LUMENS

PROJECT REPORT 1

Eoghan O’Keeffe – 08543453
BSc (Hons) Entertainment Systems, 4th Year

TABLE OF CONTENTS

SECTION I – OVERVIEW

1 - Concept

A light-emitting protagonist navigates a dark environment, home to lurking swarms of phosphorescent entities, which are attracted to and feed on light. These predators flock passively in the darkness until they are lit; they then leave the swarm to pursue the source of light and consume it.

In order to survive and progress through the environment, the protagonist negotiates its predators in the dark, skirting the edges of the ever-moving and swirling swarm, drawing away manageable numbers of individuals and eliminating them subtly. It must strategically hunt, lure and pick apart the swarming enemy without themselves becoming prey to their overwhelming numbers.

The dark conceals the details of the environment and its other occupants, and the player is never sure of exactly what lies outside their protagonist's small pool of light – the ever-moving swarm of glowing predators can be an unpredictable and deceptive threat.

Everything in *Lumens* is emergent – the environments the player navigates, the behaviour of the enemies they face, and even what they see and hear – and nothing is predictable.

2 - Game Flow Summary

The player negotiates the game's core mechanism – a balanced negative feedback loop, where the more aggressive a player is, the more difficult the game becomes – in pursuit of their objectives – speedy and skillful progression.

There are two main components of this mechanism; the first is how light affects the game and its dark environment:

- Enemies hunt out the source of light when they are lit by it
- The player can only attack those within range of their emitted light

The second is the effect of the player’s activity in the game:

- The range of the player’s emitted light grows with each attack
- This range is only reduced back to its minimum again slowly over time
- The minimum range is increased for each enemy eliminated

The player balances their objectives with survival and risk limitation.

They must get in close to the swarm in order to mount an attack; they risk attracting overwhelming numbers of predators at once; each attack increases this risk as the pool of light grows; the risk abates over time, but the goal of quickly eliminating the swarm encourages the player to continue attacking; finally, the minimum radius of light grows inversely to the number of predators remaining, keeping the overall level of threat stable as the player progresses.

This mechanic promotes strategy and controlled risk, as well as speed and skill, and makes the game more interesting, challenging, and resistive the harder the player tries. The swarm is an organic, unpredictable, and mobile enemy – the player must advance, but not recklessly. Frantic and aggressive play is likely to result in the player facing impossible odds, whereas overly cautious and reserved play will not result in a fast completion.

3 - Feature Set

The game features:

- Artificial intelligence modelling collective, emergent flocking behaviour (*Reynolds, 2001; Buckland, 2005*)
- Convincing real-time light simulation, with rays, shadow-casting, reflection, refraction, caustics, and after-image, which will form the core of the visual effects
- Soft body physics, with collision-detection and influencing light rays
- Procedurally-generated levels
- Interactive, emergent audio, generated by and enhancing the gameplay
- Variety of antagonists, abilities, and settings
- Support for multiple devices, with elegant solutions for smaller, touch-screen and re-orientable interfaces
- Persistence – saving and loading
- Web-accessible high scores and leaderboards

It aims to emphasise procedurally-generated features – effects, animations, physical simulations, behaviours, audio, and environments should all be largely algorithmically generated at run-time, and react to the user and the game. Pre-defined resources are kept to a minimum, and the experience instead emerges from activity within the bounds of the game environment.

4 - Genre

Action game with shooting and strategy elements. 2-dimensional, with a top-down or side-on view.

5 - Look and Feel

The aesthetic experience of the game is dynamic and emergent – generated according to some initial state or input reacting to the physical rules of the game environment – and there are few if any pre-created resources which are used as-is to create the experience.

Light simulation plays the central visual role in the environment – everything interacts with the rays emitted by the protagonist, and these effects combine to produce the main visual element of the game.
The protagonist casts an even, circular pool of light all around it from a bright, glowing coil at the center of its body. It also casts a stronger, narrow beam in the direction it is facing, which shows what it is aiming at.
Objects within the pool of light block, reflect and refract rays, casting shadows and caustics in the environment.

Significant objects (obstacles in the level and enemies) beyond that pool are phosphorescent, having some part of them that glows in the dark; they also react to the distant light, glowing brighter when facing the light.
In the absence of these lights, everything is dark and concealed.
The "living" entities are soft bodied (jelly- or raindrop-like), changing shape as they move or collide with other objects. This effect changes the way light travels around and through their bodies, and the shadows and caustics they cast as a result.
When the protagonist attacks, it fires a ropey tendril in the direction it is aiming. This tendril as long as the radius of light, and unfurls from the protagonist's body until it either hits an object or reaches its limit; it is then retracted, pulled tight and back into the protagonist's body, like a rope jerked back from its end.
Key effects accumulated from all this include the glittering, swarming predators shining and flashing as they collectively move, like bright fish in a "bait-ball"; the dramatic and interesting shadows and caustics constantly changing in the pool of light; the mysterious semi-concealed environment and its occuppiers in the dark; and the elongated trails and morphing bodies of the soft bodied entities.
Colours will be kept subtle and appropriate, with only key points of interest being given a highly-saturated primary colour – such as the protagonist's light-emitting core, and the predators' glowing.

The auditory aspect of the game is similarly generated in-game, and forms an integral part of the experience – it is often a sidelined aspect of games.
A simple, subtle, metronomic beat plays as atmospheric background music, and a basis for the generated sounds to be layered over.
Each significant action in the game has its own simple sound, which grow louder the closer they occur to the protagonist – for example, the predators each play a sound constantly, which grows louder with their speed and angular velocity, and so changes in time with and describes their movement. Various sounds exist for entity motion, state changes, damage, attacks, and deaths; menu and HUD (Heads-Up-Display) actions; and more.
These sounds are largely simple, repeatable notes – samples from real-world instruments, or digitally synthesised notes – which can combine to produce a fusion of sound over and in step with the metronomic background beat. They react to the action in the game, and so the music describes it – they take the place of "normal" sound effects.

Finally, other considerations for the "look and feel" of *Lumens* include simple and absolutely minimal menus and HUD – everything will preferably appear on the same screen, but slide in and out of view as needed – with simple and appropriate typeface, reacting to the light of the game screen beneath; haptic feedback (used sparingly) during significant events, for game controllers and other supporting devices; and indicative guiding visual cues to aid user interaction on touch devices – consisting of a subtle, glowing ring appearing around the positioning finger, and a glowing anchor point on that ring for the directional finger, which follows the angular motion it traces (see *Section IV: 1.2* for a detailed explanation of the touch-screen input system).

6 - Target Audience

Lumens is a game particularly suitable for short bursts of play, with strategic, non-repeating scenarios and a strong, recognisable product identity. It is aimed at players looking for either quick diversions or longer sessions of use; who enjoy action and strategy games, unusual ideas, and memorable "interactive experiences".

7 - Project Scope

This is an individual project, with an agile software development approach. Each version will produce a new component of the project's modular architecture, as described in *Section VII*; it is expected, however, that these goals will adapt and change over the course of development, so they represent initial, speculative goals for planning purposes only. The expected minimum requirements to be reached by the end of development are outlined as *Iteration 1.0* in that section, with further goals outlined in the subsequent iterations.

By the end of the first phase of this project, I aim to have reached *Iteration 0.1* – that is, achieving a demonstrable working prototype of the core gameplay functionality, and development of the system components needed to achieve it. This involves the design of a suitable architecture for the system – which will be explored in the next document – and the developement of the basic components of that system – including AI, imput, and rendering with simple shapes, but not light simulation, level generation, etc. – as well as the investigation of and experimentation with each of the used software components and the first target platform.

In addition to the work involved in designing, programming and developing the components of this project, there is work which must be carried out in research and investigation of those unknown components and areas which I will encountering for the first time here, or have encountered little before. Numbering among these are real-time light simulation and soft body physics theory, approach, and implementations; the production of full engines to handle game logic, physics, rendering, and artificial intelligence; platform-specific concerns such as capabilities, user-interface, programming interfaces, development and production tools, and enhancements; and development of efficient systems for fast tweaking and balancing of the system, and platform abstractions, where possible. This investigative work will add substantially to the process, and I will be tackling many new areas through it.

SECTION II – GAMEPLAY AND MECHANICS

1 - Gameplay

1.1 - Game Progression

- The player progresses through stages
- Each stage is generated algorithmically, and becomes progressively more difficult, according to the level generator
- When a player completes the objectives at a stage, they transition to the next generated stage
- This process may carry on indefinitely until the protagonist is eliminated
- The player then sees their score in a scoreboard, and may begin again or load a previously saved game

1.2 - Objectives

- The player must eliminate the predators occupying the environment
- The faster they achieve this, the better their score
- Each enemy they eliminate gives the player a set amount of points, which is increased by a factor according to the score multiplier system
  - A successful attack starts the multiplier, which immediately begins reducing with time
  - Each attack following it has its reward points multiplied, and adds a little to the multiplier
  - Multiple attacks and attacks on more challenging predators are worth more points and add more to the multiplier
  - The multiplier may increase to quite a high level, but may only reduce by a much smaller amount over time before it vanishes – this smaller amount is increased at a fraction of the speed of the multiplier increase
  - The multiplier system promotes speed and constant activity to keep it going
- Counterbalancing these goals is the goal of survival – although speedy completion and maintaining the multiplier promote speed and activity, the more ctive the protagonist is, the bigger a target they are to their predators
- The game may also be enjoyed as a goal-less experience
- Further objectives will be explored in any additional modes that may expand the game in later iterations

1.3 - Challenges

- The swarm of predators and the environment they are in present challenges to the player
- The predators hunt the player just as the player hunts them
- Depending on the type and number of individual predators in the swarm, it behaves in different ways as a whole
- The swarm as a whole moves in dynamic and difficult to predict ways, swirling around the environment, presenting a moving, shape-shifting target and threat to the player
- As the player's activity increases, they present a larger target to the swarm
- The dark environment partially conceals and distorts information to the player
    - Predators have different visibilities in the dark, which can misinform the player (see *Section III: 3.2* for further details on each type's behaviour)
    - The obstacles in the environment also have phosphorescent surfaces, but only those facing the player glow in the dark, so the full environment isn't always visible
- Procedurally generated stages present additional challenges
    - Players may never encounter the same one twice, meaning they cannot "learn" a particular stage, and are always "blind" to the layout of the environment and what it contains
    - The level generator produces progressively more challenging stages, varying environmental complexity, size, the number and types of predators, and other factors
- In all, the dark conceals threats; the predators' various natures mislead and surprsie; the ever-changing swarm outnumbers, entraps and surrounds; the player's own activity makes them more vulnerable; and the stages are unpredictable and progressively more challenging

1.4 - Gameplay Flow

- The player enters a new environment, casting a small radius of light and appearing at a safe distance from the predators dwelling there
- They immediately begin navigating the environment and seeking out the swarm, exploring the darkness
- The swarm moves obliviously in the dark, its members unaware of the player until they are lit
- The player moves towards the swarm to begin eliminating predators, trying to lure away and attack small numbers of them without becoming overwhelmed
    - The number of the predators' glowing cores and their movement offer a clue as to what type they are, how difficult they will be to defeat, and how large a reward they will leave if beaten – in addition, the number and brightness of the cores they have indicates how much health they have remaining
    - Although only the predators lit will attack the protagonist, others may follow them, also become lit, and join the attack – the more attracted at once, the greater their pull on the rest of the swarm
    - The player will usually try to attract as few predators as possible at one time
    - As the player's radius of emitted light grows with their activity, it becomes more difficult to avoid accidentally attracting unwanted extra attention – the light cast is a growing, uniform shape negotiating an irregular, changing, difficult to predict shape at close range, which may surround or cross it at any time
    - The player may also succumb to the tricks of the environment and predators, and become trapped in an unexpected series of obstacles, or be caught off guard by deceptive types of predators (see *Section III: 3.2*)
    - The player may also target special members of the swarm, which carry collectible items that aid the player in certain ways
        - These predators have an additional glow, colour-coded according to the collectible item they are carrying, and differentiating them from their flockmates
        - The types and effects of these collectibles are detailed in *Mechanics: Interactive Objects: Collectibles* later in this section
        - This introduces a more targeted and selective hunting behaviour – if a player has taken alot of damage, for example, they may stalk a predator carrying a healing collectible (indentifiable by their secondary green glow) and wait until they are exposed at the edge of the swarm to strike, to reduce the risk of being attacked by others while injured
- If the player lures away an enemy, they attack one another
    - The enemy chases the protagonist down, trying to reach it and attack it at close range
    - The player fires at the attacking predator, and moves to prevent it getting close enough to harm them – they can outmanouevre most enemies, so attacking while backpedalling, and hiding behind obstacles may be useful tactics

    - This is often not a one-on-one confrontation, as many predators may attack at once
- The protagonist also has some special moves at the player's disposal, which all give temporary added power and advantage, but come at the cost of greatly increased profile after use (see *2.3(a): Enhanced* later in this section for details)
- If the player is finding it too difficult to negotiate the swarm with a profile that's too high, they may try and retreat to allow it to reduce over time
    - The profile reduces faster in inverse proportion to the current score multiplier
    - if they choose to retreat, the player sacrifices their score in both time and their multiplier bonus
- When the player has eliminated the swarm, or completed the stage objective (eliminating every member may prove slow and tiresome, so the objectives or mechanisms may be altered to prevent this after playtesting), they transition to the next stagethat the level generator produces
- This process may repeat indefinitely until the player is defeated (but, again, this may be altered at a later stage, according to playtesting results)
- The player then sees their final score, and may proceed from the beginning or a save point
- Currently, the player gets one "life", and when they are defeated, the game ends – but this may be revised to the following multiple life system later if neccessary
    - When defeated, the protagonist's light is extinguished, and the player moves their glowing ember to a safe starting position
    - After a short time, the protagonist's light reappears at its starting size at the new position, and play continues
    - This behaviour is similar to the "beam" special move described in *Mechanics: Combat: Enhanced* later in this section
    - This is allowed to happen only a few times, before the protagonist is permanently defeated

2 - Mechanics

2.1 - Physics

(a) Light

- Real-time light simulation forms the visual and thematic core of *Lumens*
    - Backwards Ray-Tracing is the desired target for this simulation (*Owen, 1999; Wikipedia, 2011*), but other approaches – such as Photon Mapping, Shadow Mapping, or Metropolis Light Transport, and others – will be considered to achieve suitable performance and the desired effects
    - The desired effects of this simulation are rays and shadows, with good accuracy; reflection and refraction, to a low depth (number of times a ray can bounce around a scene); and caustics, to a low degree of accuracy (this will require some additional methods – such as bi-directional ray-tracing or mapping – and is a purely aesthetic part of the game)
    - After-image will be achieved in a final additional pass over the fully rendered scene, where the results will be processed to find the parts of the image brighter than a minimum threshold, and these will be persisted and redrawn for some time in subsequent frames
- Only particularly bright effects will cause after-image, to avoid cluttering the screen

(b) Particles

- Simple force-based particle physics, primarily for use in visual effects

(c) Rigid Bodies

- Force-based simulation of non-deformable objects – such as the environmental obstacles – and their motion

(d) Soft Bodies

- Force-based simulation of deformable bodies – objects whose shape varies with external forces and tends towards a stable shape
- The body's particles exert forces upon their neighbours until a stable state is reached, but may be affected by external forces, making the system unstable again
- Need to support at least two applications – "blobby" closed shapes and "ropey" open ones – to a visually satisfactory level

(e) Collisions

- Full kinematic collision engine supporting soft bodies, rigid bodies, particles, geometry, and light rays

2.2 - Movement

- 2-dimensional, within the constraints of the physical laws of the environment and the entity's own propellant and turning forces
- The protagonist is directed by input from the player (see *Section IV: 1* for details specific to each platform)
- The protagonist may move and aim independently and in any direction
- Other entities are moved according to their AI
- Primary movement is done on the center of an entity, and the body of the entity then moves in relation to this
    - For a soft-bodied "blobby" entity, the particles defining the body's edges are pulled by a force acting towards the moving center, but do not instantaneously move with it
    - For a rigid-bodied entity, the body moves along with its center
- The particulars of movement vary for different entities and an entity's different states (see the *Behaviour* parts of Section *III: 3*, and *Section V* for details)
- When the viewport (the area of the screen in which the player can see the game) is smaller than the environment, the viewport follows the protagonist's movements until reaching the limits of the environment – it is bound by a "spring" force with a maximum threshold to the player's position, and collides with the edges of the environment

2.3 - Combat

(a) Protagonist

– Normal

- The player adjusts their aim – indicated by a strong beam of light emanating from the protagonist's center along its heading – towards enemies in range
- When they attack, the protagonist fires an unfurling soft bodied tendril in the direction they are aiming
  - The leading particle of this tendril is directed, and the trailing ones are pulled along, and subject to the soft body physical rules governing the tendril
  - When the leading particle either strikes an entity or obstacle, or the overall length of the tendril reaches its maximum (the radius of light cast by the protagonist), the tendril is retracted – the last particle is pulled back, pulling the next ones, until it reaches the protagonist's center, then the next one is pulled, and so on
  - In order to ensure quick gameplay, this all happens very quickly, and the effect of the attack is applied as soon as the leading particle strikes something – the unfurling and retraction of the tendril trailing behind it is a visual effect, and does not take precedence over usability
- The player may execute multiple attacks in quick succession – there may be multiple tendrils extended at a time
- If an attack strikes an entity, a collision and physical reaction are applied to both, and if it strikes an antagonist, damage is dealt to it
- Every attack increases the radius of light cast by the protagonist
  - If the attack struck an antagonist, the increase is smaller, rewarding players for accuracy
  - If an antagonist is eliminated, the minimum radius is increased in proportion to those remaining

– Enhanced

- There are certain enhanced modes of combat the player can activate temporarily
- All can only be active for a short time, take a long time to become available again, and cause a large increase in the radius of cast light after use
- They are all activated by the player and held in this active state until the player releases them again, or until the maximum allowed time is exceeded
    - In order to correctly interpret the player's input, there may be a very slight delay before the mode is activated (see *Section IV: 1.2* for details)
- While in an enhanced state, the protagonist's element (the filament encircling its core) proceeds towards a different colour over time, a sound grows in volume and tempo, and there may be some other effects – these are the player's warnings of the limited and expiring time they have left in this mode, as well as the growing penalty to follow
- Upon exiting these states, the penalty in increase of the radius of light and the period of recovery for that enhancement are applied in proportion to the amount of time spent in the enhanced state – the light grows slightly beyond its bounds, then retracts to the new radius (attracting nearby predators and get the game under way again quickly)
- The protagonist contains glowing cores for each mode, which darken and brighten according to how fully charged each of the corresponding modes is, with the same colour code as that mode

- ### Beam
    - Can be used as a way to get out of a tight spot and evade pursuing predators, as well as an offensive weapon against them
    - The protagonist's light shrinks and disappears, leaving only the protagonist's glowing core and element visible – as a result, the player is no longer pursued by predators
    - While the protagonist is in this unlit state
        - It may be directed as usual, but receives a speed boost – the player may wish to find a safe spot in the environment to relocate to
        - When it makes contact with predators, it is unharmed and severely damages them – the player may aim for predators in order to eliminate them, and the usual rewards for attacks are applied
        - Its element colour proceeds towards red
    - Upon exiting, the radius of light suddenly and rapidly reappears and expands

- ### Repel
    - The protagonist emits a pulse from its core, causing a repellant force on all objects within its range – a defensive weapon

        - The protagonist's element colour proceeds towards purple
        - Upon exiting this state, the repellant pulse is released from the protagonist's core, and grows rapidly outwards, the same shade of purple as the protagonist's core was upon exit
            - The pulse's final size and force are proportional to the time spent charging it up in this state
            - The force is stronger towards the center of the pulse
            - It applies this force in the direction away from the protagonist to any objects caught in it, and any predators receive some proportional damage
            - There may be some other visual and auditory effects
            - The radius of light simultaneously grows to its new size

        - Range
            - The protagonist's radius of light shrinks and the length of its aiming beam grows inversely by the same factor
            - Light expands around its aiming beam into a conical (or triangular, since this is viewed in 2 dimensions) spotlight
            - While the protagonist is in this state
                - The player may fire with the temporarily extended range
                - The damage done by each attack is increased, but the tendrils do not extend any faster, so take longer to reach more distant targets
                - The reduced temporarily reduced radius lowers the profile of the protagonist – though the spotlight also attracts predators, they are attracted into the line of fire
                - The protagonist's element colour proceeds towards green
            - Upon exiting this state, the usual penalties are applied, and the protagonist's light and range return to normal

(b) Anatgonists

- Predators attack in a similar fashion to the protagonist
    - They aim towards their target and fire probosces at it, then withdraw them
    - These probosces differ visually from the protagonist's tendrils, being a pair of simple extensions coming from their body, pulling the edges of the soft body, and having sharp, pointed ends
    - Successful strikes cause damage to the protagonist
- After a successful strike, the predator pauses for some time before attacking again – as though pausing to digest their food
- Predator combat uses variations of this system, depending on their type, state, and statistics (see Section *III: 3*, and *Section V* for details)

- Aiming and movement are handled by the predator's AI, as are attacks and special behaviours
    - The particulars and constraints of their attack range, speed, recovery period, damage, etc., are defined by the statistics of each predator
- If the protagonist is defeated, it emits a sustained, large burst of light
    - The predators in range cease attacking and begin a feeding frenzy
    - Each "bite" is similar to an attack, but at zero range, and reduces the size of the light and the protagonist's body

2.4 - Interactive Objects

- There are other objects in the environment with which the entities and environment react in various ways

(a) Collectibles

- These are special items which provide a recovering boost to certain statistics of the protagonist, and are identifiable by their colour-coded glow
- They appear larger depending on the size of the boost they offer
- They glow, and float around the environment, just like a particle
    - They are attracted towards entities that pass nearby
    - Predators also seek them out, and may pick up loose collectibles
- When carried by predators
    - The predator carrying them is identifiable, as the item's glow remains visible – this allows the player to target them when the item is particularly needed
    - The predator releases the item when destroyed, and it then floats around the environment
    - When a predator is created already carrying the item, the size of the boost is proportional to how difficult the predator carrying it is to defeat
- In order to collect them, the protagonist may move into contact with them or hit them with its probosces
- When collected, a sound is played and some particles are released

- #### Healer
    - Repairs some of the damage done to the protagonist
    - Glows green

- #### Score Multiplier Booster
    - Applies an instant, larger boost to the player's score multiplier
    - Glows yellow

- #### Light Dimmer
    - Applies an instant, larger reduction to the radius of light cast by the player

    - Glows white

- Enhanced Combat Booster
    - Applies an instant, larger boost to the recovery time for all of the protagonist's enhanced attacks
    - Glows purple

(b) Atmosphere and Particles

- Particles float around the environment; observe the physical, light, and collision rules of the environment; and may have a limited lifespan
- When significant events occur – such as a successful strike or consumption of a collectible – there may be some colored particles released, to indicate the event visually, show the point of contact, simulate debris, and so on
- There may also be some semi-transparent atmospheric dust particles, which have no limit to their lifespan, and are pushed around entities as they come close to contact, as though affected by the air around them

2.5 - Environment

(a) Interaction

- The environment contains a set of obstacles, which no other entities can penetrate – the boundaries of the game
- All bodies in the game collide with the environmental obstacles
    - Rigid bodies and particles collide plastically with the obstacles
    - Soft bodies squash up against them until their centre collides and stops, then elastically bounce away
    - Light rays intersect with them too
- They have infinite mass, and are immovable
- Aside from these obstacles, the environment also consists of a planar backdrop, which takes up the entire area of the traversable space
    - It provides a background to the game, and is particularly important for making the light rays striking it visible – without a backdrop to project upon, the shapes of shadows and caustic patterns could not be made out
- This background only interacts withlight rays

(b) Progression

- The environmental obstacles can just about be made out in the dark, reflective, glowing points on their surface reacting to direct light from a distance – the side facing the protagonist can be determined, but everything else is concealed
- The player may manoeuvre around, hide behind, and shield their protagonist's light with the environmental obstacles
- Artificially-controlled entities aim to avoid these obstacles

- As discussed in part *2.2* earlier, the viewport moves freely around environments larger than it, and collides with its edges, limiting the view to the traversible area

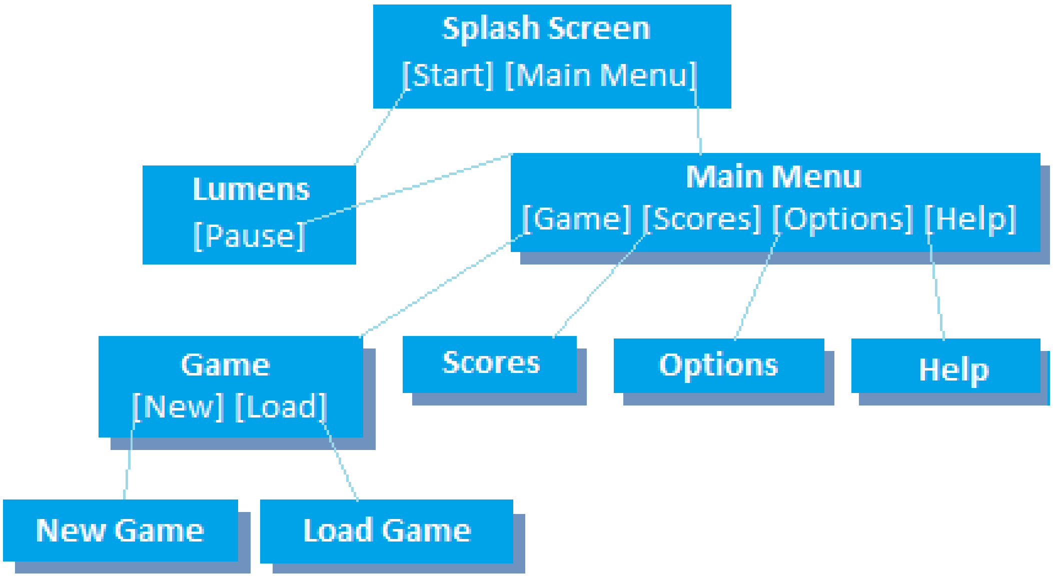
(c) Stage Generation and Transitions

- Stages in *Lumens* are generated procedurally, by a level generator module which produces them according to a set of criteria designed to ensure the level is playable
    - There cannot be too many obstacles crowding the space
    - Gaps between obstacles must be large enough for the player to pass through
    - To make the level visually appealing and interesting, functions which produce interesting and naturalistic graphical effects are used in the generation process – fractals may be used, or a generated field of 3D objects, which the protagonist moves along one dimension of, and the obstacles consist of the cross-section of thes objects along the plane at that point (like an MRI machine)
- Transitions from the completed stage to the next one occur while it's being loaded
    - After a player completes a stage, the transition is ready to be started upon the player's prompt
        - The protagonist's light is more vibrant and colourful, and pulses and scatters
        - The player's score and other details are displayed over the viewport
        - The player may direct the protagonist as usual in the background, and if left alone, the protagonist wanders around alone (see *Section V: 3* for details on this wandering)
    - If the player chooses to advance to the next stage, the score information disappears, the level begins to load, and the transition begins
        - The protagonist flashes a burst of light, and suddenly moves at great speed along the axis pointing away from the camera, leaving the level's image pulled and distorted at the point of departure
        - A trail of light and motion blur effects trail behind the protagonist, giving a sense of speed and movement from one place to the next
        - The player can still guide the protagonist around the viewport in this state, to alleviate boredom from any long waits
        - If it proves suitable, the stages will be based on a large fractal, and the protagonist will move towards another region of it upon transition, which is visible as they zoom towards it – in this case, the player may direct the protagonist to some desired area of the fractal
    - When the level is loaded, the protagonist loses speed and light intensity, and arrives at the new starting position

    - The level becomes visible as they approach, giving the player a brief overview of its layout and occupants, and allowing them to pick out a suitable starting point
    - The starting point is anywhere the player chooses, within the bounds of the environment, and not inside an obstacle
    - The protagonist lands at this position, distorting the level image upon impact and severely damaging any predators in the way
  - After a brief pause, play begins again

3 - Screen Flow

- The number of screens is minimal in *Lumens*
- All information is overlaid on the viewport, with the game visible and running (paused during gameplay) in the background, as described in *Section IV – User Interface: Visual System: Menus and Heads-Up-Display*
- This section deals with the progression of screens over the course of the game



- Splash
  - The protagonist moves around an empty environment, with the title *Lumens* used as the background material, slighlty too large to be lit all at once by the protagonist's normal light
  - At first start, its light is expanded to highlight the full title
  - The player may direct the protagonist or leave them to wander around, and can expand the light cast to highlight the title at any time
  - When the player chooses to move on, they may perform any short input (tap a key, button, or the touchscreen, as opposed to holding down) to move straight into the most recent saved game (Lumens screen), or alternatively press pause to access the main menu
  - Menus appearing over the splash screen do not cause the background action to pause while they are open, unlike during gameplay, and the protagonist wanders as usual

- Lumens
  - Any menu overlays disappear, and the splash screen briefly remains, before the protagonist transitions as usual to the starting point of the level and play commences
  - See *Section II: 2.5(c)* for details on transitions between stages on this screen
  - At any point, the player may pause the game, and the pause screen will be presented

- Main Menu
  - From here, the player is presented the options to proceed to the game, scores, or help screens

- Game
  - The player is presented options to proceed to the new game or load game screens

- Scores
  - The player is presented with their scores and, if online, those of other players
  - These scores may be ordered ascendingly or descendingly by various categories, as described in part *4.2* below

- Help
  - The help screen is a scrolling screen with a navigation sidebar, which details all of the areas of the game which the user may require information on

- New Game
  - If it is suitable to keep multiple saved games, the player may choose a name for their new game and proceed to start it
  - If the name already exists for another saved game, they are prompted to overwrite it or change the name – if they overwrite it, they then proceed to start their new game

    - Otherwise, they simply proceed to a new game, overwriting their existing single game, without this screen

- Load Game
    - If it is suitable to keep multiple saved games, the player may choose a game to load, and then begin playing from the saved point
    - Otherwise, they simply proceed to their existing single game, without this screen

4 - Persistence

4.1 - Saving and Loading

- Depending on the outcome of testing of suitability and requirements, *Lumens* may either support a single persisted instance of a game, or multiple ones which can be managed by the player
- General accumulated statistics and information relating to level generation must be stored in a persited file
- The persistence system uses a platform-independant format to save information

4.2 - Scores

- Scores are accumulated over the course of a game, and stored when a game is ended or saved
- If an internet connection is available, the player's scores are synchronised with the global online scores – a limited number of global scores may be pulled down at this time as well
- They may be viewed at any time, and ordered according to various criteria – such as overall score, time spent playing, enemies killed, number of deaths, kill-death ratio, etc. – so the neccessary information must be gathered and stored over the course of a game

SECTION III – THEMES, SETTING, AND ENTITIES

1 - Themes

Lumens focuses on a few central ideas and themes:

- Light and revelation, darkness and concealment
  - The player should be in suspense, never quite sure of what they might stumble upon as they cut through the dark – think of the feeling you have walking in the dark, before your eyes adjust, anticipating a trip or collision
  - They should be surprised by what they thought they saw in the dark, revealed as some deception in the light
  - The properties of light are the primary visual focus of the game (or those properties highlighted herein)
- Stalking, luring, and finely-balanced competition for survival
  - The player carefully hunts a prey that could easily turn on them and destroy them in kind – they are prey too
  - They must be cunning and react quickly to pick apart a threat much larger tham themselves
- Negotiating the communal behaviour and overwhelming aggression of the swarm
  - A swarm behaves like an organism, the individuals' collective movement changing it in unpredictable ways
  - Managing and negotiating numbers which behave like this is a challenge requiring constant alertness and responsiveness
- The mindless attraction of simple creatures to light
  - This is the idea that began the whole project – watching a moth repeatedly throwing itself against a lightbulb, thinking of huge swarms of insects, and cave-dwelling troglobites which have lost their skin pigment and sight in adapting to total darkness
- An organic and unpredictable environment
  - Ideally, everything in *Lumens* should be generated procedurally, emerging naturally as the result of initial conditions passed into an environment of defined rules, with unpredictable outcomes (dynamical systems and chaos)

2 - Setting

2.1 - Elements

- The obstacles contained in the environment are 2D shapes, forming the boundaries of the game
  - They are immovable, and all entities and light rays collide with them
  - The player may be surprised and trapped by an unnoticed obstacle emerging from the dark, or a misjudged shape turning out to be a dead-end
  - They may also be used strategically

        - The protagonist can be hidden from the swarm by using them to shield its light
        - The predators can be funnelled, their movements better predicted, and kept at bay by the obstacles
- There is a 2D plane, which acts as a backdrop layer
    - Its reaction to light is the main point of interest – acting as a surface for light to be projected upon, and important for making the more interesting light effects (shadows and caustic patterns) visible
    - It does not react with any entity in the game, only light
- Atmospheric dust particles also float around the environment, picking up rays when lit

2.2 - Look and Feel

- The setting is simple, abstract, and pitch black without the protagonist's light
- The environmental obstacles define its shape
    - They are 2D, and made up of irregular and complex-looking surfaces – reminiscent of craggy rocks
    - These shapes can just about be made out in the dark, by glowing points on surfaces directly facing the protagonist (and not blocked by other obstacles)
    - The player should feel unsure when navigating the obstacles, whose glittering edges only give small clues the of their shape in the dark – like they are wandering around in a dark room littered with objects, and cannot quite make anything out, so explore cautiously with their hands outstretched, with the frequent feeling of being just about to walk into or stumble over something
- The backdrop plane is simple and visually inobtrusive
    - It may be an abstract bump-mapped surface, or reveal the subsequent stages (as if through a foggy window), depending on the level generation strategy used
- Atmospheric particles vary in size, are transparent, and appear bright when lit
    - They give the effect of dusty air or cloudy water

3 - Entities

- Note that the statistics offered in the Statistics subsections following are intended as rough rankings of the entities' abilities relative to one another (5 is high, 1 is low), and are not representative of any final balancing or even any applicable unit of measurement

3.1 - Protagonist

(a) Behaviour

- The protagonist is directed by the player

- It may move and aim independently and in any direction
- Attacks, both normal and enhanced, are subject to their own delays after use
- Pending the results of testing, fully automatic firing may be used (where the player can hold the fire command, and attacks occur continuously in sequence, with the delay between each one)
- During certain parts of the game (such as on the splash screen), the protagonist may wander around with no particular aim

(b) Statistics

- Speed: 4 (the maximum speed it can reach)
- Agility: 5 (linear and rotational acceleration)
- Resistance: 4 (the amount of damage it can take before dying)
- Damage: 3 (the amount of damage it deals when attacking)
- Range: 4 (the range of its attacks)
- Attack rate: 4 (the speed with which it recovers from attacking and is ready to do so again)
- Size: 3 (its overall size)
- Swarm influence: 5 (the degree to which predators are attracted towards it – for predators, this influences how closely the swarm follows them)

(c) Look and Feel

- The protagonist has a circular soft-body, which is normally a “frosted” white colour
- Its core is a glowing gold element (like a light bulb element), surrounding the source of its aiming beam which comes from its center
- This core is the source of its bright white light, its gold aiming beam, and usually the only source of (ray-traced) light in the environment
- Its other glowing cores correspond in colour to each of its enhanced combat modes (red, purple, and green), fade and grow brighter according to their level of charge, and are semi free-floating in its body, located towards the rear and either side
- During attacks, it fires tendrils from its body, which meet seamlessly with the edges of its form and are the same white in colour
- These tendrils are ropey, soft, and thin, with a small node at the end
- They are fired and withdrawn as described in *Section II: 2.3(a): Normal*
- The protagonist moves with great agility and good speed, and should feel very responsive to control
- Upon death, the protagonist's glowing cores fade away, its element becomes a dull glowing orange, its aiming beam vanishes, its central core explodes, and a huge, sustained burst of light is released

        - While the predators are in their feeding frenzy, the protagonist's body reduces in size with each bite
        - When the light is extinguished, all that's left is the protagonist's element floating around the environment

3.2 - Antagonists

- All predators observe a variation of the same set of common behaviours
    - When unlit, all follow the swarm of their fellows according to the rules outlined in *Section V: 1*
    - They also wander, so that they are not stationary when alone, and to add further unpredictability to the swarm's overall behaviour
    - When lit, they all attack the protagonist
    - None are affected by the protagonist's cast light beyond its range – that is, they do not see it from a distance, but rather “feel” it upon contact
    - They do not have memories – when lit and then unlit, they simply return to their wandering or swarming state
    - Unlike the protagonist, they cannot move and aim independently, and must turn to face their targets
    - Just before striking, they betray a small clue as their probosces emerge slightly, giving the player a chance to evade the attack by moving out of the way before they can turn to face them again, in which case they strike and miss – the delay is proportional to their attack rate
- Differences in this behaviour arise either from differences in their statistics, or explicit differences in their AI
- They all have common traits while also being very distinct from each other, giving the impression of being different varieties of the same species, or the same species at different stages of development
    - Soft bodies with a slight purple-red hue
    - Containing semi-free-floating phosphorescent core(s) (the number proportional to their current health)
    - They have red spot(s) of pigment on their skin (the number proportional to their initial health), with generated shape and laid out symmetrically along its rear-front centerline

(a) Fly

- Behaviour
    - The fly is the basic and most common type of predator, and all of the other predators are based upon it
    - It follows the swarm when unlit, and pursues the protagonist when lit, attacking when in range and aiming correctly

    - It is small and weak, but exists in very large numbers, is the fastest and hard to hit or outmanouevre, and because of its limited renge, will get in very close to attack – many of these will smother the protagonist

- Statistics
    - Speed: 5
    - Agility: 3
    - Resistance: 1
    - Damage: 2
    - Range: 1
    - Attack rate: 3
    - Size: 1
    - Swarm influence: 2

- Look and Feel
    - The simplest and smallest of the predators, it is a soft-bodied, elliptical entity with a slight hue
    - It has a single pigment at its mouth, and contains a single point of phosphorescence
    - Its attacking probosces are short mandibles

(b) Hunter

- Behaviour
    - The hunter is only slightly affected by the swarm, the primary influence on its movement is actively seeking out the protagonist
        - It will separate from the main body of the swarn and search the surrounding area for the protagonist
        - It moves towards likely places, making forays along the environment, following the obstacle walls, and making searching criss-cross patterns of movement in open space
        - It will not leave its swarm entirely, but rather acts as something of a scout, staying at the periphery and making sorties further away, but trying to return and catch up when it strays too far
    - It is a challenging opponent on its own and in groups, intelligently searching for prey, and attacking with speed, agility, and evasiveness when it finds its prey
    - It is more intelligent than the others, and shows this in its searching and attacking behaviours – also, it can be considered a playground for creating smart AI within the limitations of the game rules
    - It can, however, be more easily drawn away from the swarm, and eliminated on its own

- Statistics
  - Speed: 3
  - Agility: 4
  - Resistance: 2
  - Damage: 1
  - Range: 3
  - Attack rate: 5
  - Size: 2
  - Swarm influence: 1

- Look and Feel
  - The hunter is – like the others – soft-bodied, but much firmer, not deforming much, with a strong hue, and has a slightly triangular shape, with twin mandibles forming two corners at its head
  - It has two pigments towards its head, and contains two glowing points
  - Its probosces come from between its mandibles, and are a long, deep-red, tongue-like thread, similar to the protagonist's, but thinner, moving more rapidly and directly, and with a more evident seam where they join at its mouth

(c) Strobe

- Behaviour
  - The strobe follows the swarm as usual when in the dark, but its movements are difficult to track
    - Its glow periodically disappears and reappears, fading in and out of view
    - It changes direction more than the other predators, and is more likely to wander off and appear somewhere unexpected when not glowing
  - When lit, it presents an indirect but significant threat if members of its swarm are nearby
    - It is slow and cautious, skirting the edges of the light and only reluctantly attacking the protagonist itself – when it does attack, it makes a deliberate move towards its target to get in range, then tries to retreat back to the edge afterwards
    - Rather, it retransmits the protagonist's own light from its own body, drawing other predators towards the protagonist
      - Its recast light has its own range and uniformity, and upon being lit, it gradually begins increasing these
      - It is most dangerous when at the edge of the light, attracting its flockmates and protected by their bodies as they rush to attack the

        protagonist, continually increasing the protagonist's profile, bringing large numbers in to attack, and out of reach – attacking by proxy
        - If the swarm contains many of them, a chain of light could also be formed
        - Its maximum and minimum ranges are decide by the protagonist's own ranges
    - It is a self-preserving and strong opponent, and a leader sending others to do its work

- ### Statistics
    - Speed: 2
    - Agility: 1
    - Resistance: 3
    - Damage: 4
    - Range: 2
    - Attack rate: 2
    - Size: 4
    - Swarm influence: 3

- ### Look and Feel
    - Its body is fat, loose and irregular, a kind of deformed version of the protagonist's circular light-emmitting body, with a medium-weak hue
    - It has 3 pigments on its skin – one at its mouth and two peripheral ones on fins to either side of its body – and three glowing cores
    - The light it casts is a duller, more orange colour than the protagonist's, and its body has a very slight red hue
    - Its probosces are a pair of thick, long limbs emerging from either side of its mouth, and not the stretchy threads of some others

(d) Behemoth

- ### Behaviour
    - This is the hidden boss, an unwelcome surprise in the dark
        - It moves slowly, surrounded by the swarm due to its strong influence on them
        - In the dark, it's difficult to tell if a cluster of glowing points is a group of individuals, or one large behemoth surrounded by flies – like seeing two headlights in the dark, thinking they are two motorcycles side-by-side, and discovering they in fact belong to a truck
        - As a result, an unwitting player may stumble across a big challenge unexpectedly
    - When lit, it is a big, slow, powerful fortress in attack

  - It cannot turn or move very fast, and its emerging probosces just before an attack are a quite visible warning
  - However, it causes a great deal of damage when it strikes successfully with its probosces
  - It also has some other strategies, subject to their own delay and range constraints
    - It is cannibalistic, and will occasionally consume other predators when damaged
      - This absorbs their health, glowing cores, and any collectibles they were carrying
      - The Behemoth turns towards an unsuspecting ally and pulls them into its mouth
        - During this time, it leaves the protagonist alone
        - If the player manages to destroy the other predator before the Behemoth slowly finishes swallowing it, the effect is prevented, and the player gains an additional boost to their score and multiplier, and the event is recorded in the score system
    - It has a secondary venom-spitting attack
      - This is more rapid and has greater range, but is much less damaging than its primary attack
      - Again, its probosces betray the onset of this attack, moving back away from its mouth to allow the venom to be fired
  - It relentlessly advances as long as it is lit, and brushes off most attacks easily
- It is the apex predator of the swarm, and is best faced evasively – as with any predator, beut doubly important here, since it takes far more time to kill it
  - Anticipating its attack cues is hugely important in dealing with the behemoth, as successful strikes are so damaging that dodging them is key to survival
  - Because of how large and cumbersome it is, use of the environment to evade it is an effective and important strategy – narrow gaps can be very helpful obstacles
  - Outmanouevring it and trying to stay out of its attacking line are hugely important
  - Leaving and returning to the fight may also be a neccessary tactic
  - Enhanced attacks and collectibles may be life-saving aids
  - Finally, the Behemoth will rarely be alone, so even if the player can easily outmanouevre it, whether they can do so when faced with other incoming predators is another question

- Statistics

  - Speed: 1
  - Agility: 2
  - Resistance: 5
  - Damage: 5
  - Range: 5
  - Attack rate: 1
  - Size: 5
  - Swarm influence: 4

- Look and Feel
  - The Behemoth's body is soft towards the rear, but firm and shell-like towards its head, has a more complex, definite bodily structure, with a strong hue, and is several times the size of any of the other entities
  - It's probosces form large, protruding claws at either side of its head, and it has a line of smaller, tooth-like probosces folding inwards towards its mouth
  - The venom it spits from its mouth is a bright green viscous liquid, which loses its opacity, speed and damaging power quickly, and glows in the dark
  - It has large pigments covering much of its skin, particularly on its claws and the hard areas towards its front, and contains several (5 or more) glowing cores, which may appear to belong to several different individuals moving together in the dark

SECTION IV – USER INTERFACE

1 - Input System

- Because *Lumens* is aimed at a variety of devices (and a variety of devices can use the web application), all of their input mechanisms must be considered
- Each must provide ways to execute the game's commands
    - Move
    - Aim
    - Attack
    - Special combat modes
        - Beam
        - Repel
        - Range
    - Pause
- Input for menu navigation is trivial, and will not be covered here
- All of the following confogurations have been devised with the player's comfort, familiarity, handedness, and personal preference and choice in mind

1.1 - Keys and Pointer (Mouse and Keyboard)

- The game may be played using either the keyboard alone or a combination of the mouse and keyboard
- The "up", "down", "left", and "right" arrow keys, and the equivalent "w", "s", "a", and "d" keys are familiar and widely used configurations for movement in games, so these direct the protagonist here - in addition, to better support left-handed players using the keyboard and mouse, the "i", "k", "j", and "l" keys will also perform the same functions
    - The left and right keys adjust the protagonist's aim anticlockwise and clockwise respectively
    - The up and down keys move it backwards and forwards along its aiming line, respectively
- The mouse may alternatively be used to adjust the aim (and is much faster), with each movement adjusting the aim relative to the mouse's last position (as normal in most games)
- To execute a normal attack, the space bar or left mouse button is pressed
- The three enhanced combat modes – Beam, Repel, and Range – are activated by pressing and holding down the "x", "c", or "v" key, or the "m", "n", or "b" key, respectively, for the desired duration
- The "p", "return", and "g" keys all toggle pause on and off

1.2 - Pointers (Touch-screen)

- With all supported multi-touch input systems, *Lumens* again uses a combination of familiar input configurations and strategies addressing the other concerns
- For the in-game controls (i.e: not the menu system), touch input is not limited to any particular zone of the screen
    - The player touches the screen, and further movements are measured relative to this initial point of contact
    - The order touches occur in is what decides which touch performs which in-game action
        - The first touch controls movement
            - The initial point of contact is encircled by a semi-transparent radius rendered on the screen, overlaid on the game so that it is clearly visible, but does not obstruct the image rendered below it
            - This circle represents the maximum applicable input, and denotes a zone on the screen, which affects what effect other touches have
        - The second touch outside the first zone controls aim
            - The initial point of contact is again encircled
        - Any third touch outside the two zones executes a normal attack in the direction the protagonist is aiming
        - If at any point in this sequence a touch occurs inside the two zones, a special attack is executed, for the duration of the touch (whether or not it subsequently moves out of the zone)
            - A touch inside the first (movement) zone activates Repel mode
                - Any touch outside the zones execute an attack
            - A touch inside the second (aiming) zone activates Range mode
                - Any touch outside the zones execute an attack
            - Two touches inside both zones (near-)simultaneously activates Beam mode
                - Since there is no way to fire the protagonists's probosces in this mode, two touches may be used for this mode, as no attacking touch is needed
    - This system achieves several goals
        - The input controls do not get in the way of what the player wants to see, since they may touch the screen almost anywhere they wish
        - It supports handedness and other issues particular to the player in question through its flexibility
        - It requires a maximum of four simultaneous touches to execute all functionality
        - It should be very intuitive, flexible, and forgiving
- The game may be paused by touching the protagonist

1.3 - Buttons (Gamepad)

- The touch-screen system is inspired by gamepads with dual analogue sticks – just made more flexible to take advantage of the medium – and so these devices use a similar system
  - The left analogue stick controls movement; the right, aiming
    - These may be swapped in the options menu for left-handed players
- The shoulder buttons (those controlled by the player's index and/or middle fingers) control combat on devices with 4
  - The top-right executes an attack
  - The top-left activates Range mode
  - The bottom-right activates Beam mode
  - The bottom-left activates Repel mode
- The face buttons (the four on the right, pressed by the thumb) and directional buttons (the four on the left) provide fallbacks with some of the same functionality
  - The top face button activates Repel mode
  - The bottom face button activates Beam mode
  - The “up” directional button moves the protagonist forwards along iits aiming line
  - The “down” directional button moves the protagonist backwards
  - The left and right face buttons, and the “left” and “right” directional buttons adjust the protagonist's aim anticlockwise and clockwise, respectively
  - This is useful both for a player's preference, and for devices with fewer buttons
    - On those with only one analogue, the aiming may be done with the face buttons
    - For those with no analogue sticks, both aiming and movement can still be performed with these fallbacks
    - On devices with only 2 shoulder buttons, the Range and attack commands remain in place, but Repel and Beam are only activated by the face buttons
- The usual pause button for the platform in question is used to pause the game – the “start” button on a PlayStation 3 controller, for example

1.4 - Other

- Other devices – such as the Wii controller and the XBox Kinect – may also be supported in the future

2 - Visual System

2.1 - Viewport

- The viewport is the visible area of the game environment – the environment's area may be larger than this, but it is not rendered to the screen outside of it
- It displays the game, menus, and information to the player
- It is elastically bound to the protagonist's position, and collides with the edges of the environment

        - As the protagonist moves, it exerts a force on the viewport to pull it along with it, subject to a maximum distance constraint between them which cannot be exceeded
        - If the edges of the environment meet those of the viewport, they collide, not allowing it to travel any further than the edge of the playable area

2.2 - Menus and Heads-Up-Display

- Menus and the HUD take up minimal visual space during the game
- Menus appear overlaid on the viewport
    - They have a semi-transparent white foggy background, covering the area around the text, blurring the view beneath, but not entirely obscuring it
    - The text has a coloured glass effect, the light of the protagonist making it glow a strong and almost opaque colour, and casting some subtle crepescular rays through it
        - Its colour changes over time between colours that contrast strongly with the background fog
        - For the sake of legibility (and trying not to be too ridiculous), smaller text may omit some of these effects
    - These menus arrive and are dismissed like condensation appearing on glass when it's breathed on, then fading (though quickly), and are accompanied by a sound
    - There are some functions that appear on every menu screen
        - Exit: leave the game
        - Paused: unpause the game, displayed in the upper-left corner
        - Back: return to the previous screen (unpauses the game for the root screen)
        - Score: the current score and multiplier, if any, are displayed in the upper-right corner
- The HUD should be minimal or not required at all in this game, and is mentioned here to emphasise that idea
    - The following are the statuses which need to be displayed
        - The protagonist's health
            - The colour of the protagonist's center (the glowing point from which its light originates) and its aiming line proceed from gold to dull orange as it sustains damage
            - Its cast light follows, though always with a much higher lightness value (whiter)
        - The amount its enhabced combat modes have charged up
            - The three coloured glowing cores – one corresponding to each mode – are within its body
            - They fade when the corresponding mode has been depleted, and grow brighter as it recovers
        - The score and current multiplier

        - These must be displayed in order to be accurately represented, but are shown as innocuous menu-style text in the upper-right corner
        - However, the status of the multiplier is quite well represented by the auditory cues which accompany it, and both are shown in the pause menu and at the end of each stage, so the player has the option to turn their display off in the options menu, and go by these cues alone

2.3 - Screen Sizes and Orientation Changes

- Because of the variety of platforms that this application is aimed for (and could conceivably be run by), a fluid layout model is employed to ensure that it displays well on a large range of screen sizes and aspect ratios
    - A fluid layout is platform agnostic, and uses information at run time to decide on ideal sizes, to avoid aiming specifically at each platform and losing generality
    - Ideally, all sizing information is expressed as direct or indirect percentages of the basic screen size information available on each platform
    - This avoids targeting each platform individually, improves the application's ability to adapt well into the future, its flexibility, and its ability to handle resizing events gracefully
    - These changes will largely affect the menus and scale of the game environment
- *Lumens* will respond to oreintation changes on platforms which support them, or the more refined accelerometer events, where available
    - The viewport will simply rotate to the new orientation, with the menus following – for this reason, menu bounding boxes will be square or circular
    - The game will also re-orientate, with the aim of giving the impression that the screen and viewport are a rotating window into a stationary environment below

3 - Audio and Music

- Dynamic audio will be used throughout the game
    - Each significant event will have a base sound attached to it
    - This sound may be modified and altered on the fly according to the particulars of the event
        - For example, entities emit a sound relating to their motion
        - The base sound is altered in proportion to the entity's linear and rotational speed and acceleration, growing in volume
    - The base sounds and alterations are carefully selected to be musically complimentary
        - This requires much experimentation
        - Sounds may be synthetic, instrument samples, or other types which suit the themes of the game
- A metronomic beat runs in the background, forming the structure of the audio
    - All sound will enter a pipeline, which is delayed until a beat is reached and it can be played

    - This keeps everything in time, and helps prevent the sounds turning into an unmusical cacophony
    - The metronome describes and underlines this tempo
    - Very significant events – advancement through a stage, numbers of predators getting very close to the protagonist – may affect the overall tempo, to create tension
- The audio pipeline has other controls to ensure the sound does not become too loud or chaotic
- The overall effect is generated audio which is unique, descriptive of the events, interesting, and musical

SECTION V – ARTIFICIAL INTELLIGENCE

Entities use simple finite state machines to govern the main parts of their behaviour.

1 - Antagonists

- Antagonists are all influenced by a variety of forces which combine to direct their behaviour, each weighted by their statistics, or handled by explicit programming
- The following behaviours are common to all predators (to differing degrees), and divided into the appropriate states
- Global state (behaviours always observed)
  - Wandering: movements which result in an indirect path being followed
  - Collision-avoidance: steer away from collisions with obstacles before they occur
- Passive state (while unlit)
  - The swarming behaviours
    - Separation: avoid crowding neighbours
    - Alignment: steer towards the average heading of neighbours
    - Cohesion: stay close to neighbours
- Aggressive state (while lit)
  - Pursuit: largely overriding other behaviours for most predators, they chase down the protagonist
  - Evasion: move away from the aiming beam when lit by it, change direction and speed up if hit
  - Anticipation: use the protagonist's current velocity to aim towards the point it will be at if it continues its current movement
  - Attack: when in range, begin attacking – this may involve more complex behaviours in and of itself

1.1 - Fly

- The fly's behaviour is simply governed by the above influences – it has no other special behaviours

1.2 - Hunter

- To achieve its more active scouting and hunting behaviour, the Hunter observes some additional guidelines
- Passive state
  - Searching for prey: scout out areas the protagonist is likely to be in – the likelihood of their presence is weighted by some simple observations
    - The protagonist is not likely to be where neighbours are already looking, so search some distance away from the swarm

- The protagonist may use obstacles for cover, and if they are near an obstacle, it is easier to corner them there than in open ground, so follow the edges of obstacles
- When covering open areas, use a zig-zagging, sweeping motion to cover ground better
- If alone, they Hunter is vulnerable and less likely to corner their prey, and if the protagonist is between it and its neighbours, it should try to force it into a “pincer” trap – so return towards the swarm if it is too far away

- Aggressive state
  - Advanced evasion: anticipate the aiming beam, moving away from it earlier
  - Stay away: keep to the edges of attacking range, as it requires more accuracy for the protagonist to strike successfully
  - Occasionally change between the following attacking strategies
    - Strafing: pick up speed, then drift and turn to face the protagonist while attacking, and repeat when speed reduces too much – lets momentum keep it a moving target while it attacks
    - Feint: make rapid changes in direction and speed
    - Rush: suddenly speed towards the protagonist while attacking, moving to “just miss” running into it, and end up on the other side of it – very hard to hit if it's moving quickly enough and catches the player off-guard, and lethal in groups

1.3 - Strobe

- Passive state
  - Unpredictability
    - Strobing: occasionally have glowing cores fade, reducing influence from and on neighbours to zero with visibility – may disappear when vanishing control is “charged”, must reappear when it “empties” and recover influence, but can choose any time in between those points at random – and while invisible, occasionally move in a wildly different direction at top speed before reppearing
- Aggressive state
  - Recast light: stay to the edges of the protagonist's cast light, so that the recast light travels as far an extra distance as possible, and keep moving with the rest of the swarm to try and attract them – only when being attacked, without nearby neighbours, or on infrequent occasions will it directly attack the protagonist, and it then immediately retreats by the fastest route to the edge

1.4 - Behemoth

- Aggressive state
  - Venom spit: if the protagonist is out of range, or on occasion, spit venom at it

- Cannibalise: if severely damaged, turn towards a nearby neighbour (using anticipation to judge the best candidate) and pull it (with probosces) to feed

2 - Protagonist

- While wandering, the protagonist observes the wandering and obstacle avoidance behaviours
- Pending playtesting, the protagonist may use an aim assisting behaviour

SECTION VI – TECHNICAL

1 - Target Platforms and Considerations

1.1 - Modern Web Application

- The initial target, this will utilise recently emerging web technologies – WenGL, improved JavaScript engines, local storage, and other HTML5 innovations
- *Lumens* will be developed to the W3C specifications, and will not be not expected to run on browsers which do not correctly implement them (looking at you, Explorer), nor will great effort be spent on trying to make it work across a wide range of current and older browsers – the focus is on modern browsers
- It should, however, support a wide range of modern platforms which may access it, including mobile and touchscreen devices
- Speed and performance limitations are a major concern, since this is relatively new territory, and the project contains several computationally heavy components – flocking, real-time light simulation, dynamic audio, and other physics
  - Solutions to optimise performance on this and the other platforms will be discussed in further documents

1.2 - iOS

- The ideal solution would be to wrap an internal web browser context in a native application, with a communication interface between them to support persistence
- If this affects performance too severely, the context does not have the neccessary features, or would violate the App Store Guidelines, then a full port may be required
- The App Store Guidelines and quality control measures must be closely adhered to
- Performance is an issue of even greater significance on mobile platforms

1.3 - PSP

- This platform will be aimed for as an attempt to make use of the access to development kits I have during the course of this project
- It will require a full port of the game, and very likely be a great challenge

1.4 - Android

- The considerations for this platform are similar to those for iOS, with the exception of the App Store Guidelines

1.5 - Windows

- A native Windows application would reduce the constraints on performance
- Implementing the existing web application (if it is successful) through a browser context may be a prefereble option in terms of development time

1.6 - Mac OS

- A native Mac application has the same considerations as Windows, with the addition of the Mac App Store Guidelines

2 - Development Environment

2.1 - Programming Languages

- JavaScript, maybe jQuery), HTML (HTML5), CSS (CSS3, SCSS) – Web Application, fast prototyping
- C, C++ – iOS, Mac, Windows, PSP, WebGL
- Objective C – iOS, Mac
- Java – Android
- XML (or other platform-independant markup) – communication and persistence

2.2 - Software

- Notepad++ (or other advanced, customisable text-editor)
- Chrome developer tools (and Webkit, Firebug, and other in-browser development tools)
- Visual Studio 2010/2008
- XCode 4
- Tortoise SVN (or GitHub, or best alternative versioning system)
- Photoshop (and other asset-production software)
- WAMP Server
- Frameworks and libraries
  - jQuery/Prototype
  - Modernizr.js
  - Three.js
  - Node.js
  - Sony PSP development plugins for Visual Studio 2008
  - Cocos2D/Box2D

2.3 - Hardware

- Dell Studio 1558, Intel Q740 Core i7, 4GB RAM, Windows 7 Home Premium/Professional
- PSP development kit
- MacBook Pro (hopefully)

SECTION VII – PROJECT MANAGEMENT AND OBJECTIVES

1 - Approach

- The project will be developed using an adaptive, iterative approach
- Working versions will be produced, incrementally adding features
- Adaptations will be made as the need arises and reviews and reflections are carried out at each iteration
- The project may change substantially, in keeping with its general aims and themes

2 - Iterations

2.1 - Gameplay Mechanisms and Basic Physics

- Iteration 0.1 will be a demonstration of the gameplay mechanics
    - Protagonist, with movement, health, and normal combat
    - Fly predator swarm, with full behaviours
    - Full core negative feedback loop in place
    - Initial game balancing
- It will include the minimum neccessary physics and visual rendering required for this purpose
- The central modules will be established, and a modular application architecture will be in place for other components – rendering, user input, etc.
- This is the core of the project

2.2 - Basic Light Simulation

- Iteration 0.2 will begin exploring the basics of light simulation
    - Lighting
    - Shadow casting
    - Non-ray-traced phosphorescence
- Good performance and a scalable architecture must also be achieved
- This is the first phase of the real-time light simulation challenge

2.3 - Advanced Light Simulation

- Iteration 0.3 will achieve the advanced goals of the light simulation
    - Reflection, with bleeding colours
    - Refraction, with dispersion
    - Caustics, with sharp patterns
    - Phosphenes
    - Crepescular rays and volumetric light
    - Rays of varying intensity
    - Reaching a good approximate solution for the rendering equation
- Performance optimisations must be implemented
- This is the final phase of the real-time light simulation challenge

2.4 - Advanced Physics

- Iteration 0.4 will have in place a physics engine comprehensively covering the use-cases required in the game
    - Particles
    - Soft bodies, both “open” and “closed”
    - Rigid bodies
    - Collision detection and resolution engine
    - Approximated air-flow reaction for atmospheric particles
    - Magnetic attraction for collectibles
- This will be achieved in a modular, scalable physics engine
- This, along with flocking and lighting, is the third major technical milestone

2.5 - Further Variety in Gameplay and Antagonists

- At the stage of Iteration 0.5, the other antagonists will all be fully integrated into the game
- The combat systems and artificial intelligence for the protagonist and antagonists will be in place
- Scoring and multiipliers will largely be in place, recording and balanced
- The gameplay and visual elements will have reached a fully working stage representative of the eventual release

2.6 - Dynamic Audio

- Full dynamic audio will arrive in Iteration 0.6
    - All audible events will produce sound and alter it suitably with variables
    - Sound pipeline/channel will be in place and managing all incoming requests
- This finishes out the core aesthetic elements of the game

2.7 - Stage Generation and Progression

- Iteration 0.7 will move the game from a demonstration mode to introduce advancement
    - The stage generation module should generate reliable results for arbitray arguments, and therefore be capable of producig a near-infinite variety of distinct stages
    - Persistence will be introduced at this stage in relation to stage generation and retrieval

2.8 - Menu System

- Iteration 0.8 will adorn the game with the neccessary trimmings for users to manage it, and support local user profiles
    - Persistence will be completed to this local level

2.9 - Scores

- Iteration 0.9 will fully support user profiles
  - Recording and retrieving scores, finishing off persistence
  - Full neccessary communications for the first target platform to manage user profiles via a server

2.10 - Tightening Up and First Platform Launch

- Any remaining major bugs will be removed, tweaks made, and a final rebalancing will be done before Iteration 1.0 is released for testing
- Pending testing and rebalancing, *Lumens* will be launched on its first platform, possibly commercially

2.11 - Revisions, Changes and Subsequent Platform Launches

- Iterations X.X will repeat as many of the above steps as neccessary to secure releases on some or all of the target platforms
- Changes and enhancements are expected to take place during these stages
- The application may also require a commercial strategy, if the results are marketable, which will be explored later

AFTERWORD

About this Document:
This report is largely based on Baldwin's Game Design Document Outline (*Baldwin, 2005*), the most suitable design report template for the project in question.
It is a first, design-focused draft, and will change and develop as this project moves forward.
Features proposed here may only be discussed briefly in this version, and elaborated upon or expanded later as they are explored; conversely, others may be discussed in this document which will not appear in the final implementation of this project as time, suitability, performance, and other factors are considered. It's spirit is very much an idealistic one – many of the described features may be dropped pending practical considerations.
This is a living document, and – at this stage – serves only to outline the ideas and approaches relating to this project, not to definitively lay out the entire course of development.

REFERENCES

Reynolds , Craig, 2001 (online), “*Boids: Background and Update*”, www.red3d.com, accessed 10.10.2011, URL: http://www.red3d.com/cwr/boids/

Buckland, Mat, 2005, “*Programming Game AI by Example*”, “*Chapter 3: How to Create Autonomously Moving Game Agents: Group Behaviours*”

Owen, G. Scott, 1999 (online), “*Ray Tracing*”, www.siggraph.org, accessed 16.10.2011, URL: http://www.siggraph.org/education/materials/HyperGraph/raytrace/rtrace0.htm

Wikipedia, 2011 (online), “*Ray Tracing (graphics)*”, en.wikipedia.org, accessed 16.10.2011, URL: http://en.wikipedia.org/wiki/Ray_tracing_(graphics)

Baldwin, Mark, 2005 (online), Baldwin Consulting, www.baldwinconsulting.org, accessed 15.10.2011, URL: www.baldwinconsulting.org