D.U.N.E

Group 11

Klas Flodin

Mikael Nilsson

Anders Ljungqvist

Erik Nikkola

Kaj Sandberg

Functional Requirements

User requirements

1. The user shall be able to, from the system menu, start a new game with a pre-made map, start a new game with a randomly generated map, load a previously saved game state, or save the current game state. The system menu shall also pause the current game state while the menu is active.
2. The user shall be able to construct units or buildings, and initiate research from the user interface using the mouse or keyboard shortcuts.
3. The system has one form of resource (credits), which is required to produce units and commit research.
4. The user shall be able to research improved technology for his units and buildings.
5. The user shall be able to choose between several different factions which all have unique abilities and technologies.
6. A user shall be able to design ground and air based vehicles.
7. The user shall be able to select units and structures with the mouse.
8. The user shall be able to give units orders using a keyboard and/or a two button mouse.
9. The system shall handle real time combat.
10. The system shall provide a computer controlled opponent for single player games.
11. Several users shall be able to participate in a multiplayer game over a local area network or Internet with one specific player as the host.
12. The user shall be able to define video, audio and game options.
13. Any user participating in a game shall be able to quit the game at any time.
14. The system shall end the game when one or more users are victorious.

System requirements

1.1Starting a pre-made map.

Invoking a pre-made map shall load a file from the system resources with a pre-defined shape that is the same each time the file is invoked.

Rationale: Playing on a pre-defined map shall let the user play on familiar territory. The conditions are all preset, and the user know what to expect. Additionally, a pre-defined map can be used to define either fair or unfair conditions which can be used or overcome.

Requirements: When invoked, the system shall load the pre-defined map into memory and use this as the foundation for a new game state. A previous game state (if any) shall be discarded.
The pre-defined map is a file containing exact information on the maps entire topography, starting conditions, starting positions, initial units and their positions, and victory conditions.

1.2 Starting a random generated map.

Making a new game on a map that the system shall generate when the game state is loaded.

Rationale: Replaying the same old maps again and again shall eventually get boring. A random map will make exploring the map interesting and will ensure that the starting conditions for each player is unknown.

Requirements: When invoked the system shall present the user with options that let the user specify different quantities on the map. Such quantities should include the

amount of terrain or resources, their cluster sizes, the amount of clusters, etc.

1.3 Save.

The save functionality shall save the current single player game state in it's entirity to the hard drive.
Rationale: For various reasons a user may be forced to abandon the current game. By saving the game state to the hard drive, the user is able to turn the system off and return to the saved game state at a later point.
Requirements: When the save game function is invoked by the user, the software shall present the user with a dialogue window, presenting a list of previous saved game states, and ask the user for a description of the saved game.
Should the given description match a previous description for a saved game state, the user is required to decide whether the old game state should be overwritten or if a new description should be given.
At all times the user should have the option to abort the save game and return to the current game state.
These game states shall be saved to the hard drive as separate files with the file name being the user specified description.

1.4 Load.

The load functionality shall restore a previously abandoned single player game state to usable state.
Rationale: The user should be able to return to a previous game state at any time. The current game might not go the way planned, or the last time the user used the system there might not have been enough time to complete a game.
Requirements: When invoked, the user is presented with a list of previous game states that have been saved to the hard drive. The user is asked which game state he or she wishes to load, whereupon the system shall load the entirity of this game state from the file.
When the load is complete, the system shall close the dialogues and present the user with the loaded game state.

1.5 Pause.

The pause function shall temporarily freeze the current single player game state, preventing any and all interaction with it, until unfrozen.

Rationale: A pause game function will enable the system to allow the user a fair chance to use the systems dialogue windows for whichever functionality, while ensuring that the computer does not overrun the player's position in the game - alternatively the user may just want a breather during the game session.

Requirements: When invoked, the pause function shall completely freeze the game state. No commands shall be possible to issue, and no interaction with the game state data shall be allowed. The pause function shall not freeze other parts of the system, leaving the user free to interact with the various functionalities of the system that does not directly involve interaction with the game state data.

This functionality should be invoked whenever a system menu is opened.

2.1 Building construction.

Placing and initiating building construction.

Rationale: Building construction will allow the user to form his or her military base after desire, allowing the user to emphasize on the necessary production facilities to further the users desired s trategy.

Requirements: The in-game menu shall provide the user with an overview of the available buildings. Activating construction of these buildings shall allow the user to deploy them on the map, where they shall be "constructed". Construction time shall be dependant on the number of construction sites and production facilities.

The system shall also evaluate whether the building placement is valid or not.

2.2 Unit construction.

Unit construction shall provide the user with the tools needed to stay competitive in the game.

Rationale: The user will want troops to destroy the enemy's base. Having a base of the user's own is not enough to win the game.

Requirements: The in-game menu shall provide the user with an overview of the available buildings. Activating training or construction of a military unit shall place it in the production queue.
The time until the unit is ready to command shall be dependant on the number of facilities capable of producing the desired unit.
Should units associated with different buildings be queued, these shall be constructed simultaneously, thus more than one unit may be produced at the same time but only one unit of the same general type.
Units which are ready to command shall be placed on the map near the production facility designated as the primary production facility for that type of unit.

2.3 Primary production facilities.

A primary production facility can be designated by the user to make sure that all troops that building can produce shall be placed on a predictable spot.
Rationale: By specifying the primary building, the user is able to eliminate problems of poor unit placement, blocked building exits, or simply managing a large base.
Requirements: Each building is associated with one unit type only. Units of a specific type shall be placed near the building marked as the primary building whenever possible. Should a produced unit be unable to be placed near the primary building, it shall be placed near an eligible building chosen at random.
If there are no eligible buildings where the unit can be placed, the unit's construction shall be put on hold until such a time as the user has made an exit available for the unit.

2.4 Unit types.

A unit type is a general indicator of what the unit is and what it can do as well as what facilities produce them
Rationale: By dividing units into types it is possible to divide the game content further and make the designated production exit easier to customize for the user.

Requirements: Each unit type shall have at least one type of production facility. Unit types have differing attributes in the game with regards to how they may interact with their environment, such as whether the unit type may drive over other units or not.

2.5 Production shortcuts.

To speed up game play the user may wish to use shortcuts to eliminate unnecessary mouse movements.
Rationale: Mouse movement and clicking is, broadly speaking, intuitive, but by no means as fast as a single keyboard button hit. Shortcuts will allow an experienced user to speed up his or her game play.
Requirements: Each shortcut must be unique, one button must not be associated with more than one shortcut. Shortcuts shall be made inactive when the user enters chat mode.

3.1 Currency

Currency is used to pay for buildings, units and research.
Rationale: Constructing a base and an army can not come for free, the currency is a limiter for how fast a user will develop in the game and will provide an incentive for not wasting resources.
Requirements: Other game mechanics shall provide options for gaining currency. Currency shall be deducted whenever something is bought.

3.2 Harvestable resources.

Spread across the map shall be harvestable resources which shall allow the user to gather the currency needed.
Rationale: Harvestable resources provides the player with an accessible form of resource, and an incentive to be pro-active and take control of other areas than merely the user's own base.
Requirements: There shall exist units that can harvest these resources. When gathered, the amount of resources on the map shall be diminished where it was gathered from. The user shall receive currency for these gathered resources.

Harvestable resources shall diminish as the game progress, and shall eventually be depleted.

3.3 Other resources.

The system shall provide the user with other options for gaining currency which are slower but can potentially last after all harvestable resources are gone.
Rationale: During long games a user will find that the resources will diminish to the point where they are gone, or when the supply line becomes unmanagable. These other resources will provide an option for continuing overly long games.
Requirements: These resources shall be slower than gathering resources from the map. They shall be renewable in the sense that, as long as the user fulfill the requirements for them, they shall last forever.

3.4 Salvaging resources.

When a vehicle is destoyed, it leaves a wreck that is waiting to be salvaged. Salvaging a wreck regains resources.
Rationale: Vehicle wrecks are an intuitive way of dealing with the question "but where did everything else go?"
Requirements: When a unit salvages a wreck, the wreck shall be removed after the salvage procedure is completed. Depending on the value of the wreck being salvaged, the user who initiated the salvage shall gain resources equal to the wreck's value.

3.4.1 Salvaging friendly units and structures.

An unwanted friendly vehicle or building can be redeemed by salvaging it.
Rationale: A user may find that a building he or she built was overly redundant and no longer needed. In order to cut his or her losses, that building or vehicle may be salvaged for a fraction of it's worth.
Requirements: Salvaging a building or unit shall reduce it's health incrementally over the period of time it takes to salvage it. After the salvage is complete the building or unit shall be removed and no wreck shall be left behind. A large fraction of the building or unit's resource worth shall be returned to the user's resource pool.

3.4.2 Salvaging technology

When salvaging enemy wrecks it is possible to gain a free research advance.

Rationale: This is a means to even out the game. If an advanced player attacks a less advanced player, the less advanced player (if he is still alive after the attack) has a chance of gaining some of the technology that he is lacking.

Requirements: When salvaged, there shall be a small chance that the user gain one of the technologies associated with that wreck. The technology gained shall be from the common pool of technologies accessible by all factions, and it shall be one that the user currently has not yet researched.

4.1 Research.

By researching, the user may unlock previously unavailable structure, unit, and upgrade options.

Rationale: Research promotes the user to expand his or her base as well as delays the more powerful units and structures until such a time in the game when they can compete in on a fair play field.

Requirements: Research requires buildings that are associated with the research to have been constructed. Research when activated is put into a research queue, which hen takes time to finish.

The time required per research depend on the number of associated research buildings the player has.

4.1.1 Unlocking research.

A sub-section of Research. Unlocking research unlocks options and make them available to the user.

Rationale: This provides restrictions that limits the user's development and allows the developers to control the game pace.

Requirements: Unlocking research may not be restricted from being researched by other unlocking research to prevent a dead-lock situation.

Unlocking research is a futher requirement other than building restrictions.

4.1.2 Upgrading research.

A sub-section of Research. Upgrading research makes units or structures better in some way.
Rationale: Upgrading research is conducted in order to gain an advantage over the enemy.
Requirements: Unlocking research requires special buildings associated with the particular research.

5.1 Faction selection.

At the start of each game, the user is required to select a faction.
Rationale: By supporting factions, the system must be able to present the player with a choice of which faction to play in one way or another.
Requirements: Prior to game start, the user shall be prompted to select a faction. This prompt shall not be made if a old game state is loaded. Once selected, factions may not be changed.

5.2 Faction differences.

The different factions shall focus more on different areas, this shall give the user a choice to customize his or her experience of the system.
Rationale: Without factional differences, the factions will only be teams or colorations. With differences each user will be able to chose a set of strengths and weaknesses.
Requirements: Although each faction has certain advantages that are unique to that faction, every faction share a certain subset of structures, units and research. Factional difference should not make one faction decidedly better than another faction.

6.1 Design dialogue.

The design dialogue is accessed from the system menu, this shall allow the user to, essentially, make his or her own unit which can then be built in game.
Rationale: The dialogue provides a user interface for the user to use when designing units.
Requirements: The design dialogue should be accessible at any time during the game play. The design dialogue shall

be accessed from the system menu. The dialogue shall also present options for saving or loading unit designs.

6.2 Designing vehicles.

Only vehicles can be designed. When designing vehicles, the user selects components for it which are assigned certain slots on the vehicle design.
Rationale: Slot limit is important to ensure that no extreme super units are designed.
Requirements: Each vehicle type shall have a limited amount of slots. Each slot shall have a specific type that equipment of that type can be placed there.

6.3 Design budget.

The design budget places a further limit on unit design. Better components make the vehicle more expensive to purchase and the design budget shall show the user how much the unit currently cost.
Rationale: The user needs to know what it is he or she is designing.
Requirements: Each component shall have a cost. The costs for the components shall be added together and presented to the user.

6.4 Saved designs.

The user shall be able to save designs he or she has made for use in latter games.
Rationale: The design process can sometimes take a long time, by allowing the user to save designs the user will be able to more quickly get on with the game.
Requirements: Designs shall be possible to save to a file. This file shall contain all the designs that are currently active. A design save file shall only contain designs for one faction.

6.5 Recustomization restriction.

The user shall not be able to recustomize an already built custom designed unit.
Rationale: Once a design has been finished all productions of units of that line are final, the only way to edit this is by creating a modified design which in turn leads to a

completely new vehicle line. This is a pure game play decision to limit the possibilities of just upgrading the same units.
Requirements: The system shall only allow new designs based on older ones, not editing the existing custom designs.

7.1 Selecting a single unit or building

The game shall let the user select a single unit to control its behavior.
Rationale: Selecting a unit is required to control the units behavior.
Requirements: The system shall provide the user with visual feedback of the selected unit.

7.2 Selecting a group of units

The game shall provide a selection system that lets the user multiple units or a single building. Multiple units shall be selected with a selection box while buildings can only be selected by specifically selecting it.
Rationale: Users must be able to select multiple units fast assign orders to their units. Selecting multiple buildings won't be necessary since all control of the building will be on a separate interface.
Requirements: The system will provide the user with a visual feedback of the selection area.

8.1 Unit mouse control

The game shall provide a system that let the user control the behavior of a single or multiple units.
Rationale: The user must be able to control his units with a two button mouse or
Requirements: The system shall be able to command units with the the mouse. If the mouse is clicked, the system shall be able to determine the most appropriate command based on whether it is an enemy, friend, or empty area clicked. The system shall also provide the user with visual and audio confirmation of the action taken.

8.2 Unit keyboard control

The system shall provide short cuts to let the user access certain functions directly via predefined keyboard commands.
Rationale: Keyboard shortcuts helps the user control the behavior of the selected units faster than only mouse control.
Requirements: The system shall be able to take keyboard commands that tell units what to do. The system shall also be able to, when a keyboard shortcut is invoked, temporarily change the behavior of the mouse clicks. Additionally, the system shall provide the user with visual and audio feedback.

9.1 Unit combat control

The game shall provide a combat system where all combat is real-time.
Rationale: The users cannot control each unit individually so the combat must be largely self managed by the system. However the user should be able to alter the behavior of a unit and withdraw the unit from combat if necessary.
Requirements: The system shall consider the behavior of the unit and modifies it's behavior if it enters combat. The combat shall be handled automatically until the user changes the units behavior.

9.2 Defensive building combat

The game shall provide a combat system for defensive buildings where all combat is real-time
Rationale: Defensive buildings should automatically attack enemy units of their intended type. The behavior can be altered by the user by selecting the building and modifying it.
Requirements: The system shall consider the behavior of the defensive building and modify it's behavior if it enters combat. The buildings behavior shall be handled automatically until the user changes it's behavior.

9.3 Fog of War

The game's fog of war is a visual representation of an area explored by a unit but not currently in it's visible range.

Rationale: Fog of war is a game design choise that makes the other users actions invisible on previously explored areas.
Requirements: The system shall provide a visual representation of the fog of war and make all action in the fog of war invisible to the user.

10.1 Computer controlled opponents
The game shall provide a computer controlled opponent if played in single player mode where the computer difficulty level may be set.
Rationale: A computer controlled opponent is needed
Requirements: The unit shall have behavior that changes during the game based on the opponents action and the environment. The changes are based on the preselected difficulty level.

10.2 Indestructible computer controlled neutral units
The system shall handle units that are indestructible but neutral.
Rationale: Indestructable units that are computer controlled are needed for specific types of units.
Requirements: The system shall provide indestructable units shall have a predefined list of behavior that is randomly selected once and may be altered during the course of the game.

11.1 Multiplayer
The system shall provide multiplayer option where one user acts as a host and let the other users connect directly to the host.
Rationale: Multiplayer option is a game design decision.
Requirements: The system shall provide means of hosting a multiplayer game and let other users connect over the internet.

11.2 Multiplayer teams
The system shall provide means of letting the users select teams in multiplayer mode during the initial setup when the all players connect.

Rationale: Multiplayer teams is a game design decision.
Requirements: The system shall provide means that lets the user select their own teams prior to a multiplayer game. These teams shall not be changed at anytime during the game session.

11.3 Multiplayer chat

The chat will provide users in a multiplayer game to communicate with each other.

Rationale: Means of communication is required to help users formulate a strategy. Certain communication should only be within the team so those messages should be hidden from other users not within the team.

Requirements: The system shall provide means of sending public and messages restricted by team.

11.4 Multiplayer anticheat

The system shall only let the users create valid units in a multiplayer game.

Rationale: A multiplayer game should only let users produce valid units

Requirements: The system shall check each unit produced by a user if it's valid.

12.1 Setting video options
The user shall be able to set video options.
Rationale: Modern monitors support a number of different resolutions, but most of them have an optimal resolution where the monitor performs the best. The user will want to be able to select this resolution to make sure the game looks it best on just his monitor.
Requirements: The system shall have a video option that the user can set as per his demand.

If no resolution has been set a default resolution shall be used.

12.2 Setting audio volume

The user shall be able to set audio volume.
Rationale: As most computers have different models of speakers/headphones their base volume will vary. This as well as the fact that the user will want to vary the game volume depending on his current desires means the in game audio effects and the music soundtrack volume should be able to be set after the users wishes.
Requirements: The system shall have an audio effect volume and a music soundtrack volume setting.
If no volume settings are defined a default value shall be used.

12.3 Custom soundtrack folder.

The user shall be able to specify a folder on his computer that has audio files in it, these shall be played in random order by the system as the in-game soundtrack.
Rationale: If the user plays the game for a longer time he might soon grow tired of the limited original soundtrack shipped with the game or not like the genre at all. It would then be convenient to provide the ability to use his own music in-game as the soundtrack from a folder of his own choosing.
Requirements: The system shall have an option to select a directory where the user has stored music files from where the system shall pick a random file to play.
If no directory has been picked the default original soundtrack shall be played in game.

12.4 In-game name

The user shall be able to choose an in-game name that shall be displayed to represent him.

Rationale: In a multiplayer game it would get very confusing if everyone had the same name in the chat,

therefore each player will have to select a unique username.

Requirements: The system shall have an option to specify an in-game username.
The system shall not allow two users with the same username in a game, in such a case the system shall add a numeric identifier to the end of the usernames to keep them separated.

13.1 Quit the game

The user shall be able to at any given time choose to quit the current game in progress and end the system.

Rationale: The user will want to be able to quit the game whenever he needs to as he might need his attention elsewhere suddenly. Anything else but an always-available quit function is not acceptable.

Requirements: The user shall immediately be disconnected from any games in progress and the full system shall quit to the original state of the computer as it was before the game started.

14.1 Victorious game by disconnection.

If all clients are disconnected from the host the system ends the game and declares the host victorious.

Rationale: If only the host remains there is no reason to continue the game as the single remaining player.

Requirements: The system shall award victory to the only remaining player, thus the host. Afterwards the system is returned to the main menu.

14.2 Victorious game by mass conquer.

If only one user team has any buildings alive that team has won the game.

Rationale: If one user team has destroyed all other buildings they have conquered the whole map and therefore won the game as no more opposing forces exist.

Requirements: The system shall award victory to the only remaining player team. Afterwards the system is returned to the main menu.

14.3 Lost game by disconnection.

If a user is disconnected from the game host in a network multiplayer game due to network failure he loses.

Rationale: If the user is disconnected he can longer participate in the game and he will therefore have no chance to win which leads to an immediate loss.

Requirements: The user shall be put as loser on disconnection from host and the game ends for the user.
Afterwards the system is returned to the main menu.
Any remaining units of the user shall be removed from the game.
Any remaining participants in the game shall be presented with a status message showing the disconnection of the user.

Lost game by annihilation.

If a user's all buildings are destroyed he has lost the game.
Rationale: If a user has lost all his buildings he can no longer produce any more units and therefore will not be able to win the game any longer.

Requirements: The user shall no longer have any buildings left in the current game. The user shall be flagged as a loser and the game ends for the user. Afterwards the system is returned to the main menu.

Any remaining units of the user shall be removed from the game. Any remaining users in the game shall be presented with a status message showing the loss of the player.

Non-functional requirements

User requirements:

1. Starting the game shall not take longer than 1 minute on the system specified below.
2. Generating a random map shall not take more than 2 minutes.
3. Loading a game shall not take more than 1 minute.
4. Saving a game shall not take more than 10 seconds.
5. The pause command shall freeze the game after at most 1 second.
6. Every command or action must be designed so that they are executed after at most 3 keyboard or mouse button clicks.
7. It shall not take more than 5 hours to learn the game for a player that has played more than 2 other games in the Real Time Strategy (RTS) genre.
8. Multiplayer mode shall support up to 8 players.
9. The system shall not crash more than once per 20 started sessions.
10. The system has certain hardware and software requirements.
11. The user shall be able to play personal music during game play.
12. All requirements assume that no other non essential background processes are running or interfering with the system.

System requirements:

1.1 Start up

A limited time shall pass between initialization of the system until first input state has been reached.

Rationale: Because it isn't very entertaining to wait for the system to start. the goal is to keep the star time down as much as possible.

Requirements: Between initiating the system and the first input state of the system has been achieved, no more than 1 minute shall pass.
Visual feedback shall be provided during the startup process.

2.1 Random map generation

After invoking random map generation a limited time shall pass before a map is presented.

Rationale: To increase game longevity the users shall be presented with the option to generate random maps.

Requirements: No more than 2 minutes shall pass before the user screen has presented a map after the function has been called.
Visual feedback shall be provided during the map generation process.

3.1 Load

After load command is given, a limited time shall pass before a previous game state has been restored.

Rationale: When the user wishes to resume a game he can choose to load a previous game.

Requirements: No more than 1 minute shall pass until previous game state has been achieved and commenced.
A process indicator shall show that the process is proceeding.

4.1 Save

After save command is given, a limited time shall pass before game has been recorded.

Rationale: When the user wishes to create a restore point he can choose to save the game.

Requirements: No more than 10 seconds shall pass until game activity is resumed after save command is given.
The user shall be presented with a dialog when the game has been successfully saved.

5.1 Pause

After the pause command is given, a limited time shall pass before game is frozen.

Rationale: The user could feel the urge to use the restroom or 50 minutes may have passed and the user need to take the mandatory 10 minute break. Having the possibility to pause is no doubt a life saver.

Requirements: No more than 1 second shall pass before system state has been frozen after the pause command is given.
The user shall be informed that the game is paused.

6.1 Command efficiency

Each action the user can perform shall be possible with 3 keyboard or mouse or less.

Rationale: In an effort to make commanding troops more efficient no more than 3 keyboard or mouse buttons should be pressed in a sequence to perform any action.

Requirements: We shall limit the maximum needed sequential inputs required to perform an action to 3.

7.1 Learnability

The key bindings of the system shall resemble those of other similar games.

Rationale: The standards set by widely used an implemented by most games in the RTS genre and should therefore present the user with a familiar system. The key bindings shall conform to those in the game “Red Alert 2” by Westwood Studios.

Requirements: Given a user that has played more than 2 previous RTS games no more than 5 hours shall pass before the user is familiar with keyboard and mouse controls and other in-game mechanics. The keyboard shortcuts shall conform to the standards implemented by Westwood studios after the release of Dune 2.

8.1 Multiplayer - Hosting

Multiplayer games will be organized based on Client-Server (CS) architecture.

Rationale: In CS, players exchange periodic updates through a central server, the game host, that is also responsible for resolving any state inconsistencies. The CS architecture is not scalable with the number of players due to a large bandwidth requirement at the server and so limits the possible multiplayer users to 8.

Requirements: The host computer must be able to handle 7 simultaneous connections.

8.2 Multiplayer

The system shall only require a certain amount of bandwidth for multiplayer games.

Rationale: Having more human opponents to play with and against increases the entertainment factor.

Requirements: The system must be able to handle 8 players, all of them running the game on a 256kbit/s connection, without any loss of game performance.

8.3 Multiplayer - De-synchronization

There shall be a limit to how often resynchronization events occur.

Rationale: Whenever players go out of sync the entire game needs to be resynchronized.

Requirements: Resynchronization of the full game state should not be necessary more than once every 30 seconds during network play.

8.4 Multiplayer - security

The system shall not interpret any packages over the network that are not part of the current multiplayer game.

Rationale: The system should not handle any unknown packages because the packages could be malicious.

Requirements: Each client shall verify that all packets received over the network are game packets to ensure that the system is not misused.

8.5 Multiplayer - Error handling

The system shall drop any multiplayer clients that are not reachable over the network over a certain amount of time.

Rationale: To prevent game time loss and wait time for bad connections the system should drop such clients after a certain amount of time.

Requirements: The system shall disconnect any multiplayer client when more than 15 seconds of 100% total packet loss has occurred

9.1 Mean time to failure (MTTF)

The system shall be stable.

Rationale: The player does not want the game to crash during game play.

Requirements: At most one critical failure per 20 games.

10.1 Space

The game shall be small enough to fit on older computers and not be of inconvenience to the user.

Rationale: Even light computer systems with small hard drives should be able to play this magnificent game.

Requirements: The system shall be designed to occupy no more than 100MB of hard drive space excluding Java Runtime Environment, soundtracks and saved content.

10.2 Implementation

The game shall be developed in a certain language.

Rationale: In an effort to increase interoperability we have decided to develop this game in Java and OpenGL.

Requirements: The game shall be written in Java using OpenGL and OpenAL as supportive libraries.

11.1 Music

The system shall allow for selecting a custom music folder.

Rationale: Players that have played the game might tire of the packaged music and should therefore be able to include their own music.

Requirements: The user shall be able to select a folder that contains music files in the OGG, MP3 or WAV format.
To consider performance requirements, at most 100 songs can be included.

Use case DUNE Ultimate Nuclear Extinction

Primary actor: Player

Stakeholders and interests

-Player: Wants mind blowing game-play and satisfaction.
-Developer: Wants the game to be played by as many people as possible so that they will buy the not for free successor.

Preconditions

The player successfully downloads and installs the game on his PC with Windows XP.

Success guarantee

Player understands how to play the game. When the game is finished and the player quits, the computer is returned to its previous state.

Minimum guarantee

None.

Main success scenario

1. The main menu appears.
2. Player presses Start new game and then choose which map to play on, pre defined or random
   generated.
3. Game begins.
4. Player builds his base and army.
5. Player fights and brutally slaughters the enemy.
6. Player wins and the game ends.
7. Player presses exit game.

Extensions

*a. At any time the game fails.
    1. Player restarts the game
    1a. Player starts over and plays a fresh game.
    1b. Player loads a previously saved game and resumes playing.
    2. Player does something else worth wile.
    3. Player thinks the game sucks and erases it from his hard drive.
*b At any time during the game all players are disconnected from the host.
    1. Game host is pronounced winner.
    2. Game ends.
*c At any time during the game if a player disconnects while there are more than one other remaining player that player loses. All his assets are removed from the game.
*d Player can at any time during the game chat with the other players.
*e Player can at any time during the game chat with his team mates.
2a. Player presses Start multiplayer game and enters the multiplayer menu.
    1a. Player chooses Join multiplayer game.
        1a. Player enters the IP-address to the host and clicks on Connect to game host.
            1. Connection successful to the host.

1a. Game is on hold, waiting for the rest of the players to connect.
1. Everyone is connected and the multiplayer game begins.
1a. One or more players don't connect and gets timed out. Game begins without them.
2. Player chooses faction and team.
1b. Player changes his mind and clicks on Abort and returns to main menu.

1b. Player chooses Host multiplayer game.
1. Player enters number of players and chooses which map to -use, pre defined or random generated.
2. Player clicks on Host game.
1. Game goes on hold waiting for the other players to connect.
2a. All players are connected and the multiplayer game begin.
2b. One or more players don't connect and gets timed out. Game begins without them.
3. Host chooses faction and team.

2b. Player presses Load game to enter the load menu
1. Player chooses a game to load from the list of previously saved games and presses Load.
2. Player changes his mind and presses Return to main menu to get back to the main menu.

2c. Player presses Options and enters the setup menu.
1. Player enters a desired name for his character.
1a. Player settles with the pre entered name Gurgel.
1b. Player has previously entered a name which is saved.
2. Player sets the desired volume of the sound in the game.
2a. Player settles with the pre set volume of the sound.
2b. Player has previously set the desired volume which is saved.
3. Player enters desired Custom soundtrack folder by browsing to a folder of his hard drive.
3a. Player settles with the preset included soundtrack.
3b. Player has previously selected a Custom soundtrack folder which is saved.
4. Player enters desired screen resolution to match the optimal screen resolution of his screen.
4a. Player settles with the default resolution.
4b. Player has previously selected a resolution which is saved.
5. Player presses Exit and returns to the main menu.

2d. Player presses Exit to exit the game.

3a. Player enters games
1. Player chooses to build a construction building
1a. Player places construction building on a valid tile on the map.

4a. Player selects a desired building from the construction tab
1a. Player places construction building on a valid tile on the map.

4b. Player selects a desired vehicle from the vehicle construction tab and the unit appears in front of the relevant building.

4c. Player selects a desired infantry unit from the infantry construction tab and the unit appears in front of the relevant building

5a. Player spots enemy units in close proximity to his base
1. Player selects a vehicle.
1a. Player right clicks on map to move the vehicle there.

1b. Player right clicks on an enemy unit to order the vehicle to attack.
1c. Player orders the unit to stand guard enabling the unit to automatically attack nearby enemy units.

2. Player selects an infantry unit.
2a. Player right clicks on map to move the infantry there.
2b. Player right clicks on an enemy unit to order the infantry to attack
2c. Player orders the unit to stand guard enabling the unit to automatically attack nearby enemy units.

3. Player selects a defensive building.
3a. The building automatically attacks nearby enemy units.
3b. Player overrides the building target priority by right clicking on an enemy unit.

4. Player selects an air unit.
4a. Player right clicks on the map to move the unit there.
4b. Player right clicks on the enemy unit to order the unit to attack.
4c. Player orders the unit to stand guard enabling the unit to automatically attack nearby enemy units.

6a. Player loses
1a. Game ends and returns the player to main menu. (if in single player mode)
1b. Player gets to choose to stay and observe the rest of the game with the players remaining. Otherwise the game ends and player is returned to the main menu (if in multiplayer mode).

7a. Player starts a new game.
7b. Player clicks on Options to change his settings.
7c. Player exits the game.

Special requirements

-Only alpha numeric characters in players Name in the Options menu.
-Only wav, mp3 and ogg sound formats accepted for custom sound in the sound folder in Options.
-At most 100 sound files will be loaded from the Custom soundtrack folder.