Puppeteer - Technical Specification

May 24, 2019

Group Members

Sandra Andersson	Dev
Carl Appelkvist	Dev
Ludvig Björk Förare	Dev
Anton Jonsson	Dev
Kristoffer Lundgren	Dev
Filip Rehnman	Dev
Philip Stenmark	Dev
Benjamin Vesterlund	Dev
Oscar Leiner Olsson	Artist
Gustav Mårdestam	Artist
Robert Ringholm	Artist

Revision	Date	Name	Comment
A	2019-04-11	Anton Jonsson Philip Stenmark Kristoffer Lundgren	Milestone 2
B	2019-04-16	Sandra Andersson Philip Stenmark	Revised for implementation of player, interaction, weapons and health.
C	2019-04-18	Sandra Andersson	Revised for implementation of traps.
D	2019-05-24	Anton Jonsson	Implemented into LaTeX
E	2019-05-24	All developers	Final cleanup for Milestone 5

Contents

1 Game Mechanics

1.1 Platform and OS

Windows 7 or higher (64-bit only)
i7 - 6700k @ 4.00GHz or AMD equivalent
Nvidia Geforce GTX-980 Ti or AMD equivalent
16 GB RAM

1.2 External Code

Unity 2018
Oculus SDK for VR-support
Mirror by vis2k for networking
FMOD for sound

1.3 Game Objects

1.3.1 Level

Components

- LevelBuilder
  Attributes
  - List<GameObject> MultiDoorRooms
  - GameObject StartRoom
  - GameObject GoalRoom
  - GameObject Door
  - Int RoomsToSpawnBeforeDeadEndRooms
  - List<RoomCollider> roomColliderPosit ions
  - Queue<AnchorPoint> openDoorQueue
  - List<GameObject> roomsToBePlaced
  - RoomTreeNode startNode
  - GameObject parent

  Methods
  - Start
  - RandomizeRooms
  - SpawnRooms
  - SpawnRoomsOnNetwork
  - GetRooms
  - ConnectDoorsInRoomIfPossible

- ItemSpawning
  Attributes
  - uint NumberOfSpawns
  - List<SnapPointBase> spawners
  - GameObject level
  - List<GameObject> WeaponList
  - GameObject localSpawner
  - ItemsToSpawn

  Methods
  - Awake
  - FindSnapPoints
  - SpawnItems
  - RpcSetParent
  - GetRandom
  - SpawnWeapon
  - SpawnAmmo
  - SpawnPowerUp
  - SpawnMedkit
  - SpawnItem

1.3.2 Room

Room should be synced with Mirror
Tag

- Connectable

Components

- CooldownComponent
  Attributes
  - float CooldownTime
  - bool Available

  Methods
  - get/set Available
  - StartCooldown
- Mesh Renderer
- Box Collider

1.3.3 Puppet

Tag

- Player

Components

- HealthComponent
  Attributes
  - uint Health
  - uint MaxHealth
  - uint MaxRegenHealth
  - float MaxRegenRatio
  - float MaxReviveRatio
  - uint RegenSpeed
  - uint RegenDelay
  - bool AllowRegen

  Methods
  - Damage
  - Revive
  - AddDeathAction
  - RemoveDeathAction
- InteractionController
  Attributes
  - Lookahead

  Methods
  - Update
- Interactable (acts as base class for all in-level interactable items)
  Methods
  - OnInteractBegin
  - OnInteractEnd
  - OnRaycastEnter
- Image (UI image for compass)

- PowerUpBase (acts as base class for all power-ups)
  Attributes
  - int Duration

  Methods
  - OnActivate
  - OnComplete
- StaminaPower : PowerUpBase
  Attributes
  - float SpeedModifier
- InvisibilityPower : PowerUpBase
- NavigationPower : PowerUpBase
- SnifferPower : PowerUpBase
- PlayerController : Interactable
  Attributes
  - float MovementSpeed
  - float AccelerationRate
  - float SprintSpeed
  - float MaxStamina
  - bool DisableInput
  - float JumpForce
  - float JumpRayLength
  - float MouseSensitivity
  - bool HasMedkit
  - GameObject CurrentWeapon
  - int Ammunition
  - bool CanShoot

  Methods
  - Update
  - FixedUpdate
- Mesh Renderer
- Rigidbody
- Capsule Collider (For collision detection)
- Animator

1.3.4 Puppeteer

Components

- Mesh Renderer
- GrabTool
  Attributes
  - LevelBuilder level
  - int SnapDistance
  - float RaycastDistance
  - float LiftHeight
  - float LiftSpeed
  - bool EnableMovement
  - GameObject sourceObject
  - GameObject selectedObject
  - GameObject guideObject
  - AnchorPoint bestSrcPoint
  - AnchorPoint bestDstPoint
  - RoomInteractable lastHit
  - Vector3 grabOffset
  - RoomTreeNode firstParentNode

  Methods
  - Start
  - Update
  - Pickup
  - Drop
  - UpdatePositions
  - FindNearestOpenDoor
  - CanConnect
  - MouseToWorldPosition

- ItemGrabTool
  Attributes
  - GameObject[] HudItems
  - Button (All HUD buttons used for the spawning of items and enemies)
  - GameObject (All the items that can be spawned by the puppeteer)

  Methods
  - Start
  - Update
  - SetPrices
  - ClientUpdate
  - Pickup
  - Drop
  - FindNearestFreePoint
  - CanBePlaced

1.3.5 Enemy

Tag

- Enemy

Components

- HealthComponent
  Attributes
  - uint Health
  - uint MaxHealth
  - uint MaxRegenHealth
  - float MaxRegenRatio
  - float MaxReviveRatio
  - uint RegenSpeed
  - uint RegenDelay
  - bool AllowRegen
  - bool Downed

  Methods
  - Damage
  - Revive
  - AddDeathAction
  - RemoveDeathAction

- PathFinderComponent
  Attributes
  - bool UseRootMotion
  - float MovementSpeed
  - float RotationSpeed
  - float LegHeight

  - int MaxRecursionDepth
  - Vector3 TransformRaycastOffset
  - float NodeArrivalMargin
  - bool PathfindDebug

  - float RaycastAvoidDistance
  - float RaycastAvoidAngle
  - float RaycastAvoidWeight
  - float MinionAvoidDistance
  - bool ObstacleAvoidDebug

  - float MinVelocityThreshold
  - float StuckTimeThreshold
  - float UnstuckRadius

  - float DoorInteractRange
  - float ForceSmoothingValue
  - float ClampValue

  Methods
  - MoveTo
  - Stop

- StateMachine
  Attributes
  - uint tickRate
  - bool MinionType
  - EnemySpawner Spawner
  - HealthComponent TargetEntity
  - PathfinderComponent PathFinder
  - List GameObject Puppets
  - float AttackCooldown
  - uint AttackDamage;
  - float AttackRange
  - float AttackEscapeDistance
  - float ChargeAccelerationSpeed
  - float StartChargeSpeed
  - int ChargeDamageMultiplier
  - Collider[] HitColliders
  - float AggroDropTime
  - float InstantAggroRange
  - float ConeAggroRange
  - float FOVConeAngle
  - float MinIdleTime
  - float MaxIdleTime
  - Vector3 RaycastOffset
  - bool debug
  - float ChooseCurrentRoomChance

  Methods
  - SetState
  - Update
  - CheckProximity
  - Die
  - Despawn
  - Attack
  - WithinCone
  - RemoveY
  - getNearbyDestination
  - RoomContainsPlayer

States (classes)

- AttackState
  Methods
  - Enter
  - Run
  - Exit
- ReturnToSpawnerState
  Methods
  - Enter
  - Run
  - Exit
- WanderState
  Methods
  - Enter
  - Run
  - Exit
- SeekState
  Methods
  - Enter
  - Run
  - Exit
- IdleState
  Methods
  - Enter
  - Run
  - Exit
- BigAttackState
  Methods
  - Enter
  - Run
  - Exit
- ChargeAttackState
  Methods
  - Enter
  - Run
  - Exit

- Network Transform
- Network Identity
- Network Animator
- Minion Sounds

- Studio Event Emitter
- Rigidbody
- Capsule Collider
- Animator

1.3.6 EnemySpawner

Components

- EnemySpawner
  Attributes
  - GameObject EnemyPrefab
  - GameObject spawnedEnemies
  - bool TankSpawner
  - int MaxEnemyCount
  - Transform spawnPoint
  - int MinDelay
  - int MaxDelay
  - List<StateMachine> LocalMinions
  - List<StateMachine> AllMinions
  - Transform MinionContainer

  Methods
  - Start
  - Spawn
  - CmdOnTakeDamage
  - RpcPlayAnimation
  - OnDeath
- Transform
- Network Identity
- Snap Fucntionality
- Custom Network Transform
- Enemy Spawner Sounds
- Puppeteer Item Sounds

- HealthComponent
  Attributes
  - uint Health
  - uint MaxHealth
  - uint MaxRegenHealth
  - float MaxRegenRatio
  - float MaxReviveRatio
  - uint RegenSpeed
  - uint RegenDelay
  - bool AllowRegen

  Methods
  - Damage
  - Revive
  - AddDeathAction
  - RemoveDeathAction
- Box Collider

1.3.7 Weapon

Tag

- Weapon

Components

- WeaponComponent : Interactable
  Attributes
  - int Capacity
  - int LiquidLeft
  - int LiquidPerRound
  - float FiringSpeed
  - int Damage
  - float Spread
  - float RecoilAmount
  - float ReloadTime
  - int NumberOfShots

  Methods
  - Use
  - Reload
  - OnInteractBegin
  - OnRaycastEnter
- Mesh Renderer

1.3.8 Trap

Tag

- Trap

Components

- TrapComponent
  Attributes
  - uint Damage
  - float ActivationTime
  - float DestroyTime
  - List<GameObject> Puppets
  - Animator Anim

  Methods
  - TrapTimer
  - DestroyTimer
  - ActivateAnim
  - OnTriggerEnter
  - OnTriggerStay
  - ActivateTrap
  - DestroyTrap
- Box Collider
- BasicTrap : TrapComponent
- BearTrap : TrapComponent
- BearInteract : Interactable
- FakeItem : Interactable
- SnapFunctionality
- Mesh Renderer
- Animator

1.3.9 Door

Tag

- Door

Components

- DoorComponent : Interactable
  Attributes
  - (private)bool locked
  - (public with getter and setter)bool Locked
  - Float RotationSpeed
  - Float openAngle
  - Float defaultAngle
  - Vector3 adjustmentVector
  - Float currentAngle
  - Bool open

  Methods
  - Start
  - OnInteractBegin
  - FixedUpdate
  - OnInteractEnd
  - OnRaycastEnter
- Mesh Renderer
- Box Collider

1.3.10 Compass

Components

- CompassComponent
  Attributes
  - List<GameObject> targets
  - Vector3 north

  Methods
  - AddObject
  - RemoveObject
  - Update

1.3.11 PowerUpRecharge

Components

- MeshRenderer
- Recharge/Refill : Interactable (used to refill the player's power)
- Box Collider

1.3.12 Medkit

Components

- Medkit : interactable
  Methods
  - OnInteractBegin
  - OnInteractEnd
  - OnRaycastEnter

1.3.13 HUD

Components

- HUDScript
  Attributes
  - Transform Owner
  - RectTransform HealthBarFill
  - RectTransform StaminaBarFill
  - RectTransform MedKit
  - RectTransform PowerUpFill
  - Text CurrentAmmo
  - HealthComponent healthComponent
  - PlayerController playerController
  - PowerupBase powerUp
  - uint health
  - uint maxHealth
  - float xScaleHP
  - float xScaleStamina
  - float HPIncrement
  - uint previousHP
  - uint lerpToHP
  - float HealthBarSpeed
  - bool medKitToggle
  - float previousStamina
  - float lerpToStamina
  - float staminaIncrement

  Methods
  - Start
  - Update
  - DrawHealthBar
  - DrawStaminaBar

1.4 Control Loop

Unity handles everything that has to do with the control loop.

1.5 Data Flow

Unity handles the storing and loading of all files and assets. Every component written by the dev team should aim to load everything that is going to be used frequently into memory during the loading phase in the unity Start() and Awake() functions.

1.6 Game Physics and Statistics

1.6.1 LevelBuilder

The LevelBuilder Start() method should only happen on the server. When spawning a new room as the host, you need to specify that with Mirror.

Start
: All room modules that may be included in the level generation are provided in the Modules game object array with an exception of the start and goal rooms, which are provided separately as StartModule and GoalModule. Once the LevelBuilder is started, the level is generated in its entirety. It is however required that the start and goal rooms are reachable from each other.

1.6.2 ItemSpawner

The ItemSpawner is run when all rooms have spawned in. It runs once per rooms and spawns items correctly parented to avoid issues when moving rooms.

Awake
: Adds all weapon prefabs to the weapons list.

FindSnapPoints
: Finds all Snappoints in the current room. After first time, just return the points without the search.

SpawnItems
: Gets all potential spawners, then choses a set number of them to spawn a item, by silently clamping the set percentige then randoming a item or weapon.

GetRandom
: Produces a list with length of num with unic random numbers between min and max.

1.6.3 HealthComponent

The Health attribute should be synced with Mirror, meaning all methods changing health must be synced (Heal & Damage must happen on the server)

Damage
: Damage is a method used for removing health from the Health attribute in HealthComponent. It takes the amount of health to lose as a parameter. This function is called from either the WeaponComponent on the player or from the enemy if it does damage to a player. If the health reaches zero upon taking damage, some action need to be handled, i.e. the player downing, spawner exploding or enemy dying. This is done using by registering actions using AddDeathAction.

Revive
: Revive is a method used for restoring the health to a certain state. The health is always restored to MaxHealth * MaxReviveRatio.

AddDeathAction
: Registers a new action used when the health reaches zero.

RemoveDeathAction
: Unregisters a registered action.

1.6.4 PlayerController

The player must be synced with Mirror. Internal variables such as ammo and movementSpeed does not unless implementing anti-cheat.

Update
: Handles player movement using inputs. Limits velocity according to the MovementSpeed.

OnInteractBegin
: The interactor attempts to start reviving the player if below zero health. The function also verifies that the interactor has a medkit to perform this action. The revive action is not instant, but instead uses the ReviveTime attribute.

OnInteractEnd
: The interactor will stop the revive if not done already.

OnRaycastEnter
: If the player is downed and the player that is looking at the downed player has med Med Kit, show the interact tooltip.

1.6.5 PowerUpBase

PowerUpBase is used as a base class for all the individual puppet powerups.

OnActivate
: Override this function to specify behavior when the power up is activated. The activation will start the power up usage countdown using the Duration. The OnActivate function may be implemented as a coroutine to simplify logic.

OnComplete
: Override this function to specify behavior when the power up is suspended. The completion function may for example remove the Pekko's goal icon from the compass.

1.6.6 StaminaPower : PowerUpBase

Froggo's individual powerup. Increases speed and stamina to allow Froggo to run through traps before they are activated. Inherits from PowerUpBase.

1.6.7 InvisibilityPower : PowerUpBase

Gekko's individual powerup. Makes Gekko invisible by changing shaders for puppets and puppeteer. Puppets will see a different color while puppeteer sees nothing. Inherits from PowerUpBase.

1.6.8 NavigationPower : PowerUpBase

Pekko's individual powerup. Shows the direction of the goal on the Compass by adding a new target to the CompassComponent. Inherits from PowerUpBase.

1.6.9 SnifferPower : PowerUpBase

Doggo's individual powerup. Shows if every door is locked by displaying a colored paw-print on the doors. Red for locked, green for unlocked. Inherits from PowerUpBase.

1.6.10 GrabTool

Update

Checks all inputs relevant to the grabbing and dropping mechanic of the puppeteer and calls the relevant methods depending on the output. Also updates the currently selected room and placement guide depending on mouse/VR hover.

Pickup

Pickup works with both items such as spawners and traps, and also rooms. For rooms specific rules need to be followed: Pickup picks up the selected room from the puppeteers perspective. The pickup function creates a copy of the selected object that can be lifted from the ground. The source object is however still in place until a new position is fully selected. Furthermore, a guidance object is also created from a lightweight clone of the source object that acts as an outline and views the most optimal placement option.

Drop

Drop has to do different checks depending on if you are holding a room or just a trap/spawner. For a room, it drops the picked up room into the most optimal position in the level layout, if any. When a room is dropped, the selected object clone and its guide object is destroyed. If a dropped room does not fulfill all rules for connecting (using the CanConnect function), it is discarded.

CanConnect

Returns a bool telling whether or not a room can be dropped in a certain place. The function should handle all placement rules. A source anchor point and destination anchor point should be provided. To modify the maximum snapping distance between anchors, the SnapDistance property is used. The connection rules are:

- Both anchors must be valid.
- Anchors must be open for connection (not already used).
- The forward vector of the anchors must be opposite of each other.
  - Vector3.Dot(srcForwardNormalized, dstForwardNormalized) == -1
- The distance between the anchors must be less than the SnapDistance.
  - Vector3.Distance(srcPosition, dstPosition) $<$ SnapDistance

FindNearestAnchor

Locates the nearest anchor (door) to the selected object, if any.

1.6.11 ItemGrabTool

Start
: Gets all the components and values needed.

SetPrices
: Sets up the HUD to display the correct cost for all items that can be placed.

Update
: Checks for inputs from the player and calls other functions depending on the input.

Pickup
: Instantiates the item you want to place add makes it follow your cursor.

Drop
: Places down the item on the position of the guide object.

FindNearestFreePoint
: Finds the nearest point on the map where the item can be placed

CanBePlaced
: Checks if the item can be placed down on the current snap point.

1.6.12 InteractionController

Update
: The update function raycasts straight ahead and looks for objects with the Interactable component. In case the player presses the Use button, an interaction is started. If a player hits an object with the Interactable component the OnRaycastEnter function is called.

1.6.13 PathfinderComponent

MoveTo
: Pathfinds to coordinates specified in parameter.

Update
: Follows the current path by steering/rotating the object and moving it forwards. If no path exists, do nothing.

Stop
: Aborts the current pathfinding

1.6.14 StateMachine

CheckProximity
: Checks for players nearby.

SetState
: Changes state.

Update
: Runs the current states Run function.

1.6.15 EnemySpawner

Needs to be synced with Mirror

Update

Spawns an EnemyPrefab instance at the position of the member SpawnPoint whenever it's coroutine timer hits zero. The min-/maxDelay variables determin the range in which the spawntime can vary. Number of spawned minions per spawners is capped to the attribute MaxEnemySpawner.

1.6.16 WeaponComponent

Use

Attempts to fire the weapon. The function may immediately fail if there is no ammunition left or if the time since last weapon use does not exceed the firing speed. The weapon may also not be fired while reloading. In case the firing was successful, the following steps are performed:

- For each bullet in the total number of simultaneous bullet fired:
  - Calculate bullet spread using the Spread variable
  - Raycast directly from the viewport center into the direction of the camera plus the angle deviation from the spread
    - ViewportPointToRay(new Vector3(0.5f, 0.5f, 0.0f))
  - Collect the raycast hit information and fetch its HealthComponent
  - If there is a health component, deal some damage using the Damage variable
  - Create bullet impact effect or other effect
- Play appropriate firing animation and effects
- Perform recoil physics on the weapon
- Play appropriate weapon firing sound
- Reset firing speed timer
- Decrease ammunition count

Reload

Reloads the weapon using the ammunition the weapon owner is carrying. The function attempts to fill the weapon to its maximum Capacity, if possible. If the weapon is fully loaded, its LiquidLeft will equal its Capacity. A reload sound and animation is also played.

OnInteractBegin

The interactor picks up this weapon. If the interactor already has a weapon component attached, the new weapon is automatically swapped for the new one. Some logic is required here to properly attach transforms between weapon and interactor.

OnRaycastEnter

Shows the interact tooltip to the player that is looking at the weapon.

1.6.17 TrapComponent

Must be synced with Mirror.

OnTriggerEnter
: Override this function in the subclass for detecting when a game object is within the trap activation area. The function shall immediately trigger the TrapTimer function. If the timer reaches ActivationTime, the trap and activation animation is activated and is therefore considered consumed.

OnTriggerExit
: Override this function in the subclass for detecting when a game object leaves the trap activation area. The function shall be used for handling damage on targets when the trap is activated.

TrapTimer
: This timer is started when a puppet enters the collider and is used for activating the trap after an amount of time. When the timer runs out it will start the animation for the trap.

DestroyTimer
: This timer is started when the trap has activated and the animation has finished. When the timer runs out it will destroy the object for the trap.

ActivateTrap
: Override this function in the subclass for handling the activation of the trap and damaging the puppet(s) inside, starting the DestroyTimer and other optional functionality. Function is often called in within the animation in an animation event.

DestroyTrap
: Override this function in the subclass for handling the destroying of the trap object after the DestroyTimer is done.

1.6.18 BasicTrap

Is placed on basic traps such as spikes and chandelier.

1.6.19 BearTrap and BearInteract

Both is placed on bear trap. BearTrap is used for the collider part of the logic, and BearInteract for the interaction logic once the trap has been activated by BearTrap.

1.6.20 FakeItem

Is placed on FakeItem for the interactable logic. Since it will relate more to a useable object than previous traps and does not activate upon entering its collider it does not use TrapComponent

1.6.21 SnapFunctionality

Is placed on all placeable traps for identifying with the item placement for the Puppeteer.

1.6.22 DoorComponent

Must be synced with Mirror.

Update
: Checks if the door should open or close depending on its current state.

OnInteractBegin
: Toggles the door from the open and closed positions. The door may not be opened when the attribute Locked is set to true. The rotation of the door is determined by the position of the interactor, meaning it may rotate both inwards and outwards. When the attribute Locked is set to false the door is instantly closed while normally the door opens smoothly.

1.6.23 Interactable

OnInteractBegin
: Override this function in the subclass for handling interaction events.

OnInteractEnd
: Override this function in the subclass for handling end of interaction events.

OnRaycastEnter
: Override this function in order to use the interaction tooltip

OnRaycastExit
: Disables the outline of the object.

ShowToolTip
: Shows the interaction tooltip for the player that is looking at an interactable object.

1.6.24 CompassComponent

AddObject
: Registers a new 'trackable' entity to the compass, e.g. teammates or level objects. Accepts any GameObject as parameter as long as it contains some Icon component.

RemoveObject
: Unregisters a game object from the list.

Update
: Updates all relative positions of the registered objects in the compass user interface. The icon of each object is displayed in the compass UI element. The compass is configured using some predefined 'north' direction. Some nitty gritty maths is here required to map 3D direction vectors to the 2D user interface element.

1.6.25 PowerupRecharge

Must be synced with Mirror.

OnInteractBegin
: Refills the player's power if applicable.

1.6.26 MedkitComponent

Must be synced with Mirror.

OnInteractBegin
: Adds the medkit to the player's inventory if the player doesn't have one.

OnRaycastEnter
: Show the interact tooltip if the player does not already have a med kit in its inventory.

1.6.27 HUD

Start
: Load all the components needed to draw the HUD to the screen, they are loaded inside the start function instead of the update to increase performance.

Update
: Draw all the information about powerups, health, medkits, and stamina to the screen, images are scaled to represent the health and stamina and the scale is lerped to create smooth transitions.

1.7 Artificial Intelligence

The enemies are controlled by a simple Finite State Machine(FSM):

States

- AttackingState
- ReturnToSpawnerState
- WanderState
- SeekState
- IdleState
- BigAttackState
- ChargeAttackState

AttackingState
The enemies enter the attacking state when they have a clear line of sight of the player and they are within range. While in this state the enemies follow the player and automatically attack them when within range.

ReturnToSpawnerState
When the spawner that spawned an enemy is attacked all the spawned enemies will enter this state and return to the spawner in order to protect it. When a spawner or an enemy is moved by the puppeteer this behavior is lost.

WanderState
When an enemy doesn't see any players it will wander on its own to nearby rooms to search for players.

SeekState
This state will pathfind to a players last known location.

IdleState
This state happens between wandering and makes the minion wait for an amount of time between two values.

BigAttackState
The attack state for the tank. The tank can never lose its target unless it dies or turns invisible so it follows and attack when possible.

ChargeAttackState
The tanks special attack, an accelerating charge at its target. The tank doesn't stop until the target is hit or the tank dies. It does damage depending on the tanks speed.

1.8 Multiplayer

Mirror by vis2k solves all of our multiplayer issues. The data is sent using TCP and a client-host system.

2 User interface

2.1 Game Shell

- Splash screen
  The first screen you see when you open the game. After any button is pressed, the menu should be opened.
- Main menu
  The main menu is just a way to get to the other menus.
- Video options
  Video options is where you can change resolution and enable/disable fullscreen.
- Audio options
  Audio options should implement sound sliders for all types of sound in the game.
- Controls
  In controls you should be able to rebind all keys used in the game and also change the sensitivity of the mouse.
- Join
  In Join you enter an IP-address to connect to a host and enter a lobby
- Lobby
  In a lobby you choose character and indicate when you are ready to start the game
- Host/Lobby
  The same as ordinary lobby but for the host of the game.

2.2 Main Play Screens

Puppet

- Health from the HealthComponent displayed as a number.
- Stamina from the PlayerController displayed as a bar.
- Ammunition from PlayerController displayer as a number.
- Power-up status displayed as a bar.
- Compass that points towards other game objects using their transforms to calculate angles.
  - Also displays cardinal directions relative to some fixed point.
  - Uses unique icons to represent different game objects, such as players.
- Ammunition and Capacity from the equipped weapon as numbers

Puppeteer

- Cooldown on rooms from CooldownComponent
- Room selection highlights calculated by the GrabTool
  - Uses some outline shader to display optimal room placement and the selected room's previous position.

3 Art and Video

Unity handles loading, storing and drawing of the art. The animation transitions are done using an Animator Controller.

4 Graphics Engine

Unity handles all graphics rendering in the game.

5 Artist Instructions

Use file formats supported by Unity. Otherwise they have free hands.

6 Sound and Music

Unity has built in support for all the standard audio formats. Sounds are played by an Audio Source and they are picked up by an audio listener which is normally placed on the player's camera. 3D sound, filters, and drop-off are all controlled inside the audio source.

6.1 Sound Engineering Instructions

Use file formats supported by Unity. Otherwise they have free hands. FMOD will be used.

7 Level Specific Code

The level is randomly generated at the start of the game to get a different layout each time. This is done using the LevelBuilder component in the Level game object. It is required that the start and goal rooms always are reachable after each level generation.