Space Simulator - Project 03

Evan Cummings and Joshua Bartz

CSCI 576 - Human Computer Interaction

May 12, 2014

1   Description

For our project idea, we would like to develop a 3D rendered engine via Python's interface to OpenGL. As it stands, we have a 3D particle simulation and an efficient way to calculate forces between particles and such. From this simulation, we would first like to create a space simulator - complete with gravity between planets and other space bodies and thrusters on the spacecraft which create rotational and directional momentum. With the addition of a rear engine and front braking thrusters, this would effectively simulate a real space craft.

Once the thruster part of the engine has been created, it would be interesting to add a missile with the same thruster system which could seek out and destroy asteroids or other space objects which may be in the spacecraft's way. Next, multiplayer could be developed where two opponents might chase each other through an asteroid field or planetary system by following some kind of particle trail.

Long-term development outside of the scope of this class includes random generation of planetary systems from a series of differential equation solvers which the user will run in the background. These simulations may take several hours to complete, but will allow the user to explore these new planetary systems with their spacecraft. From here, it would be interesting to add the option for planetary evolution, i.e., randomly generated atmosphere, oceans, and mountain ranges. The user may then enter the atmosphere with their spacecraft, complete with physics calculations and potential failure. This is a major highlight of the game as it would allow the user to generate and explore an entire solar system, all with the power of their home PC. In a sense this would be similar to entering into the computer's imagination, a completely unique experience.

1.1   Intended users

The intended users of the system are:

1. Anyone who enjoys flying is space.

2. People who enjoy the experience of speed.

3. People who enjoy competition.

4. Anyone interested in the physics of space.

5. People looking for a casual low-stress game.

6. People looking for a high-stress challenge.

7. People looking for a quick game.

8. People who enjoy wasting an entire day playing one game because it is so good.

9. People looking for a game they never tire of.

1

10. People who enjoy watching things explode.

11. People who wonder what it would be like to fire a rocket in space.

12. People who are technically-minded.

13. People who just want to have a good time.

14. Anyone who enjoys wireframe rendered 3D games.

15. People looking to escape from reality.

16. People who always wanted to be a space-person when they grew up but never could.

17. Younger people.

18. People who do not enjoy excessive violence in a video game.

19. Parents.

20. People who like flashing lights and colors that dazzle the senses.

21. People with a computer.

1.2 Usability requirements

The non-functional requirements for this game will be:

The game will be

1. fun,

2. relatively easy to learn,

3. realistic,

4. rewarding,

5. challenging,

6. artistic,

7. innovative,

8. visually pleasing,

9. surprising,

10. informative,

11. educational,

12. awe inspiring,

13. worth paying money for,

14. able to make the *user* feel artistic,

15. full of infinite possibilities,

16. replayable,

17. affordable,

18. easily executable from an PC created within the last decade,

1.3 Functional requirements

High priority:
The users of the game should be able to

1. fly the spaceship,

2. rotate the spacecraft 360 degrees,

3. move the spacecraft side to side,

4. change the speed of the spacecraft in all directions,

5. control the thrust of the main engine,

6. visually see the spacecraft's current speed and acceleration,

7. control the craft with a mouse and keyboard,

8. dodge space debris,

9. experience the gravitational pull of planets and stars,

10. fly through a nebula and see gas color changes related to velocity,

11. align the craft to a illuminated path that guides the ship into orbit of a planet,

12. see on-screen vectors for moving objects,

Medium priority:
The users of the game should be able to

1. lock onto a piece of space debris and destroy the Debra with weapons,

2. chase a network opponent around in space by following a particle trail,

3. fire a guided rocket which uses the same thrust engine as the ship,

4. fly through rings that guide the ship into orbit of a planet,

5. have a map and a way to navigate,

6. know their distance from their selected destination,

7. control the craft with a joystick,

8. adjust the sensitivities of the mouse or joystick,

9. invert either axis of input for the mouse or joystick,

Low priority:
The users of the game should be able to

1. choose from a menu to start or stop a game or configure settings,

2. configure the controls,

3. pause the game and return to the main menu,

4. scale the game's complexity to increase the framerate of the game,

5. display the framerate of the game,

6. choose from several different spaceships with different attributes,

7. customize their own spacecraft from attributes, i.e., speed, maneuverability, or color,

8. select from a list of rendering effects to increase framerate,

2 Design critique 1 feedback

2.1 Evan's feedback:

2.1.1 Cosmetic:

- *Model the known solar system.*

  Decision : In the time scales relevant to our game, this may be a challenge. We have discussed increasing time and providing near- or at-light speed travel. The physics of the planets may also be included in this simulation with gravitational forces acting between each planet and the spacecraft.

- *Include a clear navigation system.*

Decision : We will indicate the current speed, direction, and acceleration of the spacecraft in a corner of the screen, as well as a compass. Also, tiny particles will be included to give the player a sense of speed as they move through empty space.

2.1.2   Functionality:

- *Provide a goal other than simply flying around in space.*

  Decision : We plan on implementing an obstacle course and entering into orbit around a planet. We will also be able to launch missiles at space debris and learn about space and physics. Also, the challenge with the simulator lies in simply avoiding collisions with planets and other space debris.

- *Show the player statistics of the space vehicle such as speed.*

  Decision : New statistics and feedback options need to be explored.

- *Provide player customizations for the spacecraft.*

  Decision : We would like to allow the user to apply attributes to the craft such as maximum speed and maneuverability.

- *Make it possible to scale the difficulty and realism.*

  Decision : We would like to scale the size of the space to render and quality of visual effects, as well as a 'simulator' mode providing the most realistic space movement and 'arcade' mode with simplified controls.

- *Explicitly educate the players on the physics of space.*

  Decision : Ideas for this include Nebula color changes to particles as they gain kinetic and potential energy with a colorbar which indicates current energy in Joules. Also, we would like to show the player a path which places the spacecraft in orbit around a planet or star. This will include a visual indicator of direction and speed.

- *Challenge the player to put their spacecraft into orbit around a planet.*

  Decision : Rings may be included which the player could use to guide the spacecraft through while increasing or decreasing speed in order to obtain a stable orbit.

- *Allow the user to launch probes into space to investigate other systems.*

  Decision : While an interesting idea, the expanse of space would be difficult to model within the course of the class, but if there was an efficient way to randomly generate these other systems, this is an interesting option.

- *Provide a "missile cam" to guide missiles through obstacles.*

  Decision : Totally cool – If we have time, this will be included.

2.2   Josh's feedback:

2.2.1   Cosmetic:

- *Change unit of measurement for distance.*

  Decision : Light-years will be used in the next design. However, using miles or kilometers with scientific notation may be more appropriate

- *Add a load screen while the universe builds.*

  Decision : Some form of feedback is definitely required. An appropriate decision can be reached when the actual load time is known.

- *Add a thrust bar for acceleration feedback.*

  Decision : This should absolutely be implemented.

- *Provide more help for navigating the map.*

  Decision : Yes, it definitely needs to be more clear, especially for the different control schemes.

- *Use the real universe.*

  Decision : While this is would be a great option to have, it is probably beyond the scope of what we can accomplish in a semester.

2.2.2 Functionality:

- *Add educational attributes for teaching physics.*

  Decision : I would love to implement some educational aspects. While I don't believe it will be the main purpose of the game, some options should be considered.

- *Show trajectory and vector changes for moving objects.*

  Decision : This is one of the educational features I'd like to attempt.

- *If a player finishes a level, provide them with some educational information. Then, if they die, ask them to enter that information for the purpose of education.*

  Decision : While this is an interesting idea, I don't intend for this to be a question and answer game.

- *Add a tutorial to introduce the vector controls.*

  Decision : This is a great idea. I'd like it to be a low priority functional requirement.

- *Use rings to help a player steer the ship into orbit around a planet.*

  Decision : This was my favorite feedback idea, and as such am making it a medium priority requirement.

- *Be able to save and load a game.*

  Decision : This should be a part of most PC games, so I think an effort should be made to include it.

- *Fast travel between locations.*

  Decision : I'd like to include some system of warping between locations, but it's definitely low priority for now.

- *Enable power-ups.*

  Decision : Because this is meant to be more of a simulator rather than an arcade-style game, I doubt this will be implemented.

- *Allow player to choose different ships.*

  Decision : Users love to have some form of customization, so I think this should be implemented.

3 Design critique 2 feedback

- *Make the menu system easier to navigate and fancier.*

  Decision : We have decided to make the menu system a low-priority element and as such will not be implementing the interface for the class deliverable.

- *Change control configuration options to percentage.*

  Decision : Because the menu system is now low-priority, we likely won't get to the configuration options.

- *Explain configuration options more clearly.*

  Decision : See above.

- *Use a slider for configuration options and use tooltips.*

  Decision : See above.

- *Have pause menu and configuration screen be an overlay of the actual game screen.*

  Decision : The pause screen will likely be very simply done, meaning it could be implemented as an overlay.

- *Make menu system easier to see.*

  Decision : The menu system is low-priority now.

- *Use AU for units and show conversions to miles, kilometers, etc.*

  Decision : This would be a good educational element and will be considered if time allows.

- *Split up objectives and let users choose what they'd like to try.*

  Decision : For this semester, we are likely to only have a "sandbox" environment for users to explore.

4 Questionnaire Results

4.1 Questions Asked:

1. How interested would you be in playing a space simulator?

2. Would adding educational elements increase or decrease your desire to play such a game?

3. Please rank the following features in the order you would prefer to see in a space simulator, 1 being the one you would like to see the most, 5 being the least:

   - A feature that guides your ship into orbit of a planet, using rings and velocity indicators as guides.

   - Showing vectors of moving objects to demonstrate how they change due to collisions or gravity from nearby objects.

   - The ability to destroy asteroids using weapons such as guns or missiles.

   - A universe that is randomly generated.

- On-screen physics calculations.

4. Would realistic spaceflight controls add to or detract from the experience? In other words, if a thrust is applied to the ship to cause it to travel in a certain direction, it would continue to travel in that direction until an opposite force is applied.

5. What do you envision as being the ideal way to control a spaceship in a simulator? Examples: joystick, mouse, keyboard...

6. Which is more important to you: realistic visuals with fewer objects or realistic physics on more objects.

7. Would a multiplayer aspect increase your desire to play this game?

8. Other than those already listed, can you think of any features you would like to see in a space simulator? If so, please list them.

4.1.1   Results:

10 people were polled in total. Nearly all of them expressed interest in playing a space simulator. While some expressed interest in adding educational elements, the results were not overwhelming for either side. Because of this, we have decided to implement some educational features while not making education the focal point of the game. For example, strong support was given to the orbit and on-screen vector information. Because of this, those two items have been added as low-priority functional requirements. Additionally, most respondents favored realistic controls and physics to a more arcade-style system with better visuals. We envision these being features that contribute to the educational nature of the game. Because of the lack of strong support for a multiplayer system, it will remain a low-priority for the first version of the game.

5   Cognitive Walkthrough

5.1   Main representative tasks:

1. Translate and rotate the spacecraft in 360 degrees. Controls will be provided to roll, pitch, yaw, sidestep (move perpendicular to direction of travel), move forward, and move backward.

2. Dodge space debris which are randomly located throughout the area. If these object hit the spacecraft, the craft will be moved off in the opposite direction and rotated. This rotation and translation will need to be corrected by space simulator controls and will be very difficult – this provides great motivation for evading the debris.

3. Experience the gravitational pull of planets and stars. The planets will affect the flight of the space craft and will need to be accounted for at all times. On-screen vectors showing the craft's velocity and the effect of gravity from stars and planets will aid the player in navigation. If the gravity of a planet overtakes the craft, the vectors will show that escape is impossible; when the spacecraft hits the planet, the craft will respawn in the starting position.

4. Fly thorough a nebula and visually determine the change in gas energy. As the craft moves through the nebula, the gas particles change color and move in the opposite direction of impact. As the craft moves through the gas, the player can make pretty patterns of high energy gas.

5. See vectors corresponding to forces acting upon the spacecraft. These vectors allow the player to better judge how much acceleration is needed in what direction. A stable orbit will be attained when the velocity vector of the craft forward is approximately the same as the velocity vector towards the planet. If a collision occurs between the space craft and debris, rotation vectors appear showing the torque applied to the craft, as well as the force vectors that the debris applied to the craft. Rotation, current velocity and acceleration vectors will always be shown to aid the player in navigation.

5.2 List of actions needed to complete each task

1. **Translate and rotate the spacecraft in 360 degrees.**

   (a) User presses appropriate key for action.

   (b) Vector appears showing direction of applied force.

   (c) If force is applied for translation, speed indicator either increases or decreases depending on direction of acceleration.

   (d) Player sees the craft's velocity or rotation visually change with respect to the surrounding space.

2. **Dodge space debris which are randomly located throughout the area.**

   (a) User identifies a probable collision.

   (b) User steers the craft away from the collision by applying a force in the desired direction of travel.

   (c) Vectors appear confirming that the forces are being correctly applied in the desired direction.

   (d) User continues to apply acceleration in the opposite direction of the collision until it is apparent that the collision is either avoided or inevitable.

   (e) The spacecraft continues to fly onward if the collision is avoided or else the craft is rotated and/or pushed in some non-desirable location.

3. **Experience the gravitational pull of planets and stars.**

   (a) User notices a high-magnitude acceleration vector of a color matching a planet or star.

   (b) User recognizes the fact that compensation is needed to correct for the pull of space-body.

   (c) User rotates the craft until the force vector of the planet in in the opposite direction of where the user would like to apply a translation force.

   (d) User presses keys corresponding to the application of translational force to oppose the effect of gravity from the space-body.

   (e) The space-craft's velocity changes as the forces are applied.

   (f) The space-craft either escapes the gravitational pull of the planet, achieves orbit, or is overcome by the planet's gravity.

   (g) If the gravitational force cannot be overcome, the spacecraft hits the planet or star and instantly respawns in the starting position.

4. **Fly thorough a nebula and visually determine the change in gas energy.**

   (a) User locates a nebula.

   (b) User presses the controls to apply enough force to reach the nebula.

   (c) User moves inside the nebula.

   (d) Nebula gas particles move out of the way of the spacecraft.

   (e) Affected gas particles change color from low-energy blue to high-energy red.

   (f) User continues to fly through the nebula and clears a path of particles. These paths are areas where the particles have bounced off into space.

   (g) user navigates through the nebula while experiencing gas-density gradients where gas have accumulated in clumps, the earliest stages of planetary formation.

5. **See vectors corresponding to forces acting upon the spacecraft.**

   (a) User starts with a completely stationary craft.

   (b) User applies a rotation force by pressing a key.

   (c) Torque vector appears on the craft (perpendicular to plane of rotation).

   (d) Angular velocity vector appears in the same direction as applied torque.

   (e) User stops pressing key of rotation force.

   (f) Torque vector disappears.

   (g) Angular velocity vector remains showing angular velocity.

   (h) User presses translational force key.

   (i) Translation force vector appears on the spacecraft.

   (j) Space-craft velocity vector changes magnitude and direction as the craft 's direction and speed change.

   (k) User releases the translation force key.

   (l) Translational force vector disappears.

   (m) Angular velocity and translational velocity vectors remain showing the craft's current speed, direction, and angular velocity.

5.3 Detailed analysis of tasks

1. **Translate and rotate the spacecraft in 360 degrees.**

   (a) User presses appropriate key for action.

      i. *Will users be trying to produce whatever effect the action has?*
         Given that the user already understands the controls of the game, yes.

      ii. *Will users see the control (button, menu, switch, etc.) for the action?*
         Yes, the keyboard.

      iii. *Once users find the control, will they recognize that it produces the effect they want?*
         Yes, they understand what key does what.

iv. *After the action is taken, will users understand the feedback they get, so they can go on to the next action with confidence?*
   The key is pressed, resulting in a force vector appearing on the craft, see following steps.

(b) Vector appears showing direction of applied force.

(c) If force is applied for translation, speed indicator either increases or decreases depending on direction of acceleration.

(d) Player sees the craft's velocity or rotation visually change with respect to the surrounding space.

2. **Dodge space debris which are randomly located throughout the area.**

(a) User identifies a probable collision.

(b) User steers the craft away from the collision by applying a force in the desired direction of travel.

   i. *Will users be trying to produce whatever effect the action has?*
      Given that the users are actively playing the game, this situation will become obvious when it happens.

   ii. *Will users see the control (button, menu, switch, etc.) for the action?*
      Yes, the keyboard.

   iii. *Once users find the control, will they recognize that it produces the effect they want?*
      Yes, vectors appear showing that the correct force is being applied in the desired direction of travel.

   iv. *After the action is taken, will users understand the feedback they get, so they can go on to the next action with confidence?*
      Yes, the craft will move in the direction desired.

(c) Vectors appear confirming that the forces are being correctly applied in the desired direction.

(d) User continues to apply acceleration in the opposite direction of the collision until it is apparent that the collision is either avoided or inevitable.

   i. *Will users be trying to produce whatever effect the action has?*
      Assuming that the craft still appears to be moving toward the debris, yes.

   ii. *Will users see the control (button, menu, switch, etc.) for the action?*
      Yes, the keyboard.

   iii. *Once users find the control, will they recognize that it produces the effect they want?*
      Yes, the force vectors appear and the craft accelerates in the direction of applied force.

   iv. *After the action is taken, will users understand the feedback they get, so they can go on to the next action with confidence?*
      Yes, feedback is continuous.

(e) The spacecraft continues to fly onward if the collision is avoided or else the craft is rotated and/or pushed in some non-desirable location.

3. **Experience the gravitational pull of planets and stars.**

(a) User notices a high-magnitude acceleration vector of a color matching a planet or star.

(b) User recognizes the fact that compensation is needed to correct for the pull of space-body.

(c) User rotates the craft until the force vector of the planet in in the opposite direction of where the user would like to apply a translation force.

    i. *Will users be trying to produce whatever effect the action has?*
    If the craft is rotated in an incorrect orientation, it will not be able to apply a force in the appropriate direction. Rotation will be applied first.

    ii. *Will users see the control (button, menu, switch, etc.) for the action?*
    Yes, the keyboard.

    iii. *Once users find the control, will they recognize that it produces the effect they want?*
    Yes, vectors appear showing that the torque is being correctly applied.

    iv. *After the action is taken, will users understand the feedback they get, so they can go on to the next action with confidence?*
    Yes, the ship rotates in the direction desired.

(d) User presses keys corresponding to the application of translational force to oppose the effect of gravity from the space-body.

    i. *Will users be trying to produce whatever effect the action has?*
    With the craft rotated correctly, and the gravitational force vectors clearly visible, the user should in theory know that applying a force to the craft in the opposite direction would be required.

    ii. *Will users see the control (button, menu, switch, etc.) for the action?*
    Yes, the keyboard.

    iii. *Once users find the control, will they recognize that it produces the effect they want?*
    Vectors appear and show that the force is being applied and the craft moves in the direction of applied force.

    iv. *After the action is taken, will users understand the feedback they get, so they can go on to the next action with confidence?*
    Yes, vectors appear and the craft's velocity changes.

(e) The space-craft's velocity changes as the forces are applied.

(f) The space-craft either escapes the gravitational pull of the planet, achieves orbit, or is overcome by the planet's gravity.

(g) If the gravitational force cannot be overcome, the spacecraft hits the planet or star and instantly respawns in the starting position.

4. **Fly thorough a nebula and visually determine the change in gas energy.**

(a) User locates a nebula.

    i. *Will users be trying to produce whatever effect the action has?*
    The nebula will be a large multi-colored blob and will be clearly differentiated from normal space.

    ii. *Will users see the control (button, menu, switch, etc.) for the action?*
    Yes, the keyboard.

iii. *Once users find the control, will they recognize that it produces the effect they want?*
Yes, vectors appear and the craft moves.

iv. *After the action is taken, will users understand the feedback they get, so they can go on to the next action with confidence?*
Yes, vectors appear and the craft moves.

(b) User presses the controls to apply enough force to reach the nebula.

(c) User moves inside the nebula.

(d) Nebula gas particles move out of the way of the spacecraft.

(e) Affected gas particles change color from low-energy blue to high-energy red.

(f) User continues to fly through the nebula and clears a path of particles. These paths are areas where the particles have bounced off into space.

(g) user navigates through the nebula while experiencing gas-density gradients where gas have accumulated in clumps, the earliest stages of planetary formation.

5. **See vectors corresponding to forces acting upon the spacecraft.**

This situation is presented hypothetically and as such no detailed description is available. The steps presented describe the steps and feedback received in detail. Because the vector feedback is ongoing and continuous, we expect the user to be able to correctly interpret the situation given they understand the controls.

(a) User starts with a completely stationary craft.

(b) User applies a rotation force by pressing a key.

(c) Torque vector appears on the craft (perpendicular to plane of rotation).

(d) Angular velocity vector appears in the same direction as applied torque.

(e) User stops pressing key of rotation force.

(f) Torque vector disappears.

(g) Angular velocity vector remains showing angular velocity.

(h) User presses translational force key.

(i) Translation force vector appears on the spacecraft.

(j) Space-craft velocity vector changes magnitude and direction as the craft's direction and speed change.

(k) User releases the translation force key.

(l) Translational force vector disappears.

(m) Angular velocity and translational velocity vectors remain showing the craft's current speed, direction, and angular velocity.

6   Heuristic Evaluation

1. Translate and rotate the spacecraft in 360 degrees. Controls will be provided to roll, pitch, yaw, sidestep (move perpendicular to direction of travel), move forward, and move backward.

(a) *Simple and natural duologue.*

The controls provided are the minimal set required to fly the spacecraft.

(b) *Speak user's language.*

The system is analogous to any flight simulator.

(c) *Minimize user's memory load.*

Minimal controls to remember.

(d) *Consistency.*

The vector feedback and behavior of the craft is always the same given the set of inputs.

(e) *Feedback.*

A moving "starfield" guides the player in translations. Rotation will alter the viewpoint of the user and be obvious.

(f) *Clearly marked exits.*

The appropriate control to reverse the movement is given.

(g) *Shortcuts.*

Not applicable.

(h) *Good error messages.*

The feedback in this case is the error message; if the craft moves incorrectly, the user has pressed the wrong key.

(i) *Prevent errors.*

Vectors help the user control the ship.

2. Dodge space debris which are randomly located throughout the area. If these object hit the spacecraft, the craft will be moved off in the opposite direction and rotated. This rotation and translation will need to be corrected by space simulator controls and will be very difficult – this provides great motivation for evading the debris.

(a) *Simple and natural dialog.*

Feedback from the world can be clearly interpreted.

(b) *Speak user's language.*

The system follows the laws of physics.

(c) *Minimize user's memory load.*

Once understood, the physics of space can be memorized and predicted.

(d) *Consistency.*

Physics do not change.

(e) *Feedback.*

Visual feedback is continuously given to aid the user if flight.

(f) *Clearly marked exits.*

Not applicable.

(g) *Shortcuts.*

Not applicable.

(h) *Good error messages.*

Not applicable.

(i) *Prevent errors.*

Feedback is continually provided to help aid the player in guiding the craft.

3. Experience the gravitational pull of planets and stars. The planets will affect the flight of the space craft and will need to be accounted for at all times. On-screen vectors showing the craft's velocity and the effect of gravity from stars and planets will aid the player in navigation. If the gravity of a planet overtakes the craft, the vectors will show that escape is impossible; when the spacecraft hits the planet, the craft will respawn in the starting position.

(a) *Simple and natural dialog.*

The interface is as minimalistic as possible to allow the user to understand feedback from the game engine.

(b) *Speak user's language.*

Again, the physics of space.

(c) *Minimize user's memory load.*

Minimal controls to remember and physics which are consistent.

(d) *Consistency.*

Yes.

(e) *Feedback.*

Continuous.

(f) *Clearly marked exits.*

N/A.

(g) *Shortcuts.*

N/A.

(h) *Good error messages.*

If the user applies and incorrect force, the effects will be immediately noticed via on-screen vectors.

(i) *Prevent errors.*

Continuous feedback helps minimize errors.

4. Fly thorough a nebula and visually determine the change in gas energy. As the craft moves through the nebula, the gas particles change color from blue (cool) to red (hot) and move in the opposite direction of impact. As the craft moves through the gas, the player can make pretty patterns of high energy gas.

(a) *Simple and natural dialog.*

Yes.

(b) *Speak user's language.*

People generally understand that blue represents cold and red represents hot.

(c) *Minimize user's memory load.*

Yes.

(d) *Consistency.*

Yes.

(e) *Feedback.*

Constant.

(f) *Clearly marked exits.*

N/A.

(g) *Shortcuts.*

N/A.

(h) *Good error messages.*

No "errors" as such for this task; the user is free to play!

(i) *Prevent errors.*

N/A.

5. See vectors corresponding to forces acting upon the spacecraft. These vectors allow the player to better judge how much acceleration is needed in what direction. A stable orbit will be attained when the velocity vector of the craft forward is approximately the same as the velocity vector towards the planet. If a collision occurs between the space craft and debris, rotation vectors appear showing the torque applied to the craft, as well as the force vectors that the debris applied to the craft. Rotation, current velocity and acceleration vectors will always be shown to aid the player in navigation.

(a) *Simple and natural dialog.*

For the user, the interface is similar to a space-pilots technical readout; velocities are clearly shown and applied forces clearly shown.

(b) *Speak user's language.*

The torque vectors are perpendicular to the plane of rotation and as such may not be easily understood. For this, some kind of initial help screen explaining the 'right-hand rule' might be helpful.

(c) *Minimize user's memory load.*

Once the vectors are understood, the color-coding and orientation of the vectors make their meaning clear. However, extra feedback might be beneficial to the user, such as a label on the control vectors (translation and rotation), as well as a number depicting it's magnitude.

(d) *Consistency.*

These vectors are consistent.

(e) *Feedback.*

The very nature of the vector is feedback.

(f) *Clearly marked exits.*

N/A.

(g) *Shortcuts.*

N/A.

(h) *Good error messages.*

Again, the user will see immediately if they have applied an incorrect force because the vector will be in the wrong direction.

(i) *Prevent errors.*

As much as possible, this is what the vector system has been designed for.

Conclusion of evaluation

From the last milestone, our team determined that the feedback from the ship's controls were not sufficient to enable the user to effectively pilot the craft. It was for this reason that we chose to include the on-screen vectors which provide immediate and continuous feedback on the state of the craft. The Heuristic evaluation brought to light the fact that the vectors may be difficult to differentiate between, bringing us to the conclusion that labels on the vectors would be beneficial to the pilot.

For this type of system, both methods of evaluation asked questions that were either irrelevant or obvious in the context of the action. However, they both resulted in an in-depth analysis of the system and brought to light a real issue with the system.
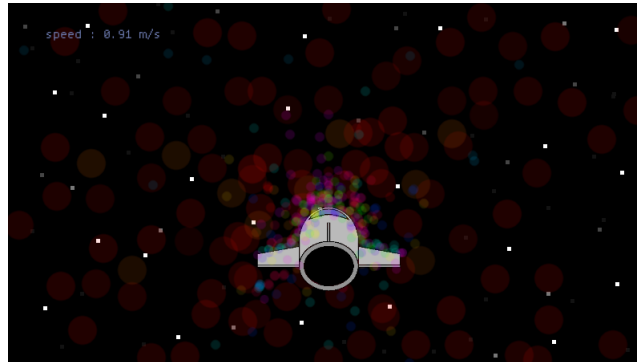
7   Immersion

Figure 1: Here we have the spacecraft flying into the nebula. The speed of the craft is shown in the top left corner. Notice that the color of the nebula changes as the craft pushes nebula gas particles out of its path. The user should be highly engaged not only perceptually, but also cognitively as they carve paths through the gas.
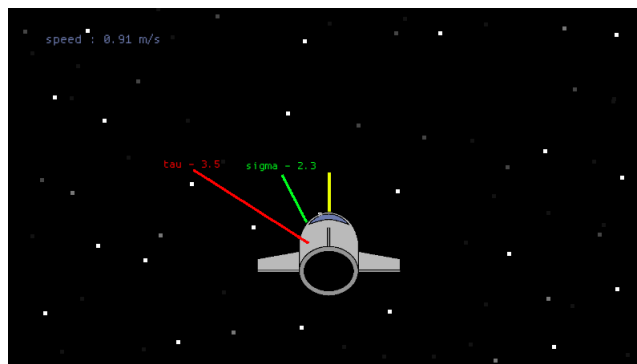

Figure 2: Vectors on the craft showing rotational velocity (tau) and translational velocity (sigma). The yellow vector represents a gravitational force vector from some nearby planet or star. The user should be highly cognitively immersed in this aspect of the game as they attempt to compensate for rotational and translational velocities.

Figure 3: Concept for entering the spacecraft into orbit; a difficult task which required complete immersion in the game.