, white font, with the tagline "Dream. Build. Dominate." in a smaller font underneath. Four white rectangular buttons are arranged vertically in the center: "CAMPAIGN (SINGLE PLAYER)", "CUSTOM (MULTIPLAYER)", "GAME OPTIONS", and "EXIT".	MAIN MENU SCREEN Screen shown at game launch. Game Options will have options for turning game music on/off and setting volume. Close Application if “Exit” button is clicked.	Transition to Campaign Screen when Campaign Button is pressed. Transition to Custom Play Screen when Custom Button is pressed.
 The Custom Mode Screen has a black background with a small white planet icon in the top right corner. The text "CUSTOM MODE" is centered at the top in a bold, white font. Below this, there are two white rectangular buttons side-by-side: "HOST GAME" on the left and "JOIN GAME" on the right. At the bottom center, there is a single white rectangular button labeled "MAIN MENU".	CUSTOM PLAY SCREEN Screen gives the player the option to either host a game or join a game over the network.	Transition to Host Custom Screen when Host Game button is pressed. Transition to Join Custom Screen when Join Game button is pressed.

	<h3>HOST CUSTOM SCREEN</h3> <p>Screen provides configurable custom game options for the host. Other games can connect to this new game. The host chooses when to start the game by clicking "Host Game".</p>	<p>Transition to Fight Screen when Host Game is pressed (and there are at least 2 players including Host and AI players).</p> <p>Transition to Main Menu Screen when Main Menu is pressed.</p>
	<h3>JOIN CUSTOM SCREEN</h3> <p>Screen allows the player to choose a name, ship and an available game to connect to.</p>	<p>Transition to Fight Screen after connected to a game and host pressed "Host Game" button.</p> <p>Transition to Main Menu Screen when Main Menu is pressed.</p>
	<h3>CAMPAIGN SCREEN</h3> <p>Screen gives the player the option to either load a saved game (if one exists) or start a new game.</p> <p>Note, there is a single saved game slot. The game is automatically saved every time the System Screen is visited.</p>	<p>When Load Saved Game is pressed, load in the game state (which includes current screen) and then show the current screen.</p> <p>Transition to New Campaign Screen when New Game button is pressed.</p>
	<h3>NEW CAMPAIGN SCREEN</h3> <p>Player enters their name and selects a starting ship for a new game.</p>	<p>When the Start button is pressed, a random location is selected then transition to relevant System Screen.</p>

	<p>SYSTEM SCREEN</p> <p>When the player arrives in a new system, the system screen is shown.</p> <p>If player has positive or neutral standing with the faction controlling the system, the player will be allowed to jump or use their hangar.</p> <p>If the player has negative standing, they will have the option to either fight or attempt a bribe.</p> <p>However, if the player is on a mission to take this particular system, the option to bribe will not be available.</p>	<p>Transition to Hyper-Jump Screen when jump is pressed.</p> <p>Transition to Request Mission Screen when request mission is pressed.</p> <p>Transition to Hangar Screen when Hangar is pressed.</p> <p>Transition to Fight Screen when Fight is pressed.</p> <p>Transition to Bribe Screen when Attempt Bribe is pressed.</p>
	<p>HYPER-JUMP SCREEN</p> <p>A player is able to select any system which is adjacent to the current system (connected by the double lines representing a hyper-space link) by pressing on the desired system.</p>	<p>Transition to System Screen when "Return to System View" is pressed.</p> <p>When "Make Jump" is pressed, update the current location to the new system and transition to the System Screen (will show for the new system).</p>
	<p>REQUEST MISSION SCREEN</p> <p>Player will be offered a mission.</p> <p>A player can only have a maximum of one active mission at any given point in the game. Upon accepting a mission, the target system will be highlighted by a green circle on the hyper-jump map screen.</p>	<p>When either "Accept" or "Decline" is pressed, return to the System Screen.</p>

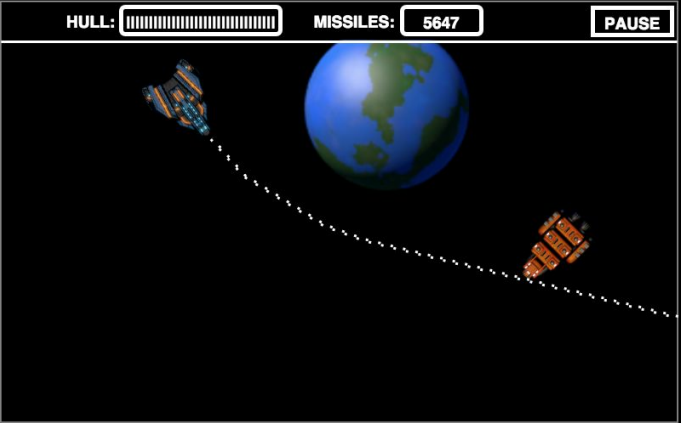
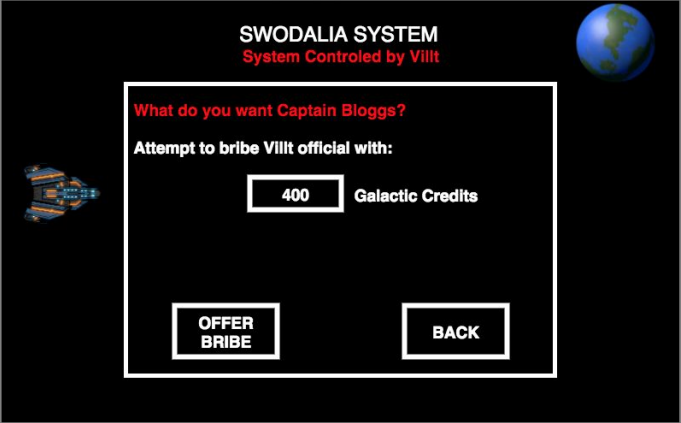
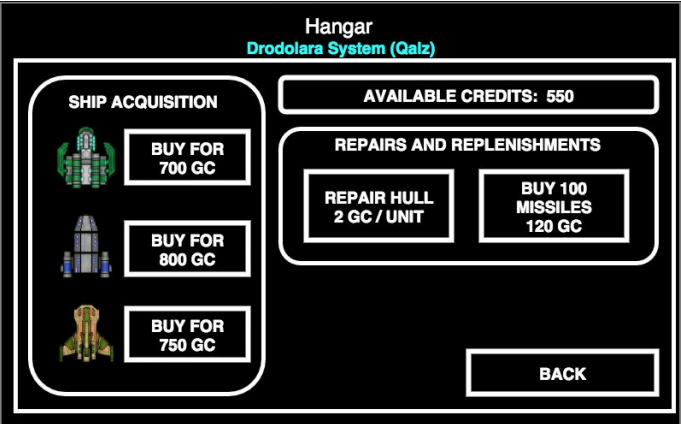
 <p>(Note: the missile's path are affected by the planet's gravity).</p> <p>See Table 2 in Platforms Section for Spaceship controls.</p>	<h3>FIGHT SCREEN</h3> <p>The window of view for the player is only a small sub-section of the entire solar-system (which keeps the player's ship in view).</p> <p>If the player's spaceship is destroyed and player has at least one more ship in their fleet, another spaceship can be selected and sent into the battle to replace the destroyed spaceship.</p> <p>Pause button pauses gameplay, text changes to "RESUME", then when button is pressed a second time, gameplay continues and button is renamed to "PAUSE" again.</p>	<p>When the battle is over (either enemy spaceship or player's fleet is destroyed), a battle outcome is displayed after which the game transitions to System Screen (for campaign) and either Host or Join Custom Screen (for custom game).</p> <p>For campaign, battle Outcome may consist of Mission Completion, Game Over or gaining temporary control of system (allowing hyper-space jump).</p>
	<h3>BRIBE SCREEN</h3> <p>The larger the bribe offered, the more likely and the larger the player's reputation with the faction may increase.</p> <p>If the player's reputation increases enough, the System Screen will change and the player will no longer have to fight.</p>	<p>Transition back to the System Screen.</p> <p>(Screen may change if reputation has been sufficiently improved.)</p>
	<h3>HANGAR SCREEN</h3> <p>Player can buy new ships to add to their fleet. They can also repair or replenish missiles on any of the ships in their fleet.</p>	<p>Transition to System Screen when back is pressed.</p>

Table 1: Screens with Mockups, Descriptions and Transitions.

Goals

For a campaign game, the player wins by becoming the leader of a faction and then wiping out the other two factions such that the player's faction controls every system in the galaxy.

To become a leader of a faction, the player must develop a very strong reputation in the faction by completing lots of missions for the faction. When the player's reputation is strong enough, the faction will ask the player to become their leader.

Player loses (and has to restart the game) if their entire fleet is destroyed in a space battle.

For a custom game, the goal is to destroy all the other spaceships while protecting your own. The player who pilots the last spaceship standing (not destroyed) wins.

Game Loop(s)

A game state object will contain all of the information about the current game. This can be converted into JSON and stored into a text file or parsed from the JSON file to create the game state object.

The game state object will include information such as current screen, player's name, reputations with each faction, number of galactic credits, information about all ships in the player's fleet and their status as well as the map state. A Model-View-Controller Pattern will be loosely used for the game, thereby making the game state object effectively the **Model** part of the game.

```
public class GameState{
    private Galaxy gameMap;
    private List<Ship> playerFleet;
    private String playerName;
    private int numberOfGalacticCredits;
    private Faction leadsFaction; //null if leads no faction
    private int reputationWithVillt;
    private int reputationWithQalz;
    private int reputationWithDoleo;

    enum Stage {MAIN_MENU, CUSTOM_PLAY, HOST_CUSTOM_PLAY, JOIN_CUSTOM_PLAY,
                CAMPAIGN, NEW_CAMPAIGN, SYSTEM, HYPER_JUMP, REQUEST_MISSION, FIGHT, BRIBE, HANGAR}
    private Stage gameStage;

    public Screen generateScreen(){
        switch(gameStage){
            case MAIN_MENU: return new MainMenuScreen(this);
            case CUSTOM_PLAY: return new CustomPlayScreen(this);
            //... generate relevant screen for each stage
        }
    }
}
```

Figure 7: Draft java code for Game State Class.

Separate classes for each type of screen are used. Each screen type, will use the game state in a different way to create a **view** and **controller** for the player. Each screen will be a subclass of the abstract class Screen implementing methods for draw(), mousePressed() and keyPressed():

```

public abstract class Screen{
    private GameState state;
    public Screen(GameState state){this.state = state;}

    public abstract draw();
    public abstract mousePressed();
    public abstract keyPressed();
}

```

Figure 8: Draft java code of the Abstract Superclass Screen.

The game loop using the game state object will therefore look like the following:

```

private GameState gameState;
private Screen currentScreen;

public void draw(){
    currentScreen = gameState.generateScreen();
    currentScreen.draw();
}

public void mousePressed(){ currentScreen.mousePressed();}
public void keyPressed(){ currentScreen.keyPressed();}

```

Figure 9: Draft java code for game loop.

Processing will call the draw method every frame and the methods mousePressed and keyPressed whenever triggered by user input.

In Context

There were two key games, which inspired the design of Galactic Conquests: Faster Than Light (FTL) and Planet Racer.

Faster Than Light (FTL)



Figure 10: Faster Than Light Title Graphic. (Ftl.wikia.com)

Faster than Light is a space role-playing game. Player pilots a spaceship jumping from system to system frequently entering into random space battles. Figure 11 shows a screenshot of a space battle. Notably,

both ships remain static, however, action takes place in Real-Time. Player micro-manages crew-members, weapon targeting and energy distribution amongst different ship systems.



Figure 11: Screenshot of a space battle in FLT.

In FLT, the player must defeat the enemy at the current location or survive long enough in the fight for the engines to charge in order to jump away. To jump, the player clicks jump then selects an adjacent location (which is close enough to jump to).

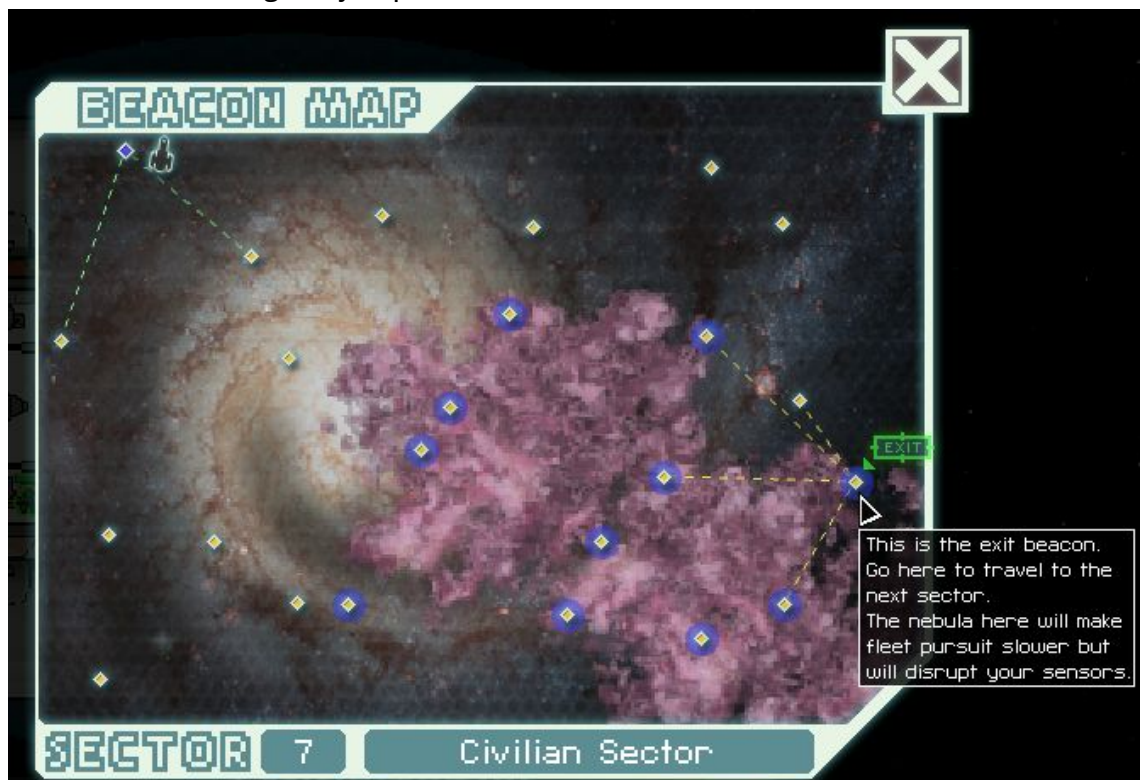


Figure 12: Screenshot of Sector Map.

In Figure 12, the player's location is shown at the top-right of the map and allowed jumps to two possible location are shown by the dashed green lines.

The idea of this jumping between locations inspired the navigation style of jumping between systems in Galactic Conquests. Furthermore, while in FLT, upon arriving at a new location there may or may not be

a battle. In Galactic Conquests, there will only be battles if it is a system controlled by faction which doesn't like the player or for a specific mission.

When the player's ship is destroyed in FLT, the game ended and the player had to restart the game from the beginning. However, the player could reload the game from where the player left off from their previous game session. The player would feel a huge sense of loss after developing an emotional attachment to their ship to then see it destroyed. This results in a player mentality of taking the game seriously and being more strategic at analysing risks - ultimately making the game more fun. A similar model has been used for Galactic Conquests of forcing a player to restart when their fleet is destroyed and only allowing the state of the previous game session to be reloaded for this reason.

Planet Racer



Figure 13: Main Menu (Left), Shop Screen (Right) in Planet Racer.

Planet Racer is a flash mini game created by FreeOnlineGames.com where the player goes from planet to planet challenging opponents to drag races. Before each race you can buy a new car, or upgrade your existing car. The player's reputation increases as they beat opponents with higher ratings. Higher reputations are needed to move to planets with larger rewards and stronger opponents.

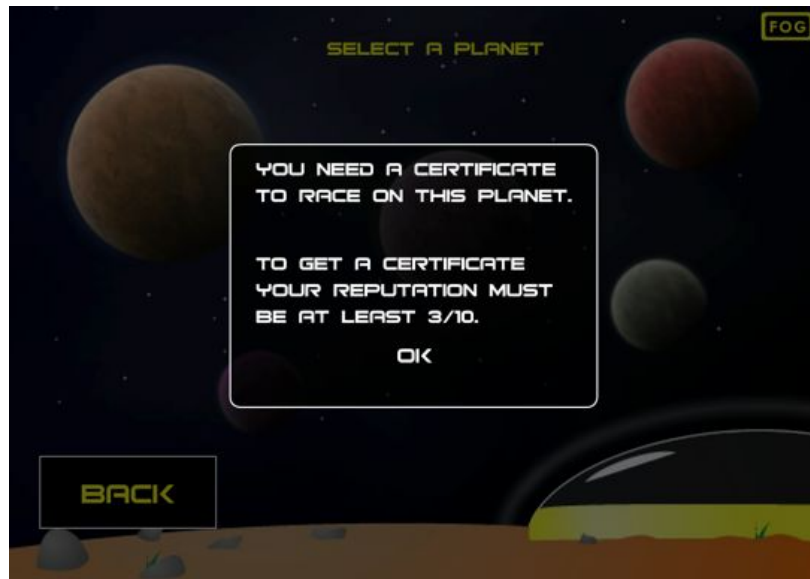


Figure 14: Player's reputation is too low to move to next planet in Planet Racer.

Galactic Conquests takes this idea of reputation further. In order to win the game, the player will need to gain a very high rating with a faction in order to become their leader. Furthermore, the player will be able to upgrade/repair their fleet and freely move through systems control by factions which feel positively or neutrally about the player.

Platform

The game will be first designed for desktop with a scope to then port the game to an android tablet. The desktop version will use the keyboard for controlling the spaceship in the Fight Screen using controls as stating in the “Desktop Controls” column of table 2.

For the tablet, there will be a virtual joystick in the bottom-left hand corner of the screen enabling the player to use their left thumb to control the spaceship movement. There will then be a virtual fire button at the bottom-right hand corner of the screen which the player can use their right thumb to press in order to fire missiles.

Spaceship Action	Desktop Controls	Tablet Controls
Rotating Ship Left	Pressing either A or LEFT -arrow key.	Move the virtual joystick left .
Rotating Ship Right	Pressing either D or RIGHT -arrow key.	Move the virtual joystick right .
Accelerate Ship Forward	Pressing either W or UP -arrow key.	Move the virtual joystick up .
Accelerate Ship Backwards	Pressing either S or DOWN -arrow key.	Move the virtual joystick down .
Fire Missile	Press SPACEBAR or ENTER/RETURN	Press fire button with right thumb.

Table 2: Spaceship Controls

Physics, Artificial Intelligence, Networking and Security

Physics

Physics will be used for modelling and updating the game play environment during the Fight Screen. Gravity, projectiles, mass objects, acceleration and velocity will all be modelled and employed.

Artificial Intelligence

AI players will be able to control spaceships, fire missiles at other spaceships and cleverly navigate the solar system. A number of different AI strategies will be tested and the combined to create AI players. For navigation either Potentials Fields or Rapidly Exploring Random Tree (RRTs) will be used in combination with a meta-strategy. Possible meta-strategies might include:

- Maintaining a fixed distance to the enemy where the distance is sufficiently small for a majority of missiles to be on target.
- Aiming at locations where the enemy is likely to move into.
- Attempting to lure enemy into the danger zone (too close to the sun).
- Attempt to avoid enemy until they have used up all their missiles.
- Hide behind a planet and sling-shot missiles at enemy using planet’s gravity.

Networking

Networking will be used for custom multiplayer games. A client-server infrastructure will be used as shown by figure 15. This is in order to ensure fairness in the game (e.g. host will have the same lag as over game players).

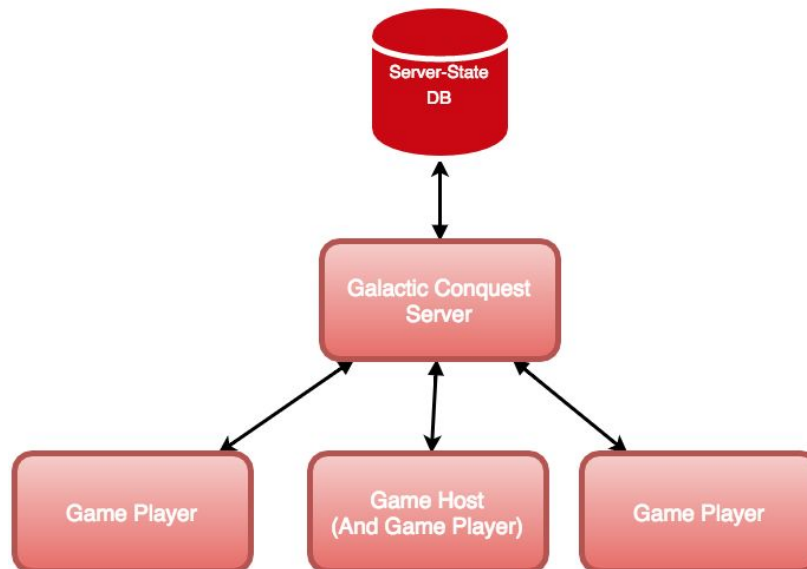


Figure 15: Networking diagram for a 3 person multiplayer game.

Each client will maintain a copy of the game state and will communicate any actions the player makes to all the other players via the Galactic Conquests Server. The Server will in turn maintain a copy of the game state and verify that each action attempted is allowed.

Security

Though the Galactic Conquests server verifies that only allowed actions are communicated to the other players, there are numerous other potential security issues:

Security Issue	Potential Solution
Players may collude (e.g. win-trading).	Use social in-game social pressure to dissuade players from cheating. For example, create an in game voting system where if 60% of players vote to kick a player, the player will be removed from the game.
Players may use bots/AI to player part or completely for them.	
Packet tampering by used to edit opponent's' moves.	
Game Server may be compromised.	Perform penetration testing. Make server code freely available, offer incentives for security vulnerability/exploits found (then patch them).
DoS attacks may be used to remove opponents from the game.	Such attacks are illegal. Attempt to log IPs of attacks and track them back to the culprit. Potentially sending a scary email or reporting the attacks to some authority.

Bibliography

Anon, (2015). [image] Available at: <http://millionthvector.blogspot.co.uk/p/free-sprites.html> [Accessed 19 Oct. 2015].

Ftl.wikia.com, (2015). FTL: Faster Than Light Wiki. [online] Available at: http://ftl.wikia.com/wiki/FTL:_Faster_Than_Light_Wiki [Accessed 19 Oct. 2015].

Planet Racer Title and Shop. (2015). [image] Available at: http://img-fotki.yandex.ru/get/3/fisherfr.a/0_1c38_35150d3_L [Accessed 19 Oct. 2015].

Vector, M. (2015). MillionthVector: Free Sprites. [online] Millionthvector.blogspot.co.uk. Available at: <http://millionthvector.blogspot.co.uk/p/free-sprites.html> [Accessed 19 Oct. 2015].