

Little People Doing Small Tasks For Big People

Raymond Micheau & Nick Whelan

January 21st, 2017

Ver. 1.0

Table of Contents

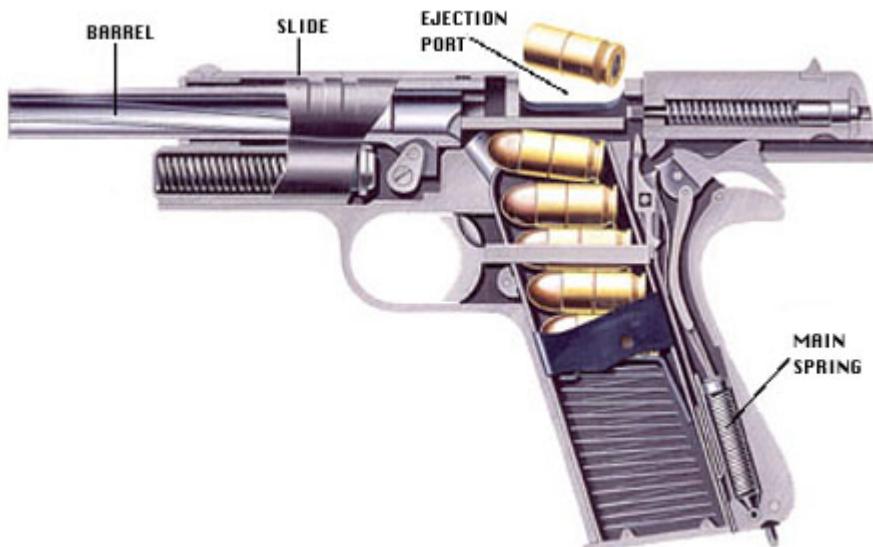
Game Summary.....	7
Game Engine Architecture.....	7
Game specific sub systems.....	7
Weapons (tools).....	7
Moveable Objects.....	8
Game Specific Rendering.....	8
Player Mechanics.....	8
Game Camera	9
Front End.....	9
Heads Up Display.....	9
In-Game Menus	9
In-Game Gui	10
Visual Effects	10
Particle/Decal System.....	10
High-dynamic-range (H.D.R) Lighting.....	10
Post Effects	10
Environmental Mapping.....	11
Scene Graph/ Culling Optimization	11
Spatial Hash	11
Level of Detail (LOD) System	11
Occlusion and Potentially Visible Sets (PVS).....	12
Gameplay Foundations	12
High-level Game Flow System	12
Level Loading	12
Event/Messaging system.....	13
Dynamic Game Object model.....	13
Multiplayer.....	13
Lobbies and Game management.....	13
Game state Replication	13
Object Authority policy	14
Audio.....	14
Digital Signal Processing (DSP) / Effects.....	14
Audio Playback/Management.....	14
Low-Level Renderer.....	15
Profiling & Debugging	16
Collision & Physics	16
Human Interface Devices (HIDs).....	17
Resources	17
Core Systems.....	18

Platform Independence Layer	19
3 rd Party SDKs	20
Game Story	21
Backstory (Known)	21
Cowboy	21
Astronaut	21
Mafioso	21
Mathematician	21
Survivalist.....	21
Ninja	22
Big Business Owner	22
Rock Singer.....	22
Strange Doctor.....	22
Wait Staff	22
Budgie	22
Backstory(Unknown)	23
Cowboy in Toaster	23
Strange Doctor in Toaster.....	23
Budgie in Toaster	23
Astronaut in Balloon Pump	23
Mafioso in Balloon Pump	23
Ninja in Overhead Projector.....	23
Toaster	24
Tape deck	24
Projector	24
Plot.....	24
Game Progress in Relation to Story.....	24
Cut Scenes	25
Introduction	25
Character Select.....	25
Level Select	25
Loading into a Level	25
Exiting a Level	25
Unlocking a Playable Character.....	25
Gameplay.....	27
Number of Players	27
Challenge / Obstacles for Player	27
Game Objective(s)	27
Game Flow	27
Game Mechanics	28

Similar Games	28
WarioWare	28
Lovers In a Dangerous Spacetime	29



.....	30
Regular Human Basketball	30
Rhythm Heaven:	31
Mario Party (Mini Games):.....	31
Balloon Pump:	32
Overview	32
Physics & Movement.....	32
Rules.....	33
Affordances.....	33
In-Game Objects:.....	33
Player(s') Actions	33
Player(s') Interactions.....	33
Gun	34



.....34

Overview	34
Physics & Movement	34
Rules	35
Affordances	35
In-Game Objects:	35
Player(s') Actions	36
Player(s') Interactions	36
Overview	36
Physics & Movement	37
Rules	37
Affordances	37
In-Game Objects	37
Player(s') Actions	37
Player(s') Interactions	37
Phone Switchboard	38
O	38
Overview	38
Physics & Movement	38
Affordances	39
In-Game Objects	39
Player (s') Actions	39
Player (s') Interactions	39
Overhead Projector	40
.....	40
Overview	40
Rules	40

Physics & Movement	41
Affordances	41
In-Game Objects	41
Player(s') Actions	41
Player(s') Interactions	41
Toaster	42
Overview	42
Rules	42
Physics & Movement	42
Affordances	43
In-Game Objects	43
Player (s') Actions	43
Player (s') Interactions	43
User Interface	44
Visual system (visual to player)	44
Control Scheme	46
Completion Timeline	47
Currently Completed Items	47
Objectives	47
9-Week Timeline	50
Appendix	51
Mini-Games	51
Virtual Reality (VR)	51
Balloon Pump	51
Phone Switchboard	51
Gun	51
Toaster	52
Tape deck	52

Game Summary

“Have you ever wondered how the toaster cooks the toast?”

Imagine if one day, every appliance you’ve ever used suddenly stopped working, every last one and for some reason they started working again for no reason. Our game is about that scenario, where the insides of appliances have been taken over by small people who make it work. Imagine a toaster where you put your toast in and it pops up toasted! Have you ever imagined how that actually works? What makes it toast our toast? Well I have news for you! It’s small people, running around the inside of the toaster wielding a flamethrower who, cook the toast to just the right amount before it pops up for us to eat! Or how about a simple Roomba? What if I told you there were two little men running around the inside of it, controlling its rotation and how it moves forward, sucking up all the dirt it can.

Game Engine Architecture

Game specific sub systems

Weapons (tools)

- A. This part of the engine handles all of the in game weapons and tools. If game requires a tool or weapon to be used by the player, this class will handle all weapons as a generic weapon.
- B. This interacts with Human Interface Device system as well as the Event/messaging system to determine when to fire/use the weapon/tool. This class will also contain a reference to an asset for the weapon model and projectile.
- C. This will allow the user to click a button and interact with the world with predictable results. If the user is given a “gun” it is assumed that the user must use a button in order to activate the “gun”. Once the user presses that button the user should then be aware of what the “gun” does. Every time the user presses the button it will perform the same as it did before.
- D. This is necessary for the user to interact with the world there must have a visual way of indicating the interaction.
- E. This part is in progress.
- F. Three to Four days to build.

Moveable Objects

- A. Objects in the game world that user can pick up, push, place, etc.
- B. This will interact with the physics part of the game engine to allow these assets to have correct movements, and a resource manager to give the object a model viewable by the player.
- C. This will allow the user to pick up and move objects to make the world more interactive, adding a new level of immersion.
- D. This is necessary because adding a degree of interaction that really draws the player into the world they are in.
- E. This is in progress.
- F. This will be created in parallel with the physics aspect in order to allow proper debugging. It will take about two weeks to complete

Game Specific Rendering

- A. These are shaders on top of the internal shaders to add a new look to each game. Like adding a cartoon look to one game and then a realistic look to the next.
- B. This communicates with the internal low-level render of the game engine.
- C. To add a new look to each game even though they are on the same engine.
- D. So that each game can have a new look to draw in a user to the world.
- E. This part of the engine is not yet created.
- F. This will take a few weeks to complete. Understanding of the full function of shaders is still in progress.

Player Mechanics

- A. These are parts of the game that give the user direct feedback on what is going on in the game.
 - a. State machine:
 - i. This is the state that the player is in. ex: idle, Running, Walking, Jumping, etc.
 - b. Animations
 - i. Using the state machine it can then pick the correct animation to apply to that state.
 - c. Camera Relative Controls
 - i. This allows the user to accurately move the character around the screen.
- B. This communicates with the input system, the resource system, and the cameras.
- C. This allows the builder to edit the states that the user can enter and the list of animations and controls in-order to tweak the game to feel the way they want it to.
- D. Yes.

- E. This has not been created.
- F. This will take about 3 days to complete.

Game Camera

- A. This captures each frame to be rendered to the screen.
 - a. Fixed Cameras: This is a camera that doesn't move. Giving the user a common view.
 - b. Debug Free Cam: This is used by developers to view the environment without being restricted by the game's physics.
- B. This communicates with all of the render aspects of the game engine.
- C. This determines where the render point is.
- D. Without this nothing will render to the view port correctly
- E. yes.
- F. This is started.
- G. This will take a few days to fully refine.

Front End

Heads Up Display

- A. This part of the engine displays information to the player, also referred to as a HUD.
- B. This connects to the rendering engine, which is utilized to display to the player
- C. This part of the engine displays the timer, tasks to complete, and where each player is in relation to the other members of their team.
- D. This is necessary to display and give out information to the player(s) so that they may make informed decisions based on time, and what their teammate(s) are doing currently.
- E. In-Progress
- F. This part will be made in tandem with the rendering engine, and is currently In Progress

In-Game Menus

- A. This part of the engine is a navigation system to allow the player to move back and forth between different parts of the game. Be it Play Game, to Settings, or to Exit.
- B. Connects to rendering engine, and the HUD to allow the player to select the various menu options.
- C. This part of the engine allows players to select menu options to travel to separate menus which will allow players to change options, select characters, levels, and play the game in sequence.
- D. This is necessary because if there are no menus, settings will be unable to be changed. Resulting in a subpar playing experience for the player overall.
- E. This part will be made in tandem with the rendering engine, and is currently In Progress

In-Game Gui

- A. This is an overlay for the view that is rendered with the camera. Displaying things like health, time, score, ammo, etc.
- B. Connects to rendering engine, and the HUD to display 2D sprites to update the player on the current important information.
- C. This allows the player to see what new and important information is relevant to the current task with a quick glance.
- D. This is necessary because the user should be given all of the information rather than having to go looking for it.
- E. This has not yet been started.
- F. This part will be made in tandem with the rendering engine, and is currently In Progress

Visual Effects

Particle/Decal System

- A. This allows for things like smoke, sparks, mist and bullet holes to exist in games.
- B. This connects with the physics aspect of the engine.
- C. When a bullet hits the wall, rather than leaving an untouched wall. You can make sparks and bullet holes appear on the wall.
- D. This is necessary because it adds more of a dynamic world for the user to interact with.
- E. This part has not been created.
- F. This will be build along with the physics side of the engine.

High-dynamic-range (H.D.R) Lighting

- A. This is how light behaves when interacting with the camera. So things like bloom and flares look more realistic.
- B. This connects with materials and shaders in the lower level render.
- C. This adds more flexibility when it comes to building a level with lights in it.
- D. This is necessary because it adds a more lifelike feel to the game's environment.
- E. This has not been started. More research needs to be done on the topic.
- F. Once lighting and shaders have been figured out it should take about a week or two.

Post Effects

- A. This adds the visual component to flairs and blooms. For example if you look at a light you see "god rays" coming out from the sides of the light source. This adds those beams to the light source.

- B. This is connected with the lower level shaders and resources to load flair billboards. (Flat 2D sprites)
- C. If the player was to look at a flashlight it will feel more like the player is looking into a light rather than a white dot.
- D. This is important because it adds an extra layer of realism to the in game lighting.
- E. This part has not been started.
- F. Will be built along with the HDR lighting.

Environmental Mapping

- A. This adds reflectiveness to a surface. Thus allowing metallic objects to reflect their surroundings.
- B. This connects with the Camera and shader.
- C. This allows for reflective objects to actually reflect their surroundings
- D. This is necessary to add a degree of realism to the game's world.
- E. Has not been started.
- F. More research needs to be done on this topic.

Scene Graph/ Culling Optimization

Spatial Hash

- A. A collision system that can be increased in size overtime dynamically as the world size increases through Binary Search Patterns
- B. This part connects to the collision system and the Unique Identifiers.
- C. Will allow for slightly faster local collision determination.
- D. Being able to determine, and resolve collisions quickly, and efficiently will help to make the game run smoother.
- E. Not Yet Started
- F. Will be made in tandem to the collision engine

Level of Detail (LOD) System

- A. Level of Detail system that will dynamically change and update the Level of Detail (LOD) in real time to lighten the load of the renderer and help improve game stability.
- B. This part connects to the rendering engine directly.
- C. This will allow us to tone detail down on objects that are further away from the camera and improve the overall runtime of the game.
- D. I think it is necessary to improve the overall feeling of the game. Being able to drop something away and watch it fade off into the distance before disappearing improves the overall game feel.
- E. Not Started.
- F. Will be made in tandem to the rendering engine

Occlusion and Potentially Visible Sets (PVS)

- A. This part of the engine covers the following:
 - a. Occlusion Culling
 - i. Disables rendering of objects if not currently visible by the camera because they are obscured or off screen.
 - b. Ambient Occlusion / Screen Space Ambient Occlusion (SSAO)
 - i. A rendering technique applied upon each point exposed to ambient lighting. Ex. Going down a tube the light becomes more occluded and darker the deeper you travel.
 - ii. SSAO is a technique used to approximate the ambient occlusion effect in real time.
 - c. Potentially Visible Sets
 - i. This is a collection of objects from a location. This is a pre-compiled method to determine visibility.
 - ii. Areas may be cut into sections to be used to determine visibility, such as a corner hallway.
- B. This part directly connects to the rendering engine to be able to do all the different parts of occlusion, be it culling or SSAO.
- C. Allows for quick culling and removal of objects that are determined to be off-camera at any given point in time
- D. Faster running of the game engine during gameplay.
- E. This part has not yet been created.
- F. Will be created in tandem with the rendering engine.

Gameplay Foundations

High-level Game Flow System

- A. This is the order in which game events happen. Example: move, collide, then draw.
- B. This connects all of the parts together.
- C. This states the order that things happen in code.
- D. This is important because it defines the order in which the game updates.
- E. This has not been created.
- F. This will take a day or two to complete.

Level Loading

- A. This is a way to set up the next part of the game for the user.
- B. This connects with the resource part of the engine.
- C. This loads levels so that it is ready for the user to enter without seeing “pop” ins.
- D. This is necessary because it loads the game.
- E. This has not been created for OpenGL. Reworking older SDL code.

- F. Three days of solid work to complete.

Event/Messaging system

- A. This sends a global message to the event handlers when an event needs to be handled.
- B. This connects with all the input and physics classes.
- C. This is used when a user presses a button. It posts a message of the button press that is then handled by an input handler.
- D. This is necessary because it allows communication between classes and objects.
- E. This part has not been created yet.
- F. This will take about 2 days to build a skeleton.

Dynamic Game Object model

- A. This is a base object that has components added to it. Similar to the way game objects work in Unity.
- B. This uses the physics engine along with shaders and resources for models.
- C. This is what makes up every game object.
- D. Yes
- E. Hasn't been created yet.
- F. Will be built along with the physics in about a week.

Multiplayer

Lobbies and Game management

- A. This part of the engine will allow players to connect to one another via the Internet and manage their games from within as a host.
- B. This part connects to all others in one way or another, as playing in a multiplayer environment will require information to be sent and parsed across the Internet.
- C. This allows players to connect and play with each other across the globe.
- D. This is a necessary part of our engine because it will allow the game to be more widely played by a larger audience than being limited to just couch co-op and playing locally.
- E. This part of the engine has not yet been created.
- F. This will take about 2 weeks to complete

Game state Replication

- A. The process of replicating the state of a game across a network and displaying it to other players via a network connection.
- B. This part connects to the part above, as it will work with lobbies and management to allow players to view the same thing across all versions of the game currently being run.

- C. Allows multiplayer connectivity and the ability to play in the same environment as all other players currently in the same lobby playing.
- D. I think this is necessary to have parity across all people currently playing the game
- E. Not Started
- F. This will be built in tandem with the rest of the multiplayer component of this game. Research must be conducted to complete.

Object Authority policy

- A. Host and Client communication via an Internet connection.
- B. Connects to the other parts of Multiplayer Networking to allow proper communication between host and client
- C. This allows players to interact with each other through an Internet connection
- D. This is necessary to facilitate communication and interaction between the players currently playing the game.
- E. Not Started
- F. One Month to complete

Audio

Digital Signal Processing (DSP) / Effects

- A. Digital Signal Processing is used to take real world signals like voices, sounds, etc. and manipulate them using math to create different effects. Ex. Reverb.
- B. This will be its own part of the engine and come into contact with the Human Interface Devices (HIDs) to play audio, and maybe networking to do voice communication via the Internet.
- C. This allows players to communicate across the Internet and be able to properly formulate strategies.
- D. This is necessary to allow communication and facilitate the construction of strategies to utilize against the other players or to just speak to one another through In-Game means
- E. Not Started
- F. This will take two weeks to complete

Audio Playback/Management

- A. This allows for the playback and management of audio files stored within the game to be played back to the player via speakers.
- B. This will connect to the HID via speakers.
- C. This plays background music, and sound effects relevant to our game
- D. Having music helps to create a feeling or emotional reaction in a player, and may help them enjoy the gaming experience.

- E. Not Started
- F. This will take One Week to complete, In Tandem with DSP

Low-Level Renderer

- A. This part of the engine handles rendering at a low level. Dealing with all of the following
 - a. Materials
 - i. Materials that are to be applied to 3D Models, or to the environment and are used to determine what they look like in the scene.
 - b. Cameras
 - i. Cameras that allow the player to view the scene and follow a character around on screen or show different perspectives of the action.
 - c. Lighting
 - i. Lighting the scene with light sources such as point lights, or spotlights. Can be static with everything pre-set, or dynamic when players move objects or interact with the scene.
 - d. Shading
 - i. The shadows that are cast by the lighting onto the scene. Can be both static and dynamic, works in tandem with lighting.
 - e. Viewports
 - i. Windows created to show and display the game when running
 - f. Text
 - i. Display text to the scene, and write with it when necessary via dialog boxes or on screen messages
 - g. Graphics Device Interface
 - i. Allows for graphics and text to be output to monitors via viewports
- B. This connects to the other parts such as the physics and collision as a visual representation of how they are working behind the scenes. And it will do the same as it connects to the resources. This will allow visualizations of our code to be displayed.
- C. This part of the engine does the entire rendering to the screen. Displaying all of the graphics and showing all the interactions the player has to do through text or graphics.
- D. If this is not present, how will information and gameplay be displayed to the player?
- E. This part of the engine has been started and is in progress.
- F. Two weeks is the current estimate to completion

Profiling & Debugging

- A. This handles showing and displaying different debug information visually or through text. Visually would be an FPS display that shows the highs and lows of current FPS through a chart or through a counter in one of the corners or through menus that allows the spawning or viewing of different information.
- B. This connects to other parts such as the physics or renderer. To be able to display different physics information and what the colliders are when looking at a model on screen.
- C. This part of the engine displays different debug information relevant to the physics and FPS that is present.
- D. If this is missing it becomes harder to determine debug information and gather information to help figure things out when building scenes and determine if and how something has gone wrong in the back-end to fix.
- E. This is in progress. Logging is in, planning to add a FPS counter and memory usage information.
- F. Two weeks after graphics have been completed to be able to visually display graphical information on-screen through text or charts.

Collision & Physics

- A. This part of the engine handles:
 - a. Physics
 - i. Gravity
 - ii. Forces
 - b. Collision
 - iii. Collision Meshes
 - iv. Collision Maths
 - b. Vector & Matrix Maths
 - i. Addition
 - ii. Subtraction
 - iii. Multiplication
- B. This part connects to the debug information to provide visual information on the physics taking place via text or visual aids (TBD) as well as to the visual portion through the renderer which will show the effect of physics upon the scene and actors inside of it.
- C. This part of the engine will do our physics for player movement and how gravity effects each player and how they collide with the world and with each other character(s)/object(s) inside of the scene.
- D. This is necessary for the engine because this will be essential for moving certain pieces of objects into place. Like for example, pulling on a dragon's tail or sliding bullets into a magazine. Being able to determine the physics of the shape and movement via friction will be an essential piece of our game.
- E. This has not been started yet.

F. This may take up to two weeks to have working, and a month to have fully, completely operational.

Human Interface Devices (HIDs)

- A. This part of the engine handles input/output through keyboards, mice, and other physical pieces of hardware like controllers or joysticks.
- B. This will connect to the three pieces of hardware mentioned above to allow physics to enact upon players, show and display what keys may be pressed via debug, and the visual representation of the movement/physics.
- C. This allows player(s) to control their character and control menu systems.
- D. If this is not a part of it, the player will not be able to control the game in any way.
- E. This part of the engine is not yet created.
- F. Approximately three days to complete.

Resources

- A. This piece provides access to a wide number of important, and necessary game assets such as:
 - a. 3D Models
 - i. Models imported from a 3D modeling software such as 3DS Max or Maya
 - b. Textures
 - i. A 2D image from a piece of image editing software to be applied to a model
 - c. Materials
 - i. Contains the details of what something should look like: bump maps, transparency, and textures, etc.
 - d. Fonts
 - i. Fonts that can be utilized to write text on screen
 - e. Skeletons for Animation
 - i. Skeletons provided via 3DS Max or Maya that are attached to a model and rigged to be moveable
 - f. Parameters for Physics
 - i. Physics parameters to determine how an object moves in an environment
 - g. Collisions
 - i. Collision math for when two objects intersect and determine how they react and move apart or what effects that has upon the world itself
- B. This will directly feed into the renderer to render the appropriate resources and the physics for the parameters used to determine the physics.

- C. This will allow the resources to be loaded and stored properly and be accessible in the correct places via manager that will store lists of all currently loaded and usable resources.
- D. If this isn't apart of the engine, there will be no real way to handle all the resources and where they're supposed to go and how they are to be used. So it is vital to be able to store and manage these resources efficiently and effectively.
- E. This part is in progress currently.
- F. This will take approximately two weeks to complete

Core Systems

- A. This part of the engine will encompass all Core and NECESSARY systems required for the engine to run and be usable, this includes:
 - a. Assertions
 - i. Used to abort the program if a required part of it does not work or is NULL
 - b. Math Library
 - i. Required to make certain maths more efficient
 - c. Debug Printing / Logging
 - i. Used to print and log debug information to file or to console
 - d. Object Handles / Unique IDs (UUIDs)
 - i. Used to store objects with unique identifiers allowing quick access to them later
 - e. Engine Configuration
 - i. Store important customization information, such as screen dimensions or start-up properties
 - f. Run Time Type Information (RTTI) Reflection & Serialization
 - i. Store run time type information, such as renaming or redefining types
 - ii. Being able to modify itself at runtime and change things
 - iii. Serialization to store and reference data in a file, or memory and save space
 - g. Module Start-Up / Shut-Down
 - i. To properly start up the program and shut it down properly and allow proper memory management during these phases
- B. These pieces will directly feed into most other places of the engine. Being able to call and reference objects for the renderer to display. Printing out debugging information via a logger with information from our debugging processes. Allowing us to use the math library to use within the physics and do matrix math, vector math, etc. Use assertions to be sure nothing is NULL at start-up. Being able to properly start and turn off the program as well, and allowing stored, and serialized start up information in an ini file for use at each boot.
- C. Answered above.
- D. If these aren't included, it will become significantly harder to properly use the engine. If something is null at start it will not be realized until it is too late and it will fail later; or the physics becoming more cumbersome and laborious without the math library to

ease that; having to manually change our display information at each program boot. These will allow us to optimize our engine and allow faster usage of the different components and resources available.

- E. Some of this part is currently In Progress / Completed
- F. These Core Systems will be completed over the course of the year, to be completed by the end of 2nd Semester.

Platform Independence Layer

- A. The Platform Independence Layer allows for platform independence and will allow cross platform compatibility such as MacOSX, Windows, and Linux. This encompasses a number of things as well:
 - a. Atomic Data Types
 - i. These are booleans, floats, integers, and characters
 - b. Collections and Iterators
 - i. Collections are a group of items, stored in some sort of data structure like a hashtable with an iterator to loop through the list to find, retrieve, or store items in the structure
 - c. Hi-Resolution Timer
 - i. A timer that gets the tick directly from the CPU and is incredibly accurate when used to determine ticks and a tickrate.
 - d. Threading Library
 - i. Library used for multithreading, which is a sequence of instructions to be done concurrently with others
 - e. Graphics Wrappers
 - i. Virtual functions to be used to call functions inside of the graphics class of functions
 - f. Physics / Collision Wrapper
 - i. Virtual functions to be used to call functions inside of the physics / collision class of functions
- B. With the Atomic Data Types being used everywhere and all over the engine. The hires timer is being used to determine game events and the current FPS being displayed to the screen, as well as the different wrappers being used to call functions that are needed. This will be used all over the engine in a massive capacity.
- C. This part of the engine allows players to be playing on different computers, and not be dependent on a singular OS platform to play.
- D. This is necessary to allow more players to have access to the game and be able to play it effectively.
- E. This part of the engine is currently in progress.
- F. Approximately Two Weeks for completion

3rd Party SDKs

- A. This piece of the engine utilizes what it says, SDKs such as:
 - a. OpenGL 4.5
 - i. A graphics library that allows us to utilize 3D and 2D graphics to create our engine
 - b. PhysX
 - i. NVidia PhysX is a multi-platform game physics engine that allows use on a wide variety of devices.
 - c. STL / STLPort
 - i. A set of libraries allowing us to move the C++ Standard Template Library (STL) to multiple platforms for use.
 - d. Boost++
 - i. A set of libraries to work with STL and allow more efficient code production
 - e. Etc.
 - i. Any other SDKs that we may deem we need later down the line.
- B. This part connects to a few places. OpenGL connects to our renderer; PhysX connects to our physics engine. STL, and Boost connect to the PIL and help towards multi-platform independence.
- C. This part of the engine allows us to render items to the screen, and utilize a well fleshed out physX SDK to improve upon our physics in a fun, exciting way.
- D. This is a necessary part of our engine to help render, and properly apply physics maths to objects and scenes in a believable way. As well as be multi-platform with STLPort and Boost to allow the continued use of the C++ standard template library on Mac OSX and Linux.
- E. This part of the engine is partially in progress while not started in others.
- F. Will be built along the engine over the course of the year

Game Story

Backstory (Known)

Players will be given the backstory to the characters they choose to play as. With a wide variety of characters to select from, players can expect a wide amount of character variety. Some examples are as follows:

Cowboy

The cowboy runs a lone ranch out in the middle of the carpet wasteland. They needed to take on extra jobs to keep the ranch running and stable. Their always seen wearing their trademark cowboy hat upon his head and an ascot around his neck. Loves his horses, heat, and 12-Gallon Hats.

Astronaut

This character came from outer space, looking to expand their horizons and learn about the larger world that they've only seen from above. Their body has become accustomed to the environment and they require a helmet to survive inside of Earth's atmosphere.

Mafioso

A member of the Appliance Alliance, a conglomerate of several mafia groups working to take total control of appliances across the globe. Works as an inside man, looking to slowly work their way to becoming a mafia don.

Mathematician

Here to figure out the answer to the ultimate question. Is it possible to define mathematical constants to everything imaginable? Can we use equations to determine how quickly we can cook toast? There's only one way to find out, and they're here to figure it out! Even if everyone tends to tune them out after a few moments.

Survivalist

A wild individual, lives alone in a forest, loves Survivorman and Survivor. Owns several books detailing how to live 'off the grid'. Loves the wilderness, and making fires in the middle of a field. May also be a slight pyromaniac.

Ninja

Comes from Edo, from another time and place where being sneaky was important to surviving, and super cool. Lives in the shadows, creeping from place to place and hiding in plain sight. The Ninja specializes in looking cool and- Wait, where'd they go?! Well, I'm sure they'll pop up in time to join us. Has a really strange affinity for boats with guns.

Big Business Owner

Changes everything at a whim to try and make others happy. "I received a message! Tuesdays will now be Taco Tuesday!" Is one of their most famous quotes. Lives to make the best business decisions they can and get it the top of wherever they're working.

Rock Singer

Famous singer, specializes in harmonious rock operas with the rest of their band. Started playing piano at a very early age. Has an affliction with being the designer of their band logo and having an incredible accent that draws in everybody.

Strange Doctor

Dubious doctor with a strange Ph.D in cloning sciences, seems like a nice character, says they're from Transylvania. Breaks out into song on occasion and might not actually be a doctor thinking about it now.

Wait Staff

Zooms around a restaurant, taking orders, and wiping down tables as other Meeples leave after eating and paying for their order. Has gotten fed up with this job and joins the Appliance Alliance in search of a more fulfilling and interesting life.

Budgie

Tweet, tweet tweet. Tweet. Tweeeeet. SCREEEEE. Tweet tweet tweet. Tweet? Tweet tweet. Tweet, tweet. Tweet. Tweet tweet, tweet tweet. SCREE SCREEEEE.
(Does anyone know how to speak Budgie? 'Cuz I certainly don't.)

Backstory(Unknown)

What would be unknown to the player, is the reasons why the characters have been put into their current position. Why is the Ninja pumping a balloon? Why does the Astronaut care about cooking toast? These wouldn't be known to the player, and may need to be inferred, or come later as an extra tidbit of information. Examples follow.

Cowboy in Toaster

The cowboy has come to the toaster in search of one of their lost ranch hands who disappeared and after getting a lead, arrived here. Has been tasked to cook some toast to get the next tip to where their lost hand has gone too.

Strange Doctor in Toaster

The strange doctor is searching for new toys to use in their experiments. The flamethrower being used inside of the toaster is the perfect tool to look into getting for use on future experiments.

Budgie in Toaster

The Budgie is here to eat the crumbs that end up falling off of the toast after it's been fully cooked. I think it's only in it for the food?

Astronaut in Balloon Pump

To breathe, they must pump up the balloon and pop it. Scaring them enough to get a deep enough gasp of oxygen to continue surviving.

Mafioso in Balloon Pump

Is looking to improve their confetti operations by expanding them into balloon pumps. Using this as a test to show off how much better balloons would be if they all had confetti inside of them.

Ninja in Overhead Projector

To perfect their sneaking technique, they've put themselves in an environment where it's LITERALLY IMPOSSIBLE to hide in shadow. They're trying to force themselves to be better and hiding in plain sight. No matter where they are.

Players also would be unaware as to why appliances have stopped working. Allowing them to discover through play about what exactly has happened to the internals of the appliances that have caused them to be this way through their environments and through short cut scenes examples of which can be read below.

Toaster

In the toaster the players may be able to see broken and shattered heating elements scattered across the bottom of the play area. With elements of foul play seen strewn throughout.

Tape deck

Instances of frayed cords, broken transistors and circuit boards lay across the arena. With traces of the slot itself scattered around and physically moveable.

Projector

Pieces of broken glass scattered around, metal parts, filaments scattered about. Singed pieces of metal that make up the frame of the projector.

Plot

It begins with appliances around the world having stopped working. This brings a call to arms by the Appliance Alliance to all willing Meeples on the planet (and off) to get to work! Regressing to an age where appliances were run by them to keep the Big People satiated. As the story progresses as players beat and complete each appliance to get it running, they discover more about a group of characters named, "The Miserly Meeples Mutuality" whom are out to ruin the world and get the Big People turned against the Meeples. Players progress through the game, completing mini-games and foiling their plots until they concede in their plans and disappear back into obscurity.

Game Progress in Relation to Story

Game play relates directly to the story progression. As players progress through each level they can see more as to what's happened in the game prior to them arriving at the appliances. With environmental details, such as burned / broken circuit boards, scattered within the scene. Showing the destruction left behind by unseen, and unknown characters. With some fleeting

glimpses of shadowy figures running around behind the scenes as soon as players get to the appliance or as soon as they finish. An unknown character may be seen running away after the level is completed and the camera is moving away from the play space.

Cut Scenes

Cut scenes in Little People Doing Small Tasks for Big People would be minimal. With an introductory cut scene when the game is started.

Introduction

Game starts with a team logo, “Dinosaur Rave Party Games” fade in. Fade to all the other logos of libraries we’ve used. Fade to the actual game logo, which will turn into the main menu.

Character Select

Shows the importance of how characters use their hidden homes to travel from their home to the next menu, the Breaker.

Level Select

After the character select screen the camera will fly along a cable to the Level Select screen. Where players will select breakers to select the mini-games to be played.

Loading into a Level

Camera moves from along the cable to the level and zooms into an overview of the play area. However, the camera is meant to look for each individual level.

Exiting a Level

Characters run towards the entrance they came in from. The camera zooms along a cord, and into the next level.

Unlocking a Playable Character

Cut scene of a character busting their way out of a box, destroying it as they become free to use and striking a victory pose being beautiful.

Gameplay

Number of Players

- A potentially infinite number of players
 - o Dependent on hardware available
- However, it will generally be between 2 and 4

Challenge / Obstacles for Player

Obstacles and the challenge for the player will be the variety of mini-games that are available to choose from and the challenges each of those games brings out while in game.

Game Objective(s)

The game objective is for players to figure out the best way to overcome the given mini game challenge and beat the other team in points across a set of mini games selected at the start of the game.

Game Flow

The players start at a character/team select screen. They select the team by plugging in power cables to the outlets. Once a team is selected the player can then switch characters. Once the teams are selected and everyone is ready the level select screen will appear. This screen is setup like a breaker switch with each switch representing a mini game. Each player selects a mini game. Then once each player is locked in it loads the first mini game. The Mini game is played. Once that game is finished points for the game are rewarded. This continues until the end of the last mini game in the list. Once finished the winning team is displayed and it goes back to the player select screen.

Game Mechanics

Similar Games

WarioWare





Video: <https://www.youtube.com/watch?v=b64K5asTJCc>

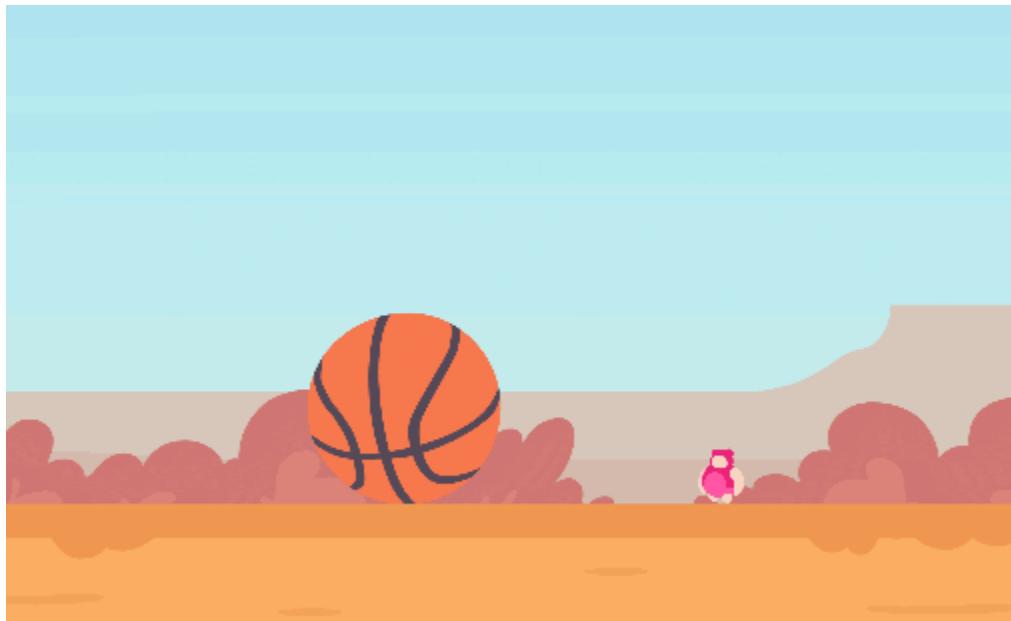
Lovers In a Dangerous Spacetime





Video: <https://www.youtube.com/watch?v=5L20xxqDfII>

Regular Human Basketball



Video: <https://www.youtube.com/watch?v=QDaJqJmvi0M>



Rhythm Heaven:

Video: <https://www.youtube.com/watch?v=3ZRjd3ic4pU>

Mario Party (Mini Games):

Overview of (Quilt for speed): http://www.mariowiki.com/Quilt_for_Speed

Balloon Pump:



Pump Order of Operations:

- Balloons are loaded onto the front of the pump.
- The balloons have to be stretched over the front of the pump
- Slide is pulled all the way back to cycle air into the barrel
- (For sake of fun) When the slide is let go it goes to spring to the front.
- Air is added to the balloon.
- The balloon has to be held in order to stop the balloon from flying off the end.
- Repeat till Balloon is full

Overview

Players are tasked with pumping air into a balloon until it pops (might make it so the player needs to pop more than one balloon). The first team to do this wins the game.

Pump will be on the table, with balloons scattered around it. Players are tasked with loading a balloon onto the pump. Both of the players have to work together to get the balloon on to the top by stretching the end of it over the nozzle, players working apart in order to stretch the balloon. Once the balloon is on the pump the players must slide the pump back to fill the balloon with air. It will be slower if one person pumps at a time rather than both. Once the pump is pulled back one player has to run to the front and grip the balloon to stop it from flying off.

Gameplay most similar would be Mario party balloon popping mini game.

Physics & Movement

- Will be 2D movement on a tabletop surface.
- Physics will be in 3D. To be able to move objects off the surface of the table and see them drop
- Or to see objects rotate in multiple directions in real time.

Rules

Visible:

- Timer
- Objects that need to be moved
- Player Location Indicators
- Screen Split down middle showing both team's progression
- Slide moves only slightly to indicate the requirement of multiple people

Invisible:

- Order of Operations for task
- Variable Object Movement Speed (Moving something solo vs. Working as a team)
- Players may have to work as a team to move different parts of the pump, or only a single person may do a single task. (Pulling/pinching the balloons requires one, slide and stretching requires the entire team)
- Mass of different objects to manoeuvre them around the environment

Affordances

- Only one balloon can be placed at a time.
- The balloon will not fill unless it is stretched over the top
- Two players must pull back the slide in order to fill it faster.
- If one player pulls the slide it will move a lot slower.
- The amount of air added to the balloon will be dependent on how far back the pump is pulled.
- If balloon is placed incorrectly and a pump is tried then the balloon will shoot across the room
- If the balloon is not pinched it will also fly across the room.

In-Game Objects:

- The Pump(s)
- Balloons to be loaded

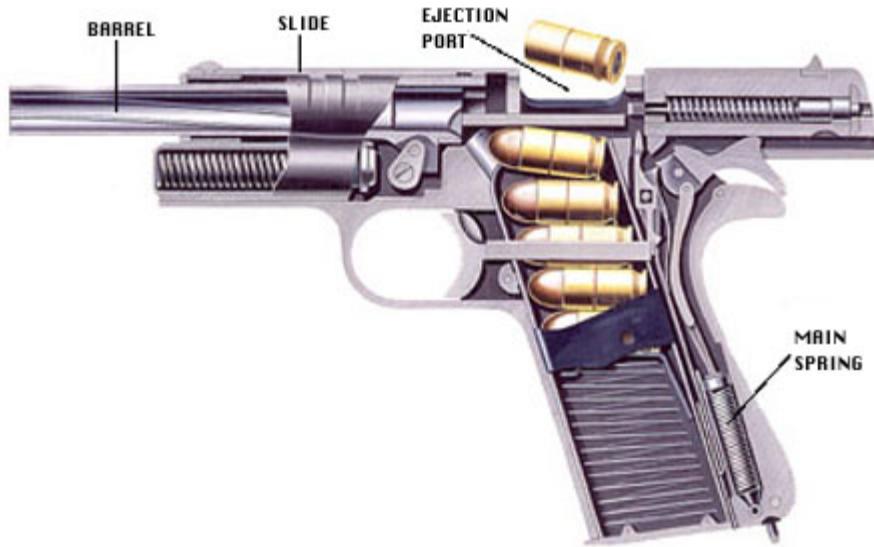
Player(s') Actions

- Movement
- Pressing Buttons to Interact with the World

Player(s') Interactions

- Dragging objects
- Pushing objects
- Grabbing players (to pull harder on objects)

Gun



Gun Order of Operations:

- Bullets are loaded into an empty magazine
- Magazine (Mag) is slid into the receiver of the weapon
- Hammer is drawn back
- Slide is pulled all the way back to cycle the round into the barrel
- Trigger is pulled
- Slide moves back, spent casing is ejected
- Slide moves forward, next bullet is drawn into the barrel

Overview

Players are tasked with loading and firing a weapon (Nerf gun, pistol, BB gun, paintball gun) against the other team. The first team to do this wins the game. Magazine will be on the table, with ammunition scattered around it. Players are tasked with loading each piece of ammunition as a team via a top down perspective. After the bullets are loaded into the magazine, players must co-operatively slide the magazine into the weapon. Afterwards the weapon must be rotated to face the opposing team and then the trigger must be pulled.

Physics & Movement

- Will be 2D movement on a tabletop surface.
- Physics will be in 3D. To be able to move objects off the surface of the table and see them drop

- Or to see objects rotate in multiple directions in real time.

Rules

Visible:

- Timer
- Objects that need to be moved
- Player Location Indicators
- Screen Split down middle showing both team's progression
- Slide moves only slightly to indicate the requirement of multiple people

Invisible:

- Order of Operations for loading / firing weapon
- Variable Object Movement Speed (Moving something solo vs. Working as a team)
- Rotation to a random position after firing and must be reset if missed
- Players may have to work as a team to move different parts of the weapon, or only a single person may do a single task. (Hammer / Trigger requires one, slide requires the entire team)
- Mass of different objects to manoeuvre them around the environment / into the gun

Affordances

- Only one bullet can be placed at a time. If player A is loading a bullet and B is behind A doing the same thing then player A is trapped between the two bullets
- The gun will not fire unless the mag is loaded
- A bullet will not leave unless all bullets are loading in the mag
- The hammer at the back of the gun must be pulled in order to pull slide.
- The slide of the gun must be pulled. Both players must pull back this part because it is too heavy for one.
- Once the gun is fired then the gun spins.
- If the bullet is fired and misses the other team. Then they have to run across the table and grab the shell of the bullet. Unload the mag. Remove the top bullet and place the shell in that place. Then the other bullet must be placed on top in order to fire a working bullet
- If a player is trapped in the barrel and the gun is fired then the player is shot across the table. Doesn't count as a hit.

In-Game Objects:

- Weapon being loaded
- Rounds to be loaded
- Magazine to be inserted into weapon

Player(s') Actions

- Movement
- Pressing Buttons to Interact with the World

Player(s') Interactions

- Dragging Objects
- Pushing Objects
- Grabbing Players (to pull harder on objects)

Tape deck



Overview

Players must play one of several music boxes based upon what is currently being put into the front of the speaker or tape deck. Players will run between these objects spinning the handle and playing the music from the speaker for a predetermined amount of time until the 'Eject' button is pressed.

At which point players must run to the deck, and push on it as a team to remove it from the player and back to the user. The speed of the song will be determined at the rate of speed players rotate the handle of the music box. Faster rotating will speed up the song beats per minute (BPM), and vice versa for rotating it slowly. Players must maintain a delicate balance of rotation when playing the song.

Gameplay most similar is rhythm heaven and little big planet.

Physics & Movement

- Will be 2D movement on the inside of a tape deck
- Climbing up and down ladders and manoeuvring around the deck
- Physics will be in 3D.
- Or to see objects rotate in multiple directions in real time.

Rules

Visible:

- The music to be played is visible by the tape being inserted
- The several different music boxes, relating to different tapes
- The eject button being pressed for players to remove the deck

Invisible:

- Timing between inserting the deck and when players can start to move
- Timing between the eject button being pressed and the object being activated to be moved

Affordances

- Visible BPM meter showing the Beats Per Minute of the song currently being played
- Visible song descriptors on the music boxes to allow players to visibly see what song should be played.
- The desired rate of BPM the song should be played at and an updating BPM number to show what the current BPM is.

In-Game Objects

- Tapes
- Music boxes
- Tape deck

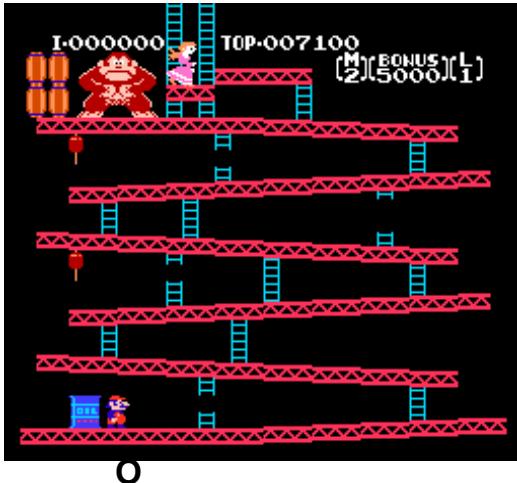
Player(s') Actions

- Movement
- Button pressing to interact

Player(s') Interactions

- Rotating handle at varying speeds
- Pushing objects around

Phone Switchboard



Overview

Players receive requests that require each time to move ends of the cords to specified slots or 'receivers' which could be anything inside of the area. Boxes, makeup kits, anything. Players must connect more 'calls' before time runs out. Unlike a regular phone switcher. Players must co-operate by dragging either end of the cord around to the right spots; or by dragging objects to the correct end of the switcher cord.

Rules

Visible:

- Objects to connect between each other
- Countdown ring to connect objects
- Connection Counter (score)
- Length of Cord available for usage to connect objects

Invisible:

- Timing between instructions being received
- Actual time to connect objects
- How many objects will be used in the game

Physics & Movement

- Will be 2D movement on a tabletop surface.
- Physics will be in 3D. To be able to move objects off the surface of the table and see them drop
- Or to see objects rotate in multiple directions in real time.

Affordances

- Game Timer
- Bubble at the top of each player to show what is to be connected.
- Ring to denote the time that players have to connect before moving to the next set of objects.

In-Game Objects

- Crate
- Tape deck
- Speaker
- Deodorant
- Pack of Gum
- Water Bottle
- Cell Phone
- MP3 Player
- USB Flash Drive
- Dental Floss
- Batteries
- Werewolf Plush
- Toothbrush
- Hard Cover Book
- Phone Charm
- 3DS / Vita
- A variety of hats
- The cord player's use
- ETC.

Player (s') Actions

- Movement
- Button Presses / Button Holding

Player (s') Interactions

- Dragging cable around
- Pushing objects around to get them closer to the cable

Overhead Projector



Overview

Players must move images in a specified order onto a light source to project an image out to viewers. Like Tetris battle Gaiden, images appear in the middle of the screen and show the order to be done. Images will be scattered all around the arena and must be scoured for and dragged onto the light and held still for X amount of seconds before moving onto the next image.

Rules

Visible:

- Where to place the image across the projector
- What image to place
 - Time until the game is over
 - Number of correctly placed images
 - Colour gradient showing timing for holding the image across the light
 - Outline of the correct image to place

Invisible:

- The order of images
- The actual time required to hold the image in place
- Number of incorrect images
- Time for light bulb to fail

Physics & Movement

- Will be 3D movement on a tabletop surface.
- Physics will be in 3D. To be able to move objects off the surface of the table and see them drop
- Or to see objects rotate in multiple directions in real time.

Affordances

- Bright outline surrounding image players must retrieve to place across the light
- Where to hold the image
- How long to hold it in place shown by changing colours on the four corners when held in the correct space. With colour blind options available
- Flickering of the light bulb failing to show that it needs to be changed

In-Game Objects

- Images
- Projector
- Light Bulb(s)

Player(s') Actions

- Movement
- Jumping
- Button presses
- Button holding

Player(s') Interactions

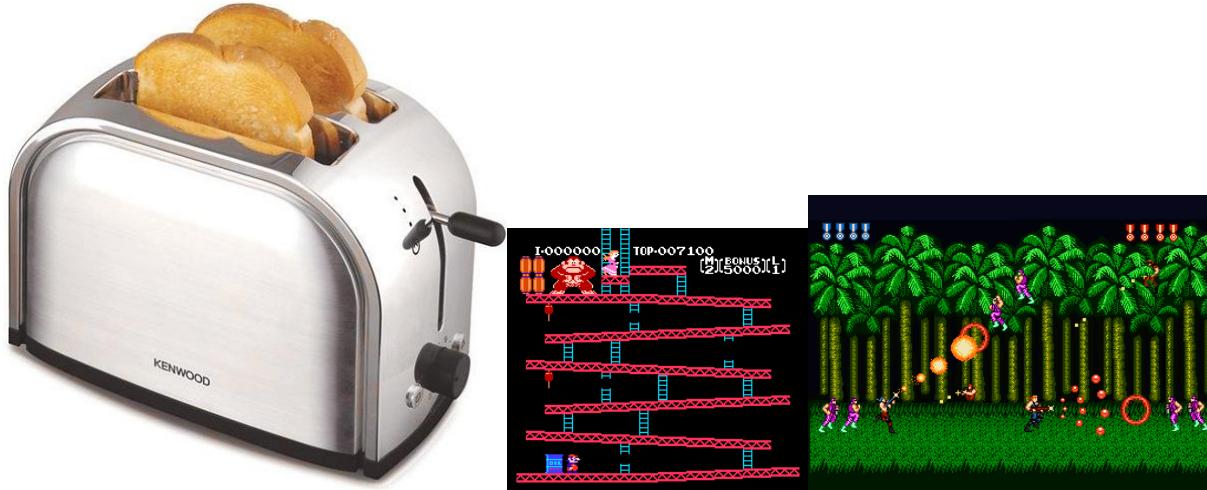
Grabbing images

Moving images into place

Removing / replacing light bulb when it fails

Tossing images off projector when finished to make room for new ones

Toaster



Overview

Teams are tasked with cooking a piece of bread using flamethrowers. They have to climb around a map that looks a lot like Donkey Kong. The player have to use the flamethrower to cook the slice of bread behind them.

This game plays like combination of Contra and Donkey Kong. Each player can jump and move left and right. They can also fire in eight different directions. (Up, Left, Down, Right). The area that is being hit by the flame is then begins to cook.

Rules

Visible:

- Player can only move in a 2D space.
- They shoot their flamethrower to cook the bread behind them.
- Once all of the bread is cooked the game is ended.

Invisible:

- The bread is split up into a grid.
- Each grid square can be cooked and is giving a value relative to how cooked it is.
- 0 being uncooked and 80+ being burnt.
- If the player is cooking a square and facing left, the square to left will start to cook at a slower rate.
- Once fished an average is then calculated by adding up all of the squares.

Physics & Movement

- Will be 2D movement on a Donkey Kong like level.
- Physics will be in 2D.

- Or to see objects rotate in multiple directions in real time.

Affordances

- If a player cooks a spot for too long the spot burns
- If the players burn too much the smoke alarm goes off and they fail.
- Cooking spreads to attached squares.

In-Game Objects

- Platforms for players to walk
- Flame throwers for the players to use

Player (s') Actions

- Jump and move left and right
- Use flamethrower to cook toast

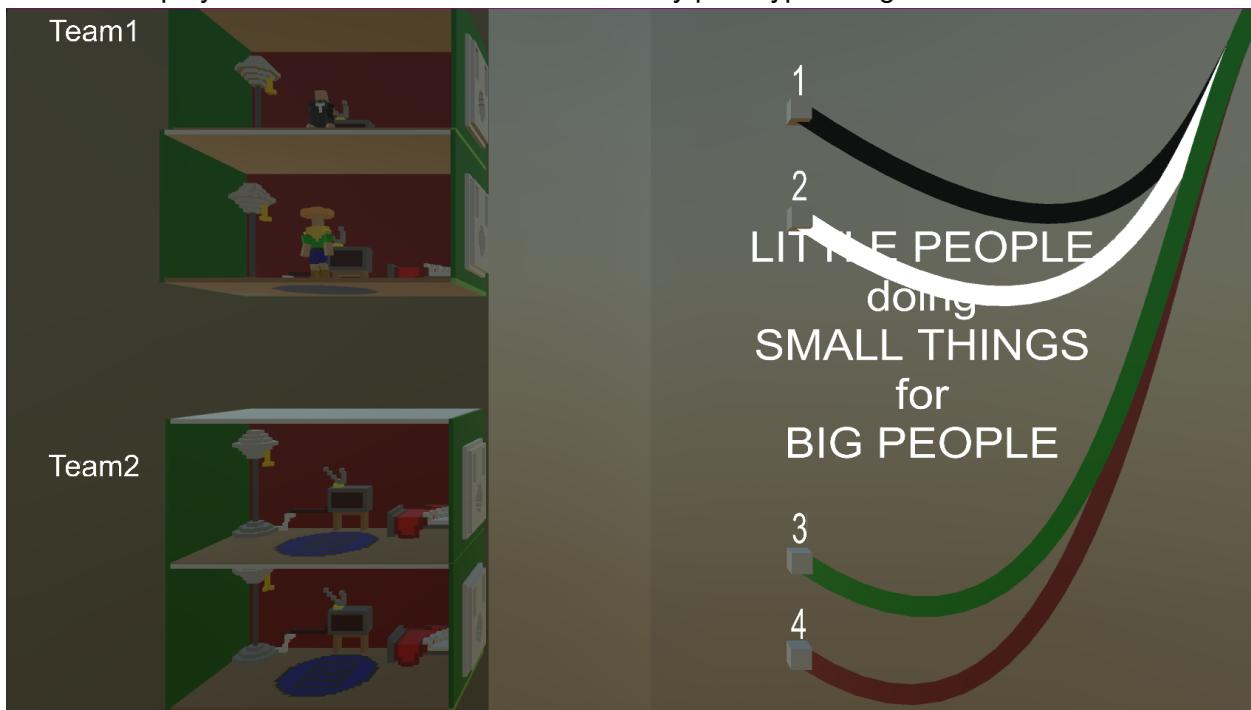
Player (s') Interactions

- Cook the toast.

User Interface

Visual system (visual to player)

The player select will look similar to this early prototype design.



There will be a game select menu, that will be similar to a breaker in a home. Where each breaker will represent a mini-game that players can choose between.



With an on and off state to determine which mini-games are to be played by players.

Control Scheme

Movement is handled by the left control stick on a controller, or by WASD on a keyboard, with an interact button available on both control options to interact with objects in each level.

Completion Timeline

Currently Completed Items

Git Hub Repository

Roomba Alpha

Menu Alpha

Objectives

Name of Item: GDD

Estimated Completion Date: Jan 24th, 2017

Task Begun: Yes

Description / Justification: This document is a requirement to show what we want our game to be at the end of production. With a list of requirements, like mini-games, characters and story, as well as control methods that will show that we've thought about

Name of Item: Timeline

Estimated Completion Date: Jan 24th, 2017

Task Begun: Yes

Description / Justification: This part of our project is important as it establishes firm deadlines that have to be met so as to have a finished capstone.

Name of Item: Menu structure

Estimated Completion Date: February 27th, 2017

Task Begun: Yes

Description / Justification: A good menu can be just as good as a game in and of itself. Having a clear, easily understandable menu will make the game more enjoyable, and accessible as a result.

Name of Item: Roomba

Estimated Completion Date: Jan 27th, 2017

Task Begun: Yes

Description / Justification: The first mini-game we used to test our idea and show off at a number of events. Allowing some to play the game and get a grasp of what the entirety of our game would end up looking like.

Name of Item: Gun

Estimated Completion Date: Feb 17th, 2017

Task Begun: No

Description / Justification: This game will be the next, alongside boombox to be created. This will focus around taking the different parts of a dart gun to put together and shoot at the opposing team. With VR being the player watching the gun being built and shooting the gun at the other player.

Name of Item: Toaster

Estimated Completion Date: February 10th, 2017

Task Begun: No

Description / Justification: The original idea for the game, and used for the pitch. This mini-game centres around players climbing up and down the inside of a toaster cooking toast with a flame thrower to the right temperature without burning the toast to a crisp.

Name of Item: Boombox

Estimated Completion Date: February 3rd, 2017

Task Begun: No

Description / Justification: This mini-game will be one of the more complex mini-games to create, with the necessity of detecting what song was input into the tapedeck and then having the player choose a gramophone to use. Rotating the handle on the gramophone to play the music at the right speed.

Name of Item: Alpha Release

Estimated Completion Date: February 20th, 2017

Task Begun: Yes

Description / Justification: First official release of the game, with multiple mini-games playable during a presentation. No VR capability as of yet, with four different selectable characters integrated.

Name of Item: Models

Estimated Completion Date: January 30th, 2017(Initial), March 1st, 2017(Final)

Task Begun: Yes

Description / Justification: Models to be used for both the initial and final release of the game. With the initial release being the models we've used prior, and with the final models being polished and professionally prepared. Both textured and rigged up to be used.

Name of Item: VR Support

Estimated Completion Date: February 20th, 2017

Task Begun: No

Description / Justification: This will come after the mini-games are completed. It will add a new player with a completely new viewpoint to the games. Allowing them to interact freely with the other players working on the mini-games and add a new layer of interactivity that they wouldn't have seen from playing with the controller inside of the appliances.

Name of Item: Final Release

Estimated Completion Date: March 22th, 2017

Task Begun: Yes

Description / Justification: The final release build of the game to be displayed and shown off to employers and as our final capstone project for marks.

9-Week Timeline

Task	Start Date	Duration	End Date
Audio	20-1-17	4	19-4-17
Boombox	20-1-17	2	3-2-17
Boombox VR	17-2-17	2	20-3-17
Final Release	16-11-13	1	22-3-17
GDD	13-11-16	2	24-1-17
Gun	3-2-17	2	10-2-17
Gun VR	17-2-17	2	20-3-17
Menus	13-1-17	2	27-2-17
Models	30-1-17	4	1-3-17
Roomba	13-11-16	3.5	27-1-17
Roomba VR	17-2-17	2	20-3-17
Sketches	20-1-17	2	19-2-17
Timeline	13-1-17	2	24-1-17
Toaster	20-1-17	3	28-3-17
Toaster VR	17-2-17	2	20-3-17
Alpha Release	13-11-16	3	20-2-17

Appendix

Mini-Games

Printer - A mini-game where players are given an image and forced to manoeuvre ink cartridges with the other players controlling the Red, Green and Blue values of the ink.

Egg-Beater - A mini game where players will control the rotations of the whiskers attached to the egg beater and control the angle of the beater to manoeuvre around a bowl.

Virtual Reality (VR)

Virtual Reality compatibility.

Adding an additional player character that utilizes a VR headset to interact and play with the main players in different ways with some examples following

Balloon Pump

The player in the VR helmet will be able to reach and grab balloons to use to assist the players on the table to help them pump each balloon; or pop the opposing team balloon to force them to restart. As well as players being able to block and prevent the other player from popping your own balloon.

Phone Switchboard

Players that are playing the game will be in control of manoeuvring the ends of wires into a variety of objects. VR Players will receive commands from the main board only they can see and must attempt to work co-operatively with the players to connect calls together. Objects can vary from squares to CDs and shipping crates; VR players must search for the object and bring it to the players with the wire to connect.

Gun

Players on the table will be loading / assembling the gun for the VR player to pick up and shoot the other VR player. VR players can interact with the table players by cheering them on, bringing weapon components closer to the table players to allow faster building of the weapon before shooting it. (Could be spun out into something bigger)

Toaster

Players in the toaster will be interacted with via VR by inserting toast and when the toast is done being cooked. VR players must be ready to catch and stack the toast on a plate. While, having to worry about preventing the opposing team from unplugging their toaster.

Tape deck

Players in VR will be able to select the tape that the players in the deck must play. The players in the deck will be tasked to control the speed of the song playing which the VR player will set by turning a dial. Which will display an updated BPM inside of the tape deck. VR Players must prevent the other team from knocking their tape deck off of the table. Knocking the pieces around and forcing them to be picked back up and set down again to be usable.