

Members

John Ho, jho49@gatech.edu, jho49

Chloe Lynne Choi, cchoi70@gatech.edu, cchoi70

Julie Pierides, juliepierides@gatech.edu, jpierides3

Yi LI, yli813@gatech.edu, yli813

EunBin Dana Lee, danaeblee@gatech.edu, elee358

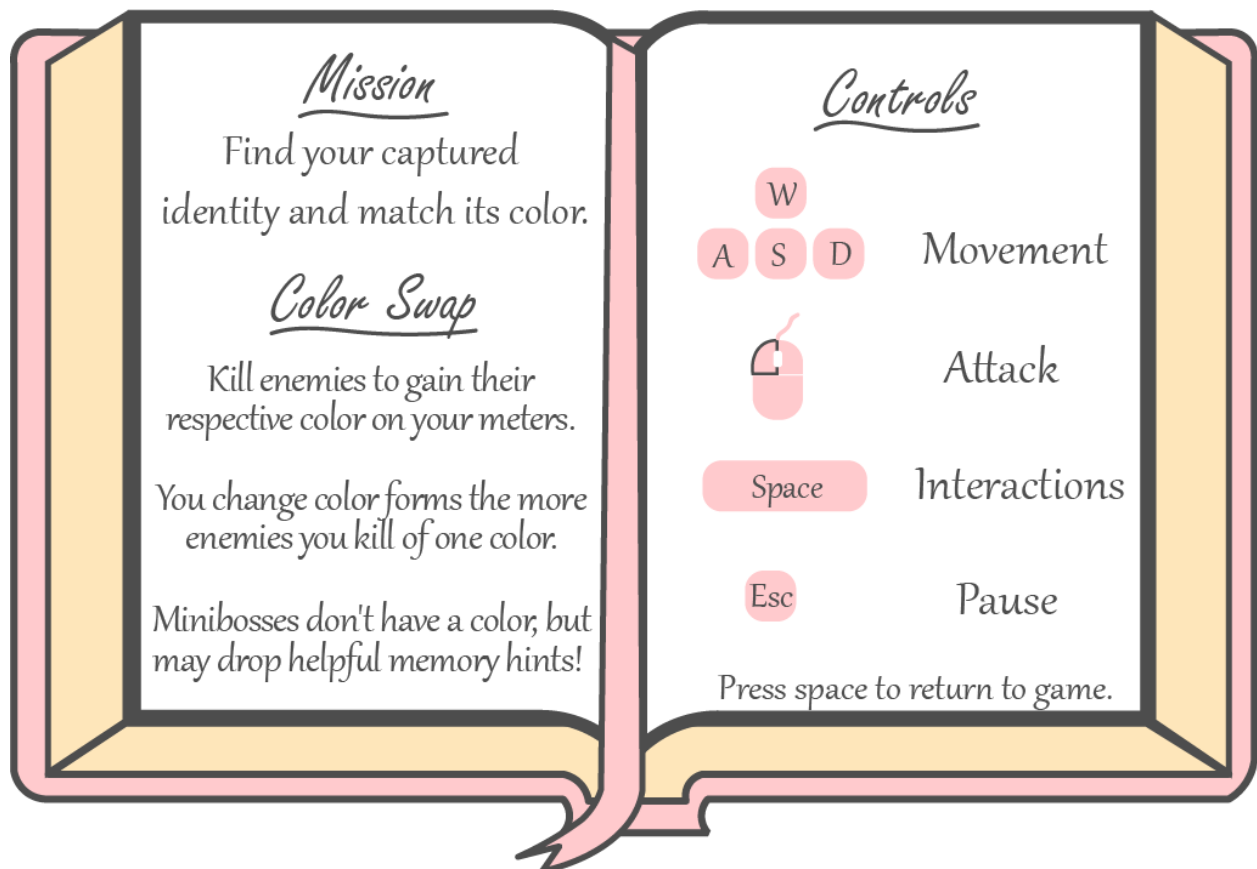
Installation Requirements

No additional installation is required. All assets are retained in the Unity game folder.

Gameplay Instructions

Inside the Build folder, start up the game by double clicking "PiratedIdentity.exe". Then on the main menu, hit play. After a brief cutscene, you will be brought to into the main game.

Below is our controls book detailing the objective and controls of the game:



Rubric Requirements

- *Pirated Identity* is a 3D game that can be defined as a Game Feel game.
- The game has a clearly defined, achievable objective/goal. The player's object is to search for their reflection and match their color. This is hinted during the cutscene, and also in the book instructions in the starting room. The book is placed in one of the corners of the starting room indicated by a ring of blue light. Press the spacebar to read the book. If the player reaches the reflection door without a key, a popup will indicate they need to find a key
- The game communicates to the player clearly whether they have succeeded or failed when the player enters the end room. If you proceed to the end room and match the color of your reflection then win screen will be shown. If you lose, character falls off map, or character health went to 0 then lose screen is shown
- The game has a starting splash screen that can load up the game cutscene as well as the credit menu and the option to quit the game.
- The player is able to reset and replay on win or lose screens.
- Our game is a third person action/shooter
- Goals/subgoals are effectively communicated to the player. The goal of the game is communicated to the player through the opening cutscene. In-game, the starting room contains a book that explains the battle mechanics of the game, controls, and objectives. This book is indicated as important through shiny particle systems as well as a UI that tells the player to press the spacebar to read the book.
- Our game provides interesting choices to the player. Depending on the choices, the player can also face consequences. This is seen in our battle mechanic. Whenever a player defeats an enemy, the respective color bar will be filled. When the player has gathered enough points in one color they will transform into that color. The player has the choice of defeating or hiding from the enemies to strategically transform into the correct color.
- Player choices engage the player. This can be seen when the player rides the boat and the squid comes out to say hi.
- Player choices engage the player with the game world. This can be seen through the hallway trigger element in our game. Depending on the placement of the player and where they need to go.
- Avoid Hollow, Obvious, and Uninformed Decisions. This can be seen when the player receives the hint from the bosses telling the player which color to collect. The player must use the hint in order to make his decisions of which color enemies to kill next.
- Your game must achieve balance. Our game is challenging, but fair. Enemies do a light amount of damage and our bosses do much more. It is conveyed to the player that the bosses do more damage because of their large sizes. Another fair aspect of the game is the boat. It can be a shortcut, but it is a risk to get on it since falling off will automatically kill you.
- Our game rewards player for successes when they defeat the boss and receive hints that allow the player to know what her true identity is. Also, the player can be punished for failure by reaching 0 health and having to restart.

Team Seniors*

- Our game has in-game learning/training opportunities through the interactive book in the starting room.
- Our game has an appropriate progression of difficulty as smaller enemies do less damage and the bosses are easier to kill.
- Our game avoids opportunities for players to trivially beat your game challenges as it requires the player to find the key room, obtain the key, find the end room while being the correct color which requires them to kill enemies.
- Our game does not use tutorial characters, CS 4455 milestone characters, nor does it use a complete prepackages asset. Instead, we created a well animated character via mecanim that blends between different animations based on the player's inputs. The blend tree has a combination of walk, run, and turn animations based on the x and y variables of its motion. This allows the player to walk in short distances but run for longer. The base layer allows the character to move and fall via root motion. You can see these animations by using the WASD keys. There are additional layers for extra animation while moving. These include attacking, getting hurt, and a healing animation. You can see the attacking animation by pressing the left mouse key. You can see the hurt animation when the enemy comes too close to the player. Lastly, you can see the healing animation when you enter the health room. The animations are well-responsive and very smooth.
- Our game's choice of controls is intuitive and appropriate. This is because we chose the common mmorpg style controls. WASD to move the character and the left mouse button to attack. This allows the player to comfortably move the character with their left hand and shoot using their right hand.
- The player model has fluid animation that is well blended. Even when the player is not doing anything, there is an idle motion to make it interesting. There are no instantaneous pose changes, no glitching, unintended teleporting, a ground-based character stuck floating in air.
- Our humanoid characters do not slide or moonwalk. We used root motion, with IK corrections.
- We have low latency responsiveness to control inputs.
- Our camera is smooth with little delay. This allows the player to move quickly during combat without becoming frustrated by the camera.
- Our camera has limited passing through walls, near clipping plane cutting through objects (e.g. player model), etc.
- Auditory feedback on character state and actions (e.g. footsteps, landing, collisions, tire squeal, engine revs, etc.)
- Coupling with physics simulation via animation curves, animation callback events, handoff to ragdoll simulation, IK adjustments to animations, vehicle that breaks up in wrecks, etc.
- We created the entire environment from scratch using 3D cubes and planes as well as prefabs from the asset store for additional props. Obtained seamless texture images to create materials. Created normal maps of the texture images through Photoshop for extra detail.

Team Seniors*

- Both graphically and auditorily represented physical interactions.
 - Player attacking:
 - Sound effect
 - Energy ball is shot out
 - Enemy hurt:
 - Sound effect
 - Enemy is pushed back
 - Player hurt:
 - Sound effect
 - Screen flashes red
 - Health bar decreases
 - Health room:
 - healing sound
 - Screen flashes green
 - Health bar increases
 - Falling off of the stage:
 - Sound effect
 - Bridge switch:
 - Sound effect
 - Bridges on the map and in the environment switch positions
 - Key pickup:
 - Sound effect
 - Key disappears
 - Key icon appears on the player's UI
 - Door opens:
 - Sound effect
 - Door disappears to let the player through
 - Player enters a new, undiscovered room
 - Minimap displays icon of room
- Graphics are aligned with physics representation
- The player is appropriately confined to the playable space through walls. However, the player may fall out and drown into the ocean.
- Our game has a variety of environmental physical interactions including:
 - Interactive scripted objects in the form of triggers that change the layout of the hallways. Keys that can be picked up to open doors.
 - Simulated Newtonian physics rigid body objects in the form of crates and barrels have newtonian physics rigid bodies. Players can move them around and shoot at them.
 - Animated objects using Mecanim such as the squid's animation and programmatic pose changes in the form of a moving boat that allows the player to get on and off. Shortcut around rooms but are hard to get on and off.
- Our game has 3D simulation with six degrees of freedom movement of rigid bodies

Team Seniors*

- Our game has an interactive environment:
 - Barrels and crates can be pushed around and shot at
 - Animated boat can carry the player around the map
 - Health room heals the player
 - Book in the starting room can be read when player is near it
 - Key can be picked up
 - Bridges can be switched
- Our game has consistent spatial simulation throughout the game as running speed remains same regardless of framerate.
- Our game uses a combination of assets from the asset store that have been well modified from its original form.
- Multiple AI states of behavior. AI would have an idle and attack state. When the player is not in the range of the room, the AI will be in idle state. When the player is in the range, the AI will be in attack state.
- The enemy jump up and down and try to reach you when they find you.
- The AI can sense the player enter or exit the room.
- There is no sudden change of animation. All the animation are blended.
- There is sound when the enemy being hit and dead. Visually, you can see the enemy falls back when it is being hit.
- Small enemy deal fair amount damage. The boss would deal more attack in a slower rate than the small enemy. Small enemy and boss would both walk slower than player in order to avoid unbeatable enemy.
- AI has a fair intercept between each attack and gives a chance to the player to run away.
- AI can use barrels as a block to protect himself from the attack of the player while trying to attack player.
- Our game implements a start menu GUI on game load up with the ability to load the cutscenes, a credits screen, and a quit button
- Our game implements a in-game pause menu with the ability to quit. This is done by pressing the ESC key at any point. This includes the main game and during the cutscenes with the additional ability to skip.
- Our game feels like a game from start of execution to the end. This is done with looping states from the start menu, in game, win/lose screens. Player has the ability to quit at any given point.
- There are no debug output visible as we removed team name from alpha and got rid of all debug output print statements.
- Our game has the ability to exit software at any time. The player has the ability to quit on pause menu and on starting screen/win screen/lose screen. Can also quit on the main menu, and you can press ESC key during cutscene to bring up pause menu to quit.
- We have no implemented any “test-mode”/“god-mode” buttons, etc.
- Our game’s GUI elements are styled appropriately and consistently. The elements have been scaled to fit within canvas and readable.

Team Seniors*

- Transitions between scenes are done aesthetically. This is achieved when the Start menu “Play” button loads up the animated cutscenes. Going from cutscenes to in game, or pressing win/lose screen restart button, makes the camera pan from out of the world towards the player.
- Our game’s environment acknowledges the player:
 - Has scripted aesthetic animations through water animations. The waves can be seen through the openings in the rooms.
 - Proximity-based events. A giant squid will appear and float when you get on the boat. More specifically, if you are standing in the boat on either the left or right side of the map, the squid will enlarge and bob.
 - Auditory representation of every observable game event
 - Physics event-based feedback such as particle effects and one-shot audio for collisions of different types
 - Sonifications that map physical properties to audio effects. Examples include sound effects when you enter a healing room.
- Our game has a semi-realistic art style with a pastel and neutral color palette
- Our music themes are from other games that have both dramatic tracks and upbeat tracks such as Super Mario Odyssey, Breath of the Wild, Final Fantasy, and Super Mario Galaxy.
- Our game has appeal because there are no glitches, no way to easily escape the confines of the game world, the player does not get stuck, and the game is very stable

Known Bugs

StartMenu.Unity

- Error: Assets/Fast Cutscene Producer Free/Demo Assets/Misc/Concrete_02.sbsar: Built-in support for Substance Designer materials has been removed from Unity. To continue using Substance Designer materials, you will need to install Allegorithmic's external importer from the Asset Store.

This bug occasionally appears ONLY in the unity editor, not during the game scenes. This bug does not impact the game in any way, just that it's a depreciated asset that we still needed to use for our cutscenes.

MainGame.Unity

- Error: Screen position out of view frustum (screen pos 125.000000, 0.000000, 1000.000000) (Camera rect 0 0 125 100)

Sometimes this bug appears when in Unity editor mode. We aren't sure why this appears, but it may be due to various display screen resizing. However it does not impact the game and does not show an error when running the game.

External Resources used

(List all the external assets we downloaded)

Sounds:

- Start menu music -
 - Hateno Village - Breath of the Wild (<https://www.youtube.com/watch?v=80mIIWlemlk>)
- New BG sound - calming or upbeat?
 - Star Festival - Super Mario Galaxy (<https://www.youtube.com/watch?v=Ih9KffOGPBk>)
- Cut scene music - scary?
 - Catastrophe - Super Mario Galaxy (<https://www.youtube.com/watch?v=dcxNJVPvRwU>) (more dramatic)
 - Tension - Super Mario Galaxy (<https://www.youtube.com/watch?v=uABri40TrGc>)
- Win screen music
 - Fossil Falls - Super Mario Odyssey (<https://www.youtube.com/watch?v=P8cKiffNad0&t=20s>)
- Lose screen music
 - Somnus Instrumental - Final Fantasy XV (<https://www.youtube.com/watch?v=THht1RqCh8U>)
- Enemy hurt sound effect (<https://www.youtube.com/watch?v=N677YtKuq-k&t=20s>)
- Enemy death sound (<https://www.youtube.com/watch?v=2zB4vMz5sVQ>)
- Player hurt sound (<https://www.youtube.com/watch?v=NuwzTYIjyJg>) (last second)
- Health room healing sound (<https://www.youtube.com/watch?v=BXfk4QtMkG8>)
- Collision sound: <https://www.youtube.com/watch?v=RGyfbqD81uM> (first second)
- Falling sound (taken from free online sound database)
- Player attacking sound (taken from free online sound database)
- End room door opening (taken from free online sound database)

Assets from Asset Store:

- Crates and Barrels Pack Volume 1 - Free Version
- Crystal Package
- Fast Cutscene Producer Free
- Free Dungeon
- GE Common Assets
- Lanterns and Candles
- Particle Collection Free Samples
- Sail Character Pack
- SD Character

Individual Contributions

John Ho - Coded C# scripts for various UI screens, pop-ups, hints, elementType and randomType scripts. Handled scene management to switch between different win/lose and pause screen. Implemented minimap and icons on trigger. User testing script contributor and

Team Seniors*

recruiter of four students. Created pop up dialogues and designed hint UIs. Major Q/A tester to detect game bugs and fix issues. Also acted as a GitHub manager ensuring all files were on the repo.

Yi Li - C# code scripts: enemy attack, player attack, player attack object(bullet script), environment set up, enemy handler, enemy movement, enemy room manager, miniboss room manager, health room manager, reflection creator, reflection color, onBoat. Different room events. Setting up the environment with proper code and properties. Simple AI for the enemy. Health bar, element bars, both UI and Code. Environment baking. Set up the player properties. Debugger in the team. Give general coding advice when coding new stuff.

Chloe Choi - Coded C# scripts such as hallway triggers, player to pick up the key, and the end room opening. Created environment from scratch using basic Unity shapes with materials created with custom textures and normal maps created with Photoshop. Imported character models and a combination of animations. Blended animations through mecanim and set up additional layers for attacking, getting hurt, and being healed. Created ragdoll skeleton. Set up the player camera. Created particle effects on the book, the key, and the healing effect on the character. Created proximity based animation on the squid. Created the animation for the boat to move around the map. Ideated base story and brainstormed artistic design.

EunBin Lee - Coded C# script for player changing types based on point system. Created custom player and enemy texture using online Photoshop, and also created the particle game objects that was used for enemy attack depending on player type. Imported models and prefabs for the environment and created starting cutscene, win scene, and lose scene animation. Also contributed in setting up the player environment and scaled objects based on character size.

Julie Pierides - Coded C# script for enemy and player health, added sound effects for player attacking, enemy damaged, player damaged, healing, falling off of the stage, bridge switch, key pickup, door opening, winning, losing, menu scene, and cut scene. Found music and sound effects, found and changed key icon in the UI, found and changed skybox. Also created minimap icons, color icons, and custom pirate font and eye patch for the title screen in Adobe Illustrator. Also created the start screen splash with the characters jumping on the boat with barrels and crates. Contributed to the backstory of the game, wrote playtesting report, wrote playtesting script, and created Likert scale questionnaire on Google Forms for playtesters.

Scene to open in Unity

You should be able to start and play the entire game with the game with the **StartMenu** Unity scene. Alternatively, any other game scene can be looped back to the main game or starting menu.