TWILIGHT

Dirty Pixel

Storyline

- *In the year 2525 the prophecy of the Goths came true and the galaxy Quasar appeared in the middle of the universe…*
- *Goths ,highly religious humans, proclaimed that the apocalypse was near…*
- *The war between Goths, Humans and a new breed living in Quasar called Sylans began…*

Twilight is a;

- Third Person,
- Sci-fi,
- Massively Multiplayer Online,
- Space Simulation Game.

General Features of Twilight

- Thrilling action
- Endless gameplay
- 3 different races
- 6 different ships
- Race-specific missions
- Experience and rank system
- Dynamic political stance
- Dynamic pricing

Thrilling Action

- In Twilight you become a pilot of one of the three races and try to make your way up as a commander.
- By upgrading your ship with the experience and credits you gain in your journey, you’ll have a chance to defeat your vicious enemies.

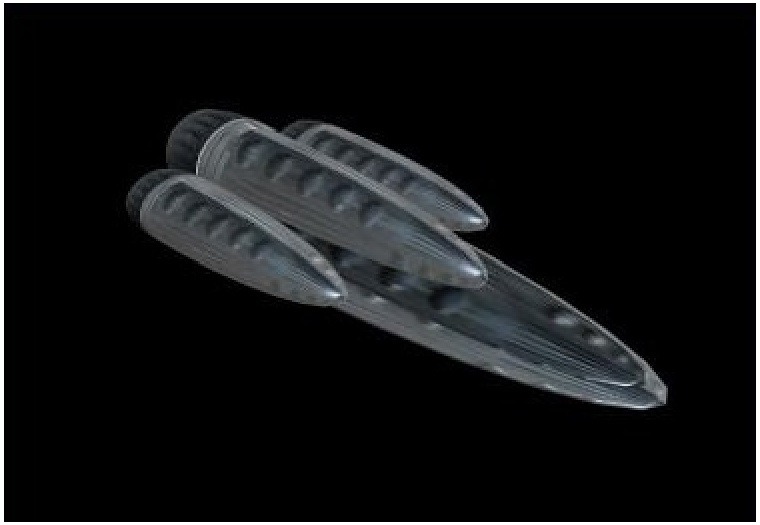
Endless Gameplay

- Twilight has a consistent universe.
- Players log-in to the ongoing game and log-off whenever they wish.
- When a player dies, he/she only loses his/her ship. A player returns back to the game by buying a new ship or using a previously bought one.

Three Different Races

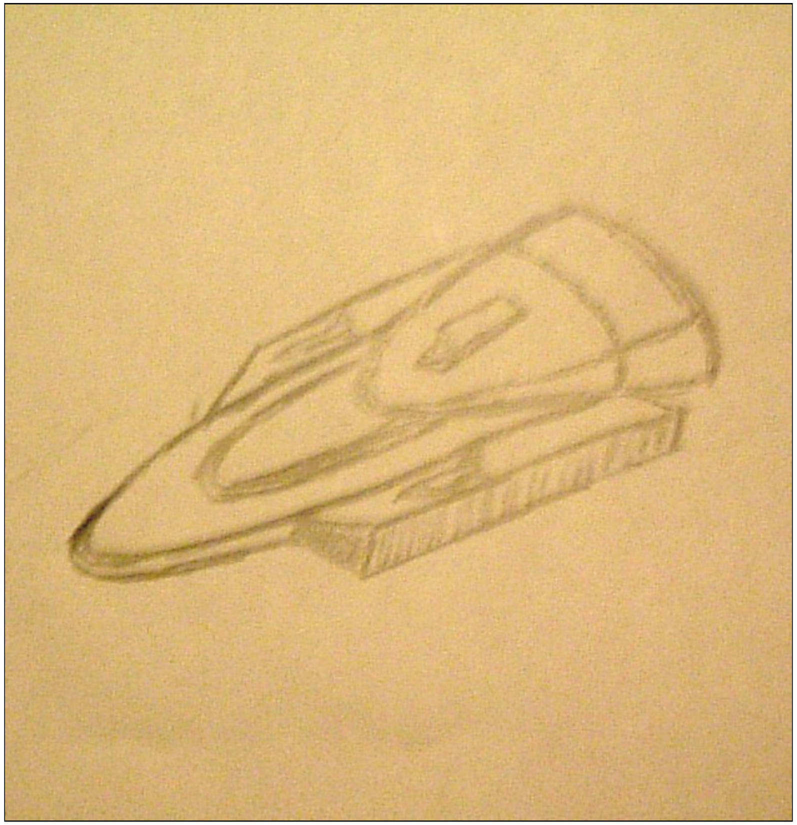
- Humans: Neutral against the other races. Ships have medium characteristics.
- Goths: Hostile against the Sylans and neutral against humans. Ships have heavy weapon power but light armor.
- Sylans: Hostile against Goths and neutral against humans. Ships have heavy armor but light weapon power.

6 Different Ships



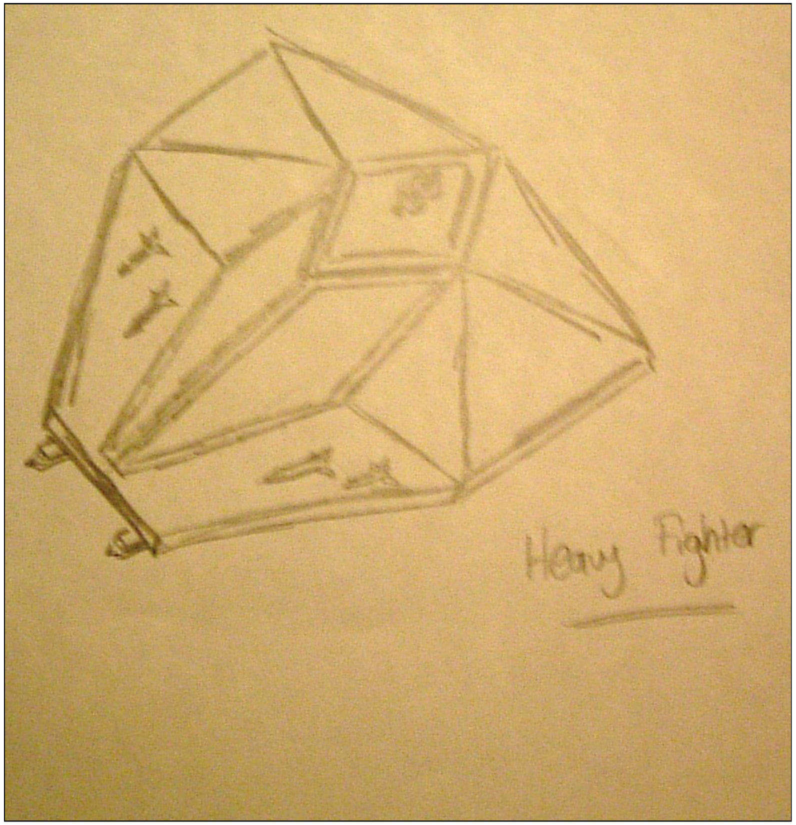
- Light Fighter:
  - Laser
  - Mine
  - Light Armor
  - Very Fast

6 Different Ships



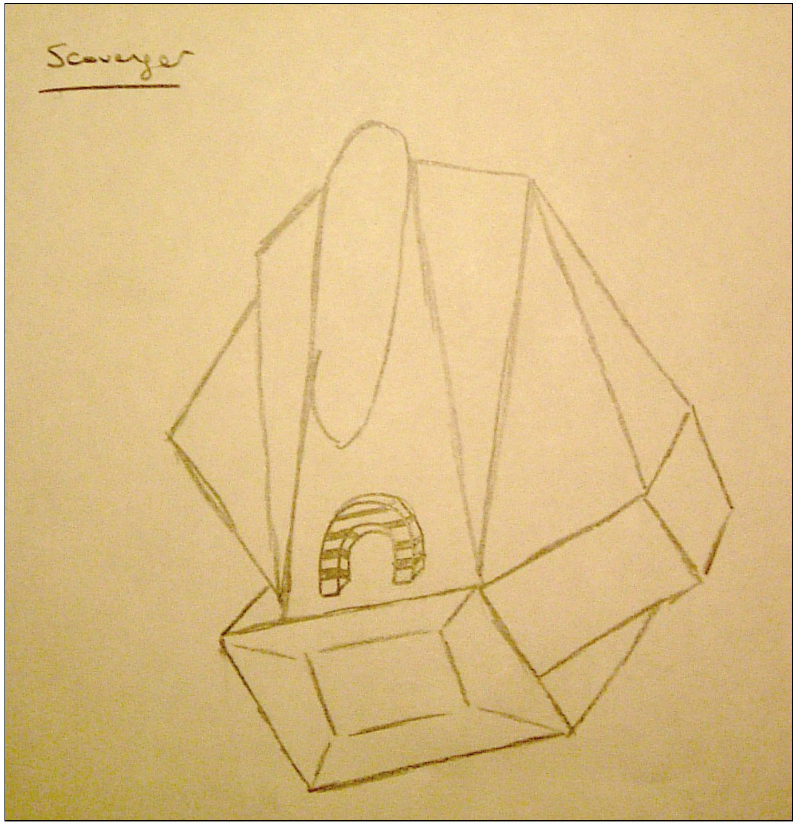
- Medium Fighter:
  - □ Laser
  - □ Rocket
  - □ Mine
  - □ Medium Armor
  - □ Medium Speed

6 Different Ships



- Heavy Fighter:
  - □ Laser
  - □ Rocket
  - □ Tesla Coil
  - □ Heavy Armor
  - □ Very Slow

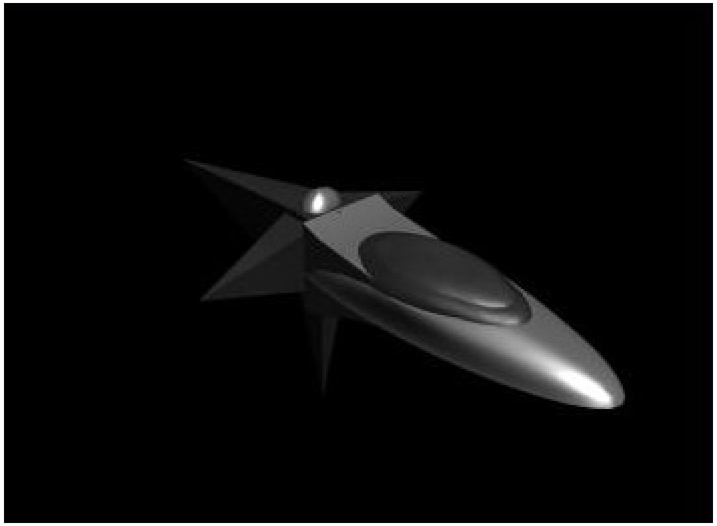
6 Different Ships



- Scavenger:
  - Laser
  - Magnet
  - Can pull leftovers of another ship and other ships as well
  - Can sell the collected particles in the hangar
  - Heavy Armor
  - Slow

TWILIGHT

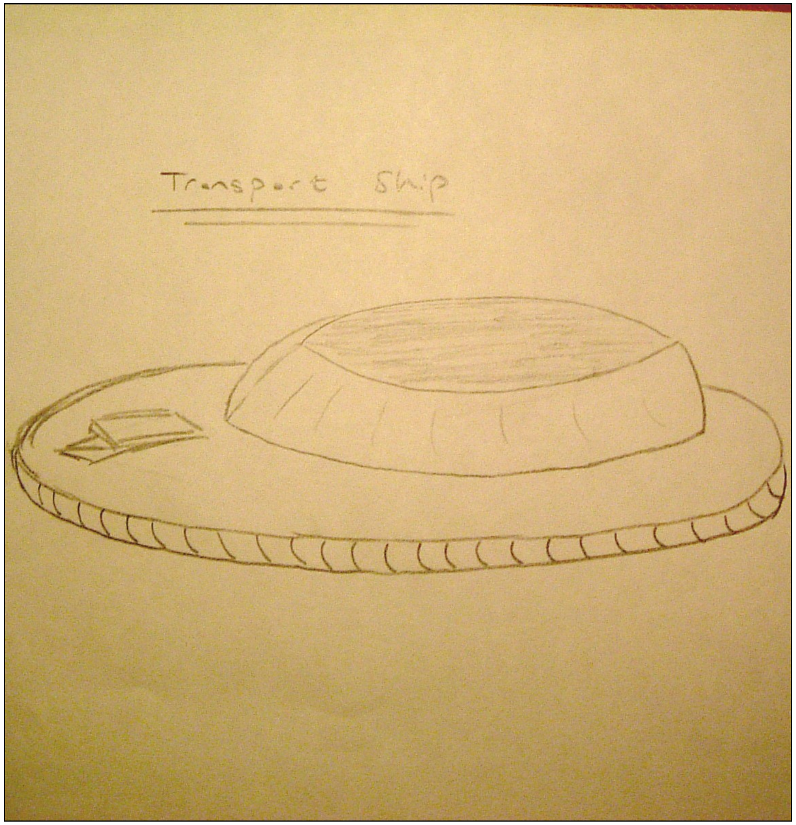
6 Different Ships



- Demolition ship:
  - □ Mine
  - □ Stealth Generator
  - □ Can demolish itself causing great damage
  - □ Can be invisible
  - □ Medium Armor
  - □ Medium Speed

Dirty Pixel

6 Different Ships



- Transport Ship:
  - Laser
  - Can be used to transport VIP in the missions
  - Very Light Armor
  - Very Fast

Race-Specific Missions

- Players can select a mission from their hangar.
- The selected missions varies according to the player’s race, experience and owned ship’s type.
- Missions grant experience points and credits to the user.

Experience and Rank System

- Players gain experience points throughout the game by destroying enemy ships and by completing missions.
- Player’s rank increases according to his/her experience points.
- New weapon upgrades will be available to higher ranked players.
- The ship’s speed, armor and weapon power increase with the player’s rank.

Dynamic Political Stance

- The races have two stances against each other which are: Neutral and Enemy.
- These stances change dynamically when a high ranked player selects a mission requiring stance change.
- The stance neutralizes after the mission.
- Bonus experience can be gained by destroying enemy ships.
- Headquarters will attack enemy ships on sight.

Dynamic Pricing

- The prices of the ships and the weapon upgrades change according to the other players ships and upgrades, creating some kind of inflation.
- There are only a limited number of ships and upgrades depending on the number of current players.

Gameplay Features

- Simple physics engine
- Independent ship parts
- Dynamic third person camera
- Easy controls
- Hyperjump
- No need to save

Simple Physics Engine

- An object will be defined by it’s position and rotation angles.
- A force acting on an object will change the speed and the moment of the object in all directions.
- The speed will be used to calculate the next position and the moment will be used to calculate the new rotation angles.

Independent Ship Parts

- In Twilight, ships consist of body and weapons.
- This way upgrading a weapon will change the appearance of the ship.
- Damage computation will be done for all these parts, which means it will be possible to blow up a weapon of any ship.

Dynamic Third Person Camera

- Twilight will dynamically change the position of the camera in gameplay.
- These changes will be defined by the speed, rotation angle and external forces acting on the ship.
- Faster speeds cause zoom out, while slower ones cause zoom in to the user ship.
- An unexpected change in the resultant force(i.e getting hit by a rocket) will also change the camera position in a logical manner.

Easy Controls

- The gameplay will be controlled by mouse and keyboard combinations.
- Strafe and accelaration will be controlled by keyboard, whereas other directional changes will be controlled by mouse.
- Controls are intended to simulate the behaviour of spaceships moving in space.

Hyperjump

- If a player has enough hyperjump cells, hyperjumping to another location in the gigantic universe is possible.
- While hyperjump is animated to the user there will be no packet transfer between the server and the client. Client will compute its own coordinates and inform the server at the end of the hyperjump.
- This will create some sort of a warping effect.

No Need To Save

- Exiting from the hangar will trigger a save since users are able to buy, sell weapons or repair their spaceships.
- When the user disconnects and logs-in the last saved state will be loaded from the server.
- If a user disconnects while the game is running the last gained experience will be lost since the game saves only in hangar.

Technical Features

- Networking
- Artificial Intelligence
- Graphics
- Sounds & Dynamic Music

Networking

- We will use TCP/IP protocol via Winsock API for messaging between server and the clients.
- We will take advantage of Windows threads in both server and client side.
- Our own encoding/decoding schema will be used for messaging.

Networking – Server Side

- Server side will make the computations for NPC creatures.
- There will be a huge game data in the server side from which the necessary information for every client will be fetched.
- Every received packet will be decoded and updated to this huge game data.
- The packets to be sent will be encoded into our packet prototype and sent to clients.
- No compression/decompression will be used.

Networking – Server Side

- Server side will calculate the active area for every client and send only the updated and active object informations to the clients.
- This will be achieved by simple boundary checking and timestamping.
- Worst case analysis shows a packet size of 2.1 kilobytes for 10 players in active area.
- Server will also access a compact database of users for saving the user state and log-in purpose. We will use a mysql database and mysql++ API for this purpose.

Networking – Client Side

- Clients will send very small packets to the server regarding to their states.
- All the object positions belonging to a client will be sent to the server as well as collision informations.
- Every client is responsible for its own position calculations.
- Every client is responsible for its own collision detection. In case of a collision, suitable information will be sent to the database which in turn will be sent to the client of the collided object for experiment gaining.

Networking - Synchronization

- Twilight networking principle is: "Fastest Connection Survives"
- A new packet will be sent to clients whenever a packet is received from that client. If the client does not send its information, no information will be returned by the server.
- This means if the client has a faster connection than another client, its chance of winning a battle increases since the information of the slower client is only updated when a packet is received by that user. This case; in turn, causes inevitable death of a lagged client in a dog-fight.

Artificial Intelligence

- AI will be controlled by the server side.
- There will also be an experiencing & ranking system for NPCs which will cause a change into more complex AI states.
- We are planning to implement 4 different AI states for NPCs which are Attack, Defence, Neutral and Retreat.
- NPC creatures will be another race called Eaons that are neutral creatures on which the Sylans depend.

Graphics

- We will use OpenGL for graphics rendering.
- For increased level of detail we will implement segmentation for the surfaces.
- For modelling, 3DMAX will be used. We are planning to implement our own 3DS loader.
- For lighting we will use 2 different light sources which will represent 2 suns in space.

Sounds & Dynamic Music

- Direct-Sound API will be used for sounds in Twilight.
- Animation states for the objects will also trigger sound events in the game.
- We’re also planning to implement transitions in music between aggressive and normal states throughout the game.
- Also the game menus will be enriched by music.

Team Members

- Anıl Yiğit Filiz (Project Leader)
- Berkehan Altınkaya
- Güneş Efe
- Derya Akpınar

Thank You

- Contact Us

  dirtypixel.twilight@gmail.com

- Any Questions ?