

Gleeb-Glob Game Design

Revision History

Date Modified	Authors	Notes
11/12/2013	Brian Spates, Maggie Hewitt, Jason Grieves, Brian Hansen	Initial Draft
12/3/2013	Brian Hansen	Added and Fixed some of Vallino's comments, added player motion description

Table of Contents

[Market](#)

[1.2 Target Audience](#)

[1.3 Platform](#)

[1.4 Milestones](#)

[Introduction](#)

[2.1 High Concept](#)

[2.1 Summary Overview](#)

[2.2 Third-Party Software Used](#)

[Gameplay](#)

[3.1 First Minute](#)

[3.2 Gameflow](#)

[3.3 Victory Conditions](#)

[3.4 Graphics](#)

[3.5 HUD](#)

[3.6 Sounds](#)

[3.7 Controls](#)

[3.8 Molecules](#)

[Represented Geometries](#)

[Molecule Behaviors](#)

[Cluster Molecules](#)

[Barrier Molecules](#)

[3.9 Level Designs](#)

[Objectives](#)

[Flow](#)

[Upgrades](#)

[3.10 Plot](#)

[3.11 Characters](#)

[3.12 Artificial Intelligence](#)

[Player Controls](#)

[Molecule Movement](#)

[Collision Handling](#)

1. Market

1.2 Target Audience

The target audience of this game are college students in entry level chemistry courses. A common stumbling block for these students is conceptualizing molecules as 3D objects instead of scientific hieroglyphs. This game will illuminate the physicality of these molecules through 3D exploration and interaction.

1.3 Platform

As the game may become a mandatory part of a chemistry class, the game platform must be accessible to all users. In this vein, we have chosen the Unity3D development suite as it supports exporting to multiple platforms with limited platform dependent code changes. The focus of our initial development will be for the web platform as it is the most widely available platform. During development, we will lend an eye towards multiple platform compatibility so that the game may be ported in the future.

1.4 Milestones

Basic Level

The bare bones of a playable level are designed and implemented. At this point the level is much like a dress rehearsal for the game.

Complete Level

The basic level is built upon and all game play aspects of a level are implemented. The level is feature complete including music, art, SFX, and game states.

Zone 1

The complete level is extrapolated upon to generate the first set of levels included in zone one. This milestone also implies certain game logic to be completed that involves multi-level interactions (i.e. upgrade store, level selector, game state saving for users).

Zone 2 and Zone 3

Using the infrastructure developed during the last milestone the game will be expanded to include Zones 2, 3, and a Game Win state. This involves zone design, implementation, and usability testing.

Multiple Device Support

Porting the game to mobile devices.

2. Introduction

2.1 High Concept

The Moose is a three-dimensional space adventure on the molecular level where players race against the clock to collect molecules.

2.1 Summary Overview

In Moose the player will navigate their way through the vast expanses of molecular space to collect resources in the form of molecules. These resources will be used to complete missions provided by the player's base ship. The base ship will utilize different molecules as the building blocks of resources like fuel. The game will be split up into a number of levels, in each the player will have to collect a number of specific molecules. The player will have a set amount of time to complete each level. The player must avoid molecules too large to be collected by the ship's current capacity. Collisions with molecules too large to collect will damage the ship, if enough damage is incurred the ship will be destroyed and the player must restart the level. Points are given on mission completion. Bonus points are achieved for finishing missions early, collecting bonus molecules, and style.

2.2 Third-Party Software Used

Unity

The game will be written and tested using the free edition of Unity3Ds software suite. The free edition comes with some legal stipulations enumerated in the licence agreements links in section 6. The most important stipulations are that co-mingling of code written in unity pro and unity free is strictly forbidden.

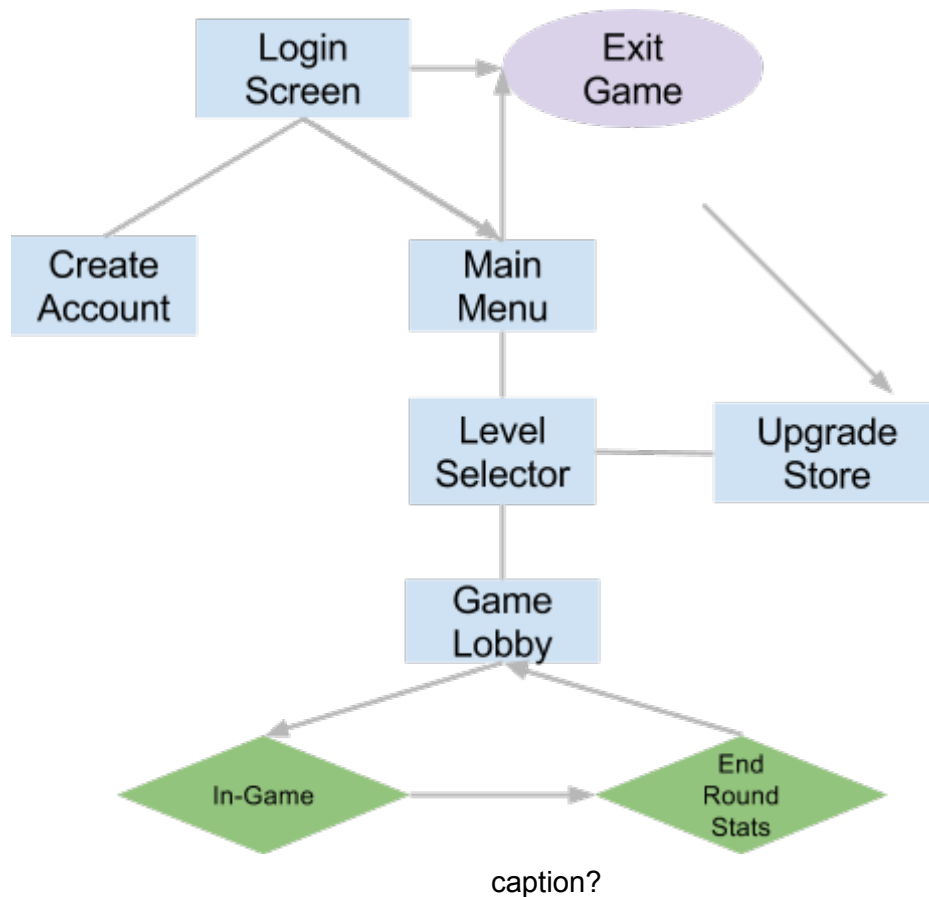
Blender/Maya

The game's graphical properties will be designed and developed using a 3D modeling tool. Some entities have already been built in Blender. Blender is an open source 3D modeling suite. On the advice of the experts in the school of Interactive Games and Media we may be switching to the industry standard modeling tool, Maya.

3. Gameplay

3.1 Gameflow

The diagram below illustrates the basic flow throughout the game.



3.2 First Minute

The user is prompted to either create an account or login. If the user does not have an account, they create one by selecting a username, password, school, and class. If the username is valid, a new account is created and the user can log in. If the user has a valid login, they progress directly to the main menu. The user can then select a level.

3.3 Victory Conditions

The victory conditions for each level involves collecting a number of a certain type of molecule(s) for the home base to use. The player has three minutes worth of fuel to explore the level to collect the molecules. Once the molecules have been collected, the ship must return to the home base to deliver them. The player must complete the level objective in order to move on, meaning if they don't meet the requirements to complete a level before fuel runs out, the player has failed the level and must try again. The player can also fail a level by destroying the ship through collision with other hazardous objects like TNT. As the levels must be completed in a linear fashion, the victory condition for a zone is to simply complete the last level in that zone. Once a zone is completed, the next level will be the first level of the next zone. To beat the game, simply beat the last level in the last zone.

3.4 Graphics

Graphics will be simple and consist of barbell style molecules. The barriers will consist of macromolecules. The base will have a very non realistic cartoony design.

3.5 HUD

The ship's heads up display will show information pertaining to the stats of the ship, the molecules needed to complete the level, the molecules deposited at the ship, the remaining fuel for the ship and the score. The damage is displayed on the ship through cracks in the windshield. If the windshield cracks too much, it will shatter and the game is over. A crosshair will be placed in the center of the screen and act as the reference for where the Cluster Buster and Tractor Beam shoot.

3.6 Sounds

The game will use royalty free music and sounds. The music will reflect the mood of the each zone and will be electronic or chiptune. Music for the intro screen, win screen, and lose screen will also reflect the appropriate mood. There will also be sounds for collisions and for use of the cluster buster and tractor beam equipment.

3.7 Controls

The game will control very similar to how a first person game would control. The player has the ability to look around using the mouse. The ship is constantly moving forward. The ship can use its boosters to speed the ship up and brakes to slow the ship down. Use of the Tractor Beam using the right mouse button and the Cluster Buster with the left mouse button.

3.8 Molecules

Represented Geometries

The following molecular geometries will be represented as collectable molecules throughout the game.

1. Linear
2. Trigonal Planar (flat triangle)
3. Tetrahedral
4. Bent-Geometries
5. Pyramidal

Molecule Behaviors

Molecules, both collectables and clusters, demonstrate the following behaviors.

1. Move through space
2. Rotate
3. Vibrate

Cluster Molecules

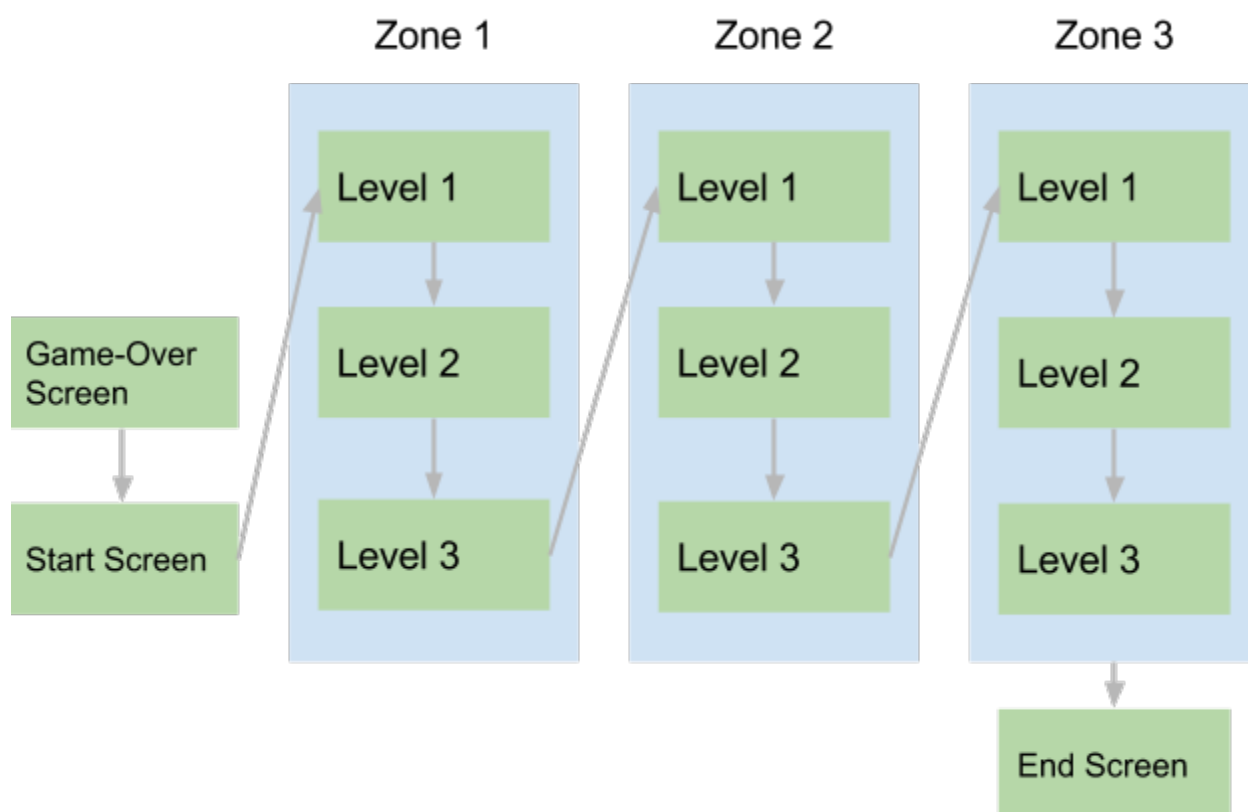
Cluster molecules are collectable molecules that are "stuck together" at the beginning of a level. Cluster can be broken apart and the molecules collected. They may or may not be rotating or moving through space.

Barrier Molecules

Barrier molecules are giant molecules that cannot be broken apart or collected. Barrier molecules act as obstacles and, upon collision, the player is penalized. They may or may not be rotating or moving through space.

3.9 Level Designs

The game is comprised of three zones each with three levels. Each zone highlights a different chemistry concept by changing the collectable molecules, molecule barriers, and movement of the molecules. Every level has an objective that needs to be completed to progress through the game. The objective for each level is similar across the zones; Level 1's primary objective is collecting fuel, Level 2's primary objective is base repairs, and Level 3's primary objective is base upgrades.



Objectives

The specific objectives for each zone and level are outlined below.

Zone 1 - Cold Zone

Level 1 - Collect molecules for fuel

Level 2 - Collect resources to repair ship

Level 3 - Acquire cluster buster

Zone 2 - Light Zone

Level 1 - Build solar panels to avoid collecting fuel molecules

Level 2 - Add armor to protect the base from future collisions

Level 3 - Upgrade to tractor beam

Zone 3 - Heat Zone

Level 1 - Heat destroyed solar panels. Collect molecules for fuel again

Level 2 - Repair ship solar panel damage

Level 3 - Upgrade cluster buster

Flow

Each level starts with the player on the base. The player has approximately 3 min to fly around and collect collectable molecules while avoiding obstacle molecules. The user must collect the required number of molecules and return to the base in the allotted amount of time to progress to the next level.

Equipment

The following items are introduced and upgraded throughout game play.

Tractor Beam

- Available to the player initially
- Used to collect molecules
- The player must hold down the tractor beam key when colliding with a molecule to collect
- If the tractor beam energy isn't on when the player collides with a molecule, the molecule will bounce off and further away in space. The player will receive a varying amount of damage based on the size of the molecule.

Cluster Buster

- Available after completing Zone 1 Level 3
- Used to break up cluster elements
- Aim at cluster molecules to break up
- When clusters are broken up, inside elements burst out
 - Inside element burst speed is a ratio of cluster buster level and the size of cluster.
 - Small clusters will violently explode.
 - Large clusters will slowly disperse molecules.

3.10 Plot

The Glib-Globicus people from the planet Globular are stuck in a sub-atomic world due an unexpected glitch in space-time travel. The Glib-Globicus have devised an ingenious way to shrink the mass (by changing the size) of space vessels in order to travel at the speed of light. The Glorb Base, although designed to travel in microspace, has suffered some unexpected damage and is trapped. In order to return to their home planet of Globular, they must repair their base through a series of repairs and upgrades. In order to retrieve the required resources for the repairs and upgrades, one brave Glib-Globicus must leave the base in a smaller ship to collect molecules.

3.11 Characters

The three main characters in this game are the Glib-Globicus spaceship and the HUD.

1. Glib-Globicus Ship - the main player and the protagonist.
2. Glib-Globicus "Glorb Base" - provides a brief description of the level mission
3. Help button - gives tips on how to overcome the level challenge.

3.12 Artificial Intelligence

User Input

Mouse - the mouse is used to rotate the ship.

W Key - accelerate up a top speed.

S Key - simulates a brake, decelerate to a minimum speed

A Key - strafe left, the player remains facing forward but moves horizontally to the left

D Key - strafe right, the player remains facing forward but moves horizontally to the right

Overall Player Motion Control

The motion is designed in that the player can not stop and is always traveling forward. In addition, the player cannot travel backwards. To travel in the opposite direction the player must turn around. All directions of player motion have dynamic acceleration and deceleration rates. For each direction, as the maximum speed for that direction is reached the acceleration diminishes and eventually becomes zero. There are separate acceleration, deceleration, maximum speed, and minimum speed definitions for strafing from those defined for forward motion.

Overall the players motion is designed as if there was air resistance. When no player movement keys are pressed the player will passively decelerate motion existing in all directions. For forward motion, this adds an additional degree of player control to the forward speed of the ship as the player is provided an additional method and rate for deceleration. For the horizontal strafing motion, passive deceleration is implemented to smoothly eliminate horizontal motion from the player when left or right strafing input is not provided.

For the rotation of the player, the player uses the mouse to point the direction of the ship. Vertical motion with the mouse simply tilts the players ship up or down. For horizontal motion of the mouse, the ship is tilted towards the provided direction as well as rotated as a typical plane would. For example, when the player moves the mouse to the right not only will the point towards that direction, but it will rotate such that if looking from a 3rd person stationary camera the plane would appear rotated 90 degrees clockwise.

Molecule Movement

Path - The collectible molecules will move following a fixed path starting at Zone 3.

Rotation - The collectable molecules rotate in place location for all levels. The rotation will be across all three axis to show all sides of the molecule.

Collision Handling

Collision detection is handled by Unity's engine. Colliders have associated scripts describing their behavior on collision.

Time Counting

The 3 min time limit for each level is calculated using Unity's Time class.

5. Technical Aspects

System Requirements:

- OS: Windows XP+, Mac OS X 10.6+, Ubuntu 10.10+, SteamOS+
- Graphics card: DX9 (shader model 2.0) capabilities; generally everything made since 2004 should work.
- CPU: SSE2 instruction set support.
- Web player supports IE, Chrome, Firefox, Safari and others.

6. External Resources

1. [Unity Web Player Licence Agreement](#)
2. [Unity3D Suite Licence Agreement](#)
3. [Unity Free vs Pro Licence Comparison](#)