CS 679 Project 3 Plan

Group 1: James Merrill, Russel Mommaerts, Joe Kohlmann & Zack Krejci

We're Making Space Wars (too)

Our plain is to implement Tessa and Ryan's "Space Wars" idea. As it turns out, three other groups are doing the same thing, so we'll be carefully planning out the structure of the game to provide a unique take on the Risk-style/4X genre of gameplay.

"Space Wars" Original Proposal (from Tessa and Ryan)

Short Description

This is a Risk-like game set in space. The goal is to conquer all the planets in the Galaxy. You do this by sending units from one planet to the next. Each planet you own gives you a certain bonus (money, a different unit type, better defense, ...). Each planet can only reach certain other planets. There are one or more (AI) opponents with the same goal. Once you own all planets, you win the game.

Detailed Description

The player starts out as the ruler of a few planets in a galaxy. Through these planets, he can produce units. Each turn, the player can send these units out to conquer other planets. At first, most planets are neutral – they are occupied by a certain amount of units that doesn't change. However, there are one or more opponents with the same goal as you have: to get all planets. The goal is to eliminate all opponents and/or occupy all planets. Additional goals include a victory based on capturing and holding on to certain planets or regions, having the biggest fleet of ships at some endtime, or defeating common enemies as part of an alliance.

Each planet you own gives a certain bonus. These can be of several types:

- Generate money and/or energy
- Provide access to a certain type of unit
- Factory to produce units (possibly limited to certain units)
- Increase stats (better overall strength/defense/whatever is used to determine
- combat resolution)
- Possible other bonuses...

Whether you win or lose (and how many units you lose) is determined by several factors. An example could be strength, amount of hits and defense. Different units will vary in these stats (eg, a unit with a lot of hits but at low strength, or a unit with weak strength but high defense). They will also have different costs and production times based on these stats. It could take several turns to produce a unit (therefore, it is a bonus to own several 'factory' planets). Planets are grouped in solar systems. If you own a complete solar system, you get a bonus. The game is set in space. The graphical look is up to the implementers.

Scalability Plan

The simplest possible version has these basic mechanics:

- Planets that provide bonuses
- Possible to attack other planets (a combat resolution system)
- Only possible to reach certain planets. Reaching planets could be based on available technologies or the planet's condition.
- Unit production
- A few different units & planet bonuses
- An AI opponent
- Technology progression. This could be based on acquiring certain planets (or classes of planets) so that expansion and tech are promoted together.

Things where scaling is possible:

- More different units & planet bonuses
- Different AI strategies
- A campaign (levels with different goals. eg. start out by eliminating 1 opponent in
- a 10-planet system, end with 4 opponents in a 50-planet system, slow introduction
- of new unit/planet types).
- An RPG – storyline
- Units could require several types of resources instead of just 'money'. We're considering one or two resources—one resource might be simpler given the time constraints for building and playing this game.
- If really ambitious: implement the combat phase as an active component where the player actually controls his units in some way. Possibly optional for the player. We're opting for a macro-level approach to combat rather than a zoomed-in phase like this.
- Could implement random generation of universes. We're pretty sure that random map generation is a must. We won't have time to hand-build individual map layouts.

Game Principles Discussion

The game has a clear goal; to destroy the enemy. Subgoals (conquering one planet or a solar system) generate rewards that benefit the gameplay. There could also be rewards for conquering an enemy.

The player chooses which planet to attack at what time. A planet with a larger enemy population might be worth attacking because it provides a bigger bonus. This creates trade-offs.

When the player gets further along in a level, he has more borders to defend. This increases the challenge. Also, he starts out bordering just neutral territories. These don't attack him, so at the start the game is easier.

Design Challenges

The implementing team will have to work out many details. The main things that have to be worked out/designed:

- The combat system
- The different types of units
- The different types of planets
- Whether or not there will be a campaign, and if so, the different level layouts. We plan on having the high-level goal simply be "win the game", without a set campaign.

- The graphical look & feel. Our game will use a top-level view (as many 4X games do) of a starmap with point-and-click interface to direct actions.
- The AI strategy/strategies
- Balancing of units
- Turn-based / real-time / hybrid. We discussed a turn-based system in which the player plans his or her moves and actions and then presses (or otherwise reaches) a “Make It So” button. At this point, all the player’s actions play out—so they would see two converging armies (as the player and the enemy picked the same planet to move to) or other repercussions of their choices versus the enemy’s choices. Visually, players would see their planned actions in some form—perhaps an arrow indicating movement, for example—before committing to them.
- RPG elements

Technical Overview & Challenges

The gameplay as we envisioned it is in 2D. It's possible to make it 3D though, or to make the graphics 3D even though the gameplay just happens in a plane.

Does every unit get rendered? We’re not sure yet. In avoiding micromanagement, it might be useful to visualize units as “swarms” that don’t allow the player to control an individual unit, but this could provide too *little* control to the user.

A data structure will have to be designed to store the different planets and their possible connections. We’ll use a graph system.

The opponents require an extensive AI. This can be hard, especially if you would like to be able to tweak the AI to adjust the difficulty level.

Most if not all of the gameplay will happen with the mouse. This should be relatively easy.

Games to look at that are similar/inspired us

- Risk (boardgame)
- Warlight (Risk in Flash game variant)
- Ultimate Wars (turns out to be pretty similar to our idea)
- Civilization (turn based combat, moving units around, combat based on stats) Rebuild (capturing territories that provide certain bonuses)
- Total Annihilation (Resource system based on income rate and low limit on max resources)
- Settlers of Catan
- Frozen Synapse (for turn-based system inspiration)

Response to Critiques

It’s clear that we need to start with a good set of balanced mechanics before building a whole game around them. We’ve already discussed focusing on combat and resource flow in our version of Space Wars in a way that lets the player focus on the high-level actions rather than have to micromanage every little part of the simulation. We’re also considering a paper-and-sketches prototype of the game before anything else. We currently plan to make the game turn-based.

We also want to look into incorporating some ideas from the “Battlefield Comm” game that Zack helped propose. One of those ideas is the concept that the player’s “influence” decreases as units move further away from the player’s “home planet”. Units themselves may also have a local area of influence that can push enemies away or create “tug of war” situations between warring planets—then positioning becomes more strategic than pure expansion.

The Risk genre itself has issues we’ve discussed, such as the “Australia” outcome where one player holes up in a corner of the game map—we want to encourage expansion and discourage small, closed networks of planets. Our ideas for this:

- Planets can have a finite resource production value (that accumulates every turn) and a finite unit capacity (I think Civ calls this “upkeep”). Having too large an army on a planet with too low a production value prevents the player from adding more units to the planet. This way, players will have to expand (or fight enemies) if they want to keep producing more units.
- Generally we will encourage players to attack from multiple fronts. There might be multiple ways to attack a planet, but perhaps one path contains obstacles such as a debris field or radiation. This way players will have to take more than just the most direct route to en enemy planet. We tossed around the idea of having multiple attack points increase damage, lower enemy defense, etc. We’ll see how this mechanic plays out in the paper version.
- We might implement the concept of “depleted” planets that no longer produce resources, have diminished capacity for units, and/or have hazards such as radiation or debris (déjà vu here). This way, nothing is certain—the player can’t just hammer away on the same planets for the entire game.

Group Inventory

Joe: 2-D and 3-D artwork, user interface design, architecture planning, storywriting.
Russel: Architecture planning and implementation, 3-D graphics programming.
Zack: Infrastructure programming, user interface design, physics implementation, data visualization.
James: Level implementation, setting development and storywriting, mechanics implementation.

Tool Choices

We’re going to use **WebGL** and potentially some WebGL engine such as Three.js, along with a healthy set of libraries (jQuery at minimum). We decided that the portability and relative familiarity of these technologies would be a good choice at this stage of the semester. We’ll build the user interface in HTML and CSS. If things get complex we may look into a UI toolkit such as Ext.js or SproutCore, or maybe run a small web server with web2py or similar technologies to save game state across sessions.

Milestones

11/18: Paper Prototype and Mechanics Complete
12/02: Basic Graph-Like Prototype with Random Map Generation
12/09: Game with AI and Orange Box Graphics, All Mechanics Implemented
12/14: Final Game with Art Assets and Refinements
12/22: Bonus “Things Are Broken (and we fixed them)” Milestone

Risks

1. The game is not fun because of micromanagement hell.
2. Developing reasonable enemy AI (or ally AI) is difficult.
3. The game mechanics don’t scale to a 10- to 15-minute long playtest session.
4. The game is too simple, so the player’s decision space is too small.
5. The game’s different mechanics don’t congeal harmoniously.
6. The player options are poorly balanced, leading to an obvious dominant strategy.

Concept Art