



**Project Document**

An Engine That Does Stuff

The Team That Does Stuff

|  |  |
| --- | --- |
| **Team Name:** | **The Team That Does Stuff** |

|  |  |  |
| --- | --- | --- |
| **Name** | **Student Number** | **Preferred E-mail Address** |
| **Nathan Sullivan** | 40059116 | csullivan06@qub.ac.uk |
| **Stewart Reaney** | 40038703 | sreaney01@qub.ac.uk |
| **Chris Wilson** | 40049073 | cwilson48@qub.ac.uk |
| **Timothy O’Neill** | 40062164 | toneill22@qub.ac.uk |
| **Huy Vu Le** | 40075766 | hle02@qub.ac.uk |

|  |  |
| --- | --- |
| **Project Overview** | To develop features for a DirectX rendering engine which make Phil happy. |
| **Key Outcomes** | * Select entities in the world * Move entities * Let entities move as a pack * Move entities in different formations * Allow entities to move towards or away from each other * Have entities be able to determine the most efficient path to a given location. * Have entities make decisions based on what they can sense in the game world. * Have mapped terrain with trees. |
| **Languages** | C++  DirectX  There was no other choice :[ |

**AI**

1. As a user I can select an entity in the world so that I can move selected entities.
2. As a user I can move a selected entity so that it can move in the world.
3. As a user I can select multiple entities in the world so that I can form a “pack” of entities.
4. As a user I can make packs move together so that I can move multiple entities at once.
5. As a user I can make packs take up formations so that they can move in different ways.
6. As a user I can make entities chase other entities so that I can have entities that chase other entities.
7. As a developer I can assign steering behaviours to entities based on the following behaviour states so that I can see a living ecosystem.
   1. Seek
   2. Flee
   3. Arrive
   4. Align
   5. Velocity Matching
   6. Look where you’re going
   7. Wandering
   8. Path Following
   9. Separation
   10. Collision and obstacle avoidance
8. As a developer? I want agents in the world to be able to determine the most efficient path to a given location in the game world, so that entities appear to move through the world in logical and natural ways.
9. As a developer I want agents in the world to be able to make decisions based on what they can sense around them, so that entities appear to act in a logical and natural way. Examples of results from decision are:
   1. Flee from a perceived threat.
   2. Investigate a suspicious noise.
   3. Interact with friendly entities.
   4. Patrol an area with some degree of randomness.
   5. Attack an enemy.
   6. Flee from a losing fight.
   7. Investigate a point of interest.
   8. Become concerned if something has happened without it noticing?
   9. Seek cover in a fight.
   10. Use resources such as ammo when needing to reload a gun or a health kit if low on health.
10. As a developer I was to create more complex behaviour structures to allow for more sensibly behaving characters.
11. As a developer I would like to use a relatively simple scripting language to construct behaviour FSMs rather than creating them in code.

**Sound**

1. As a developer, I want to able to playback sounds so that a player may hear them.
2. As a developer, I want to be able to set a sound's volume based on distance so that sounds from further away are quieter.
3. As a developer, I want to be able to set a sound's position in stereo field based on position so that sounds coming from a particular direction seem to do so.
4. As a sound designer, I want to be able to apply effects to a sound to alter it so that reverb and EQ can be applied to make the sound environment more realistic.
5. As a sound designer, I want the sound to be unable to clip (output above 0db) so that I can add many sounds without causing distortion.
6. As a sound designer, I want to be able to apply the same effect to all sounds in a space to make the soundscape consistent in a given area.
7. As a sound designer, I want to be able to synthesise sounds from MIDI input so that I can generate sound effects and music without loading in pre-rendered audio.
8. As a developer, I want to be able to apply a 'material effect' (1) to an object so that the reflections are affected by the material it is echoing from.
9. As a developer, I want to be able to use convolution to apply material effects to sounds so that these effects are applied in a realistic way.
10. As a developer, I want to be able to use binaural filtering (2) to position a sound in a space in a realistic way.
11. As a developer, I want to be able to accurately model early, middle and late reflections (3) to give a more realistic impression of a space.
12. As a developer, I want to be able to use spherecasting to accurately model real sound physics (4).

(footnotes)

(1) A 'material' would be a set of effects that would affect a sound bouncing from the object the material is applied to.  For example, a carpeted surface would reduce the volume of the sound.  Sound heard through a glass surface would lose much of their high frequency content etc.

(2) This would involve having 2 'receivers' as a counterpoint to the 'emitters' attached to objects; one for each stereo channel (left and right).  Equalization and differences in distance for each receiver would be algorithmically determined to accurately model sound as heard in the real world.

(3) This would use raycasting to create reflections with material effects to model real world reflections.

(4) The idea for this is to create a sphere of pre-determined complexity and scale it up at the speed of sound.  Its maximum size is determined by the volume of the sound (decided by a sound designer).  When a point on the sphere collides with a solid object, the original sphere will stop expanding at the point of collision (but continue on all others).  A new sphere will be created at the point of collision and continue in the same way; its maximum size is the predetermined original minus the distance travelled to the point of collision.  It will also store the material details of any surface it has collided with.  When a point collides with a receiver, the sphere is destroyed.  The volume of the sound the player hears is determined by how far the sphere has been travelled and the sound will be affected by the material value of surfaces it has collided with (one point may reflect from a surface many times, the sound heard could be made up of thousands of these).  The delay between reflections and the direct sound combined with the material effects applied to the reflections should accurately model sound as it occurs in an actual environment.

**Terrain**

1. As a user I want to be able to see an expansive sea floor.
2. As a user I want to be able to see interesting coral reefs and seaweed to make the world more immersive.
3. As a user I want to be able to explore an underwater expansive world.
4. As a user I want to be able to on the surface of the water view realistic waves of water giving a feel of a realistic ocean.

**Displacement & Lighting**

1. As a user I can switch between texture and mesh of an object.
2. As a developer I can apply textures (bump map, displacement map) to create a type of material, and then I can use it on my mesh.
3. As a developer I can apply material effect (glass, stone, and sand)for each object
4. As a developer I can light the surface of textured 3D meshes to provide a range of material effects so that objects appear realistic.
5. As a user I want to be able to see complex heighted and textured objects such as rock.
6. As a developer I want to be able to assign what type of material to each object with ease.
7. As a developer I want to be able to create different kind of lightning effects ( diffuse, ambient, specular, caustic, reflection, and refraction)

**AI**

1. Given a clickable entity in the world, when I click on the entity, then it will be selected.
2. Given that I have a selected entity, when I use [movement controls], then it will move.
3. Given multiple clickable entities in the world, when I [something] and click on the entity, then it will be added to the selected entities.
4. Given that I have a pack of entities selected, when I use [movement controls], then they will all move simultaneously.
5. Given that I have a pack of entities selected, when I use [formation controls], then they will move into a formation.
6. Given that I have entities set to chase other entities, when I move one entity, then the other will chase it.
7. Given that I have entities set to move away from other entities, when I move one entity towards another, then the other will move away.
8. Given that I have entities set to crowd around another entity, when I move the crowded entity, then the others will all move after it.
9. Acceptance: Given an initial position, a target location, and an arrangement of obstacles; an entity in the world will perform a queue of movements which will have them arrive at the target location in the shortest amount of time.
10. Acceptance: An entity in the world with react to stimulus around it. For example a noise being emitted within the entity’s hearing will make the entity try to investigate the source, potentially by moving towards it in an alerted state, or if the entity see another entity which is deemed a treat it may run away from it.
11. Acceptance: An entity’s higher state machines levels will override lower levels of the hierarchical structure. Upon seeing a predator for example any state will defer to the flee state.
12. Acceptance: A function to read a script runs and creates a state machine which entities can reference for their behaviours.

**Sound**

1. Run an executable and hear a sound played back
2. Run an executable and hear 2 sounds.  One is near, one is far.
3. Run an executable and hear 2 sounds.  One is on the right, one is on the left.
4. Run an executive and hear 2 sounds.  One is dry, the other has a reverb effect applied.
5. Run an executable.  One sound will play repeatedly and more instances of it will be added every 1 second.  The sound will never distort.
6. Run an executable.  In one area, all sounds will have one reverb effect applied.  Move to another area.  The sounds here will have a different reverb effect applied.
7. Run an executable.  It will load a MIDI file and use the synthesizer to play it.
8. Run an executable.  A ball will bounce off a metal wall onto a wooden floor.  The sound of hitting off the different surfaces will be noticeably different.
9. Run an executable and hear 2 sounds.  One will be dry, the other will have a convolution reverb effect applied.
10. Run an executable.  A sounds will move from the left to the right and will seem to do so in a realistic manner.
11. Run an executable.  An object in a space will make a sound.  The sound will seem to occur in a real space and the sound will alter based on the player's position in the space.
12. Run an executable.  An object in a space will make a space.  The Sound will seem to occur in a real space and the sound will alter based on a player's position in a completely realistic way.

**Terrain**

1. When I load into the world I can view an expansive sea floor beneath me.
2. When I view the sea floor I can see multiple coral reefs and seaweed that looks appealing and nice.
3. When I enter the world I want to be surrounded by water that looks visually pleasing.
4. When I am viewing the world from the surface I should be able to observe realistic wave behaviours.
5. Given that a texture has been applied, the user can toggle between mesh and textured mesh.
6. Given that textures have been applied, the console will display which textures have been applied to which material.
7. Given that the surface is lit, the user can see it reacts to the light source realistically.
8. Given that material object created on the screen, the user can see its effect.
9. Same as (4)
10. Given that a type of material is created, it can be applied to different objects with different mesh easily.
11. When I am view the world, I should be able to see different light effect towards different conditions:
    * On the water: the water should reflect the skybox and other objects.
    * Under the water: caustic and refraction lightning should be seen at the bottom of the sea.

The project component of the module is divided into a number of two/three-week sprints, as shown below. Each sprint will commence with a Sprint Planning Meeting and conclude with a Sprint Review meeting and Sprit Retrospective.

**Sprint 1**

**Sprint 2**

**Sprint 3**

**Sprint 4**

**Meeting Meeting Meeting**

**Submission and Demo**

Each Sprint Review will consist of a brief demonstration of development to date. The final submission, at the start of Week 12, will comprise the game engine component(s) alongside a suitable test harness enabling acceptance testing and product demonstration. The project will conclude with an in-depth demonstration of all developed component(s).

**Nathan & Stewart**AI – 1, 2, 7a, b, c

**Tim**Terrain – 1

**Chris**  
Sound – 1, 2, 3

**Vu**Displacement – 1, 2

Everything we committed to was completed.

**Start Doing:**

Stewart feels he needs to start focusing on current user stories rather than spending time on problems from the previous project. Potentially resume daily meetings to keep everyone up to date on everyone else’s progress.

**Stop Doing:**

There was nothing we feel we should stop doing.

**Continue Doing:**

Everything else. We worked well and managed to complete all the stories we committed to. Everyone was working well independently and the code works in a modular way. This sprint went well.

**Nathan – AI** 3, 4, 7d, 7e **Stewart – AI** 9 **Timothy – Terrain** 4 **Chris – Sound** 4 **Vu – Displacement** 2, 3, 4, 5

Nathan’s AI stories were completed. Stewart’s AI story did not have enough time dedicated and was left 90% finished.

Tim’s terrain story, creating water & waves, was not completed, and left at around 70% finished, to be finished next sprint. Terrain was fixed from last sprint.

Chris’ sound story was completed, albeit with less than intended functionality.

Vu’s four user stories were completed, but with minor issues in displacement mapping.

**Start Doing:**

Better communication. Tim was working on a technique which used code Vu had already implemented, and none of us realised.

Start working earlier in the sprint. Most of us started working more in the latter half of the sprint instead of keeping a consistent work amount throughout the entire sprint.

**Stop Doing:**

Playing games during hours we should be working. Earlier in the sprint some of us were spending lab time playing rather than working.

**Continue Doing:**

Working in close proximity. We tend to all work around the same times in the minor lab and area able to help each other with minor problems.

**Stewart – AI** Continue work on 7, begin 10, and do 11 if time permits.

**Chris – Sound** Improvement of 2, 3, 4

**Timothy – Terrain** continuation of 4, 2

**Nathan – AI** Improvement of 4 (flocking), 5, 7j,   
 [no story] maybe add some additional behaviours like patrolling.

**Vu – Lighting 7**

[[Provide details of what was done during this sprint against that set-out within the planning process]]

[[Reflect on the sprint. Which aspects did you feel went well? Which aspects did not go to plan? How you did you act on your intentions from the last sprint (if applicable)? How do you plan to do things differently for the next sprint (if applicable)?

[[Specify which user stories within the product backlog you intend to fulfil within this sprint. You should decompose each user story as needed and provide an account of expected time expenditure]]

[[Provide details of what was done during this sprint against that set-out within the planning process]]

[[Reflect on the sprint. Which aspects did you feel went well? Which aspects did not go to plan? How you did you act on your intentions from the last sprint (if applicable)? How do you plan to do things differently for the next sprint (if applicable)?