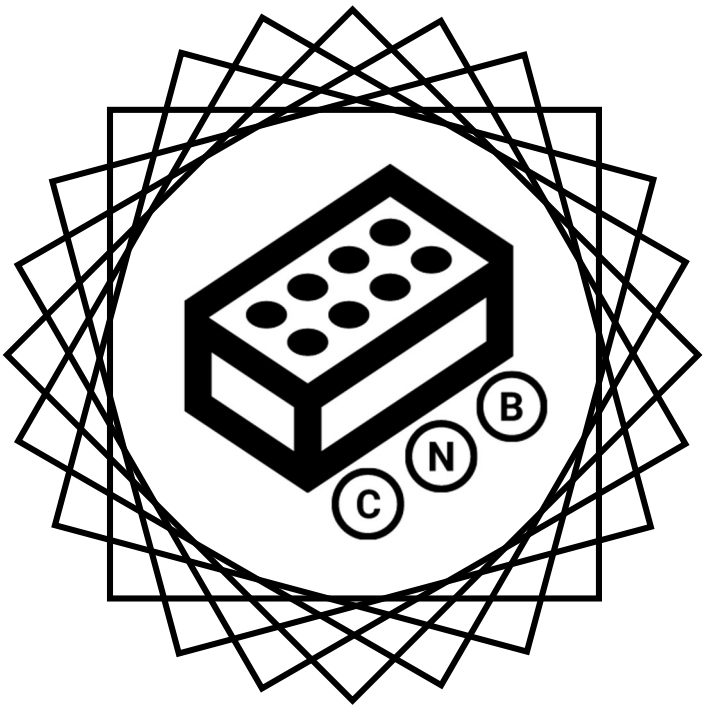
Code Name Bricks



Assignment 3 Part 1 | BITS SP1 | Group 10

The Team

Corbin Peever | S3855159

Connor Edmunds | S3872028

Hamilton Hunter | S3878833

Leonard McDonald | S3879586

Michael McQuarrie | S3884159

Ross Rhodes | S3706950

Table of Contents

Background.

Motivation.

When our team came together, we had a resounding agreement between us that we would all like to create a 2D game, and though it may be an adventurous and, for some of us, slightly self-punishing idea, we all wanted to do it in Java using LWJGL (Light Weight Java Gaming Library). It was our belief that this path for our project held the most merit for us in bettering our understanding of Java, programming, and working together in a group capacity.

Though a few of our group come from strong backgrounds in programming, others have little experience, only having learned what they have picked up during this degree. Despite these challenges, we chose to not go down the easier path of a “Game Maker”. No one on our team second-guessed the decision to program our game from scratch, and no less of us regret this choice (during at least most of the project). For the members that have more experience, this would be an opportunity to dabble in game-creation for the first time, and to learn about structuring a project and providing methods to succeed in game development.

The inspiration, or “target level of fidelity” for our game has been Mario throughout the duration of its development, a goal that may seem easy on paper, but quite difficult to achieve in ten weeks while learning on the job. This would involve swathes of new information, trying and failing, researching, conversations between members, designing, and small successes leading to greater outcomes, eventually, and quite hopefully, becoming a game.

The Team.

Name: Corbin Peever

Student Number: S3855159

Student Email: s3855159@student.rmit.edu.au

Role: Team Leader, Audio Developer.

Personal Background: Effectively my entire career has been spent in the Hospitality and Tourism industry where I was fortunate enough to achieve management positions in a variety of businesses. I have also been involved in music most of my life, completing a Certificate III in Music Industry (Technical Skills), and performing for many people over the years. These two aspects of my experience are the most relevant for my roles in this group as my managerial experience has granted me the ability of good leadership, and my music career has been beneficial in supporting my role as Audio Developer. My technical knowledge is comprised of what I have learned in previous courses of this degree, and through a keen interest I developed in Java and programming, I was able to pick up the required skills to create the code for the Audio related classes.

Name: Connor Edmunds

Student Number: s3872028

Student Email: s3872028@student.rmit.edu.au

Role: Graphics Asset Development.

Personal Background: I had no prior Art experience outside of doodling in my notebook during high school, I had no knowledge at all how to use photoshop software, nor knowledge into the aspects of how a game was made, as most of my career has been spent in hospitality and most of my technical skills lie with IT and the physical hardware surrounding that and its uses and minute details.

Name: Hamilton Hunter

Student Number: s3878833

Student Email: s3878833@student.rmit.edu.au

Role: Level Designer, Level Developer.

Personal Background: At an early age I have enjoyed arts and design, whether that was drawing, building, map creation for video games or designing 3D models using 3D modelling software. My role in this group was creating the platform and hazard classes, building the world our game character will play in and I also managed to pick up some graphics design roles further into games development which was a refreshing change from programming. What I taught myself in my spare time over the years came to good use as I was able to visualise the map while I was programming it and draw the graphics for the platforms and some of the game objects.

Name: Leonard McDonald

Student Number: s3879586

Student Email: s3879586@student.rmit.edu.au

Role: Graphics Developer.

Personal Background: I have always had an interest in the unseen mechanics that make up what digital media is capable of, so much that I had been playing around with anything where my input would create a new outcome. I developed an interest in many forms of software IT and their mechanics, sometimes just from thinking to myself “how can I iterate this process to save myself some time”. During the IT courses that I have completed I have developed minor skills that allow me to create these mechanics myself. I had never been able to guess what loading graphics from a source image into a game would require, and as odd as it sounds, I had no idea it would involve learning about how the graphics card handles data, so I am glad I had an opportunity to research and develop this skill during this group project as it has combined with what I put my interest in and has given me more ideas for the future.

Name: Michael McQuarrie

Student Number: s3884159

Student Email: s3884159@student.rmit.edu.au

Role: Lead-Developer, Testing.

Personal Background: I have a background in game development and have acquired a Diploma in Interactive Gaming from the Media Design School in Auckland. I am trying to get a more diverse IT education so I can return to work in Japan and find new IT jobs rather than being constrained to game development. I can use my experience to help explain the concepts of creating a game engine to the team. This experience should also help avoid some unforeseen problems as I may have come across them before.

Name: Ross Rhodes.

Student Number: s3706950

Student Email: s3706950@student.rmit.edu.au

Role: Co-Lead Developer, Testing.

Personal Background: I have been doing programming as a hobby for 9+ years now and I have dabbled in almost everything along the way. I decided to set my goals on achieving a bachelor's degree in Information Technology to further my personal experience and open more opportunities for my future self. Currently I am a self-contracted Software Engineer working primarily with low-level system languages like C/C++ and assembly, with a few years of experience in reverse engineering assembly code such as malware analysis. I believe my personal experience can be beneficial to other team members and assist in accelerating others learning and understanding of programming from a broad standpoint.

Aim.

We set out to create a 2D, side-scrolling platformer called T.U.R.D (Trash Unit Response Droid), that would use our own member-designed graphics and audio. Our game was to be based on a toilet, a piece of trash that had gained sentience through some strange and unknown event and was forced to battle against other scrap-monsters to survive and gain dominance over its brethren. It was to be set in a world that is dirty, with a somewhat foreboding, fearful, and highly-strung feeling given to the player using a high-risk/high-reward game style, descriptive graphics, and mood-setting audio.

Goals.

1. **Create a 2D platformer:**

Our team has been able to achieve our goal of developing a 2D platforming game, with a level system that varies in elevation, and is comprised of many hazards, enemies, scrap to be picked up, and boss fights. The player can move and jump through the different aspects of the levels, while being restricted by the range of their movement depending on the level of scrap that they have on-hand. The range of the players jump influences their ability to access certain parts of the levels that will eventually reveal bonuses, and special effects.

2. **Team Designed Graphics.**

We have been fortunate enough to have team members with experience designing digital graphics for games and have been able to take advantage of it to create a complex world of animations and graphics made by us. This adds to the overall experience of the game and provides a sense of polish and interactivity. It was our intention to design graphics that were pixelated in style, and comparable to many other platforming games from the nineties and early two-thousands. We were able to achieve this using graphical asset creators made for game images.

3. **Theme is of a scrap-monster battling other scrap-monsters.**

Our initial portrayal of a theme about pieces of scrap coming to life as monsters and battling other scrap-monsters has been realized and is consistent with what we were aiming for in the beginning. The main character is a toilet with wheels, and the main enemies are scrap piles that have grown legs to move around with. The projectiles are pieces of scrap, and it is possible to pick up a bi-product of these scrap-shots, as well as when an enemy is killed. The hazards are spikes and toxic waste, two types of hazards that may be scene in a scrapyard environment.

4. **Exciting gameplay.**

Through the benefits we gained through using our own team-designed graphics and audio, we have been able to create a specific them that is consistent throughout T.U.R.D's gameplay, and in line with what we set out to achieve. This, coupled with the brilliant level design and game physics, leads to our game having a high level of playability that is both interesting and engaging to the user. The game hosts complex navigation through the internet, and if given enough time, it will contain secret areas, and breakable boxes that reveal additional effects for the player.

Progress.

Description.

Our story began like any other, on a Tuesday night with 6 guys sitting in a call trying to think up the possibility for a game, Connor threw his hat into the ring and suggested a role-playing rogue-like game, “like The Binding of Isaac or old an old school Zelda game” he said. Giving the others some ideas, after a little while we could not feel anything good coming together collectively, so we decided to use one of the tactics we learned from that week's course work, where we all think of the worst possible ideas to get the juices flowing. After about 10 minutes or so of some of the worst ideas imaginable, we started to get onto the idea of a sentient robot coming back for revenge by turning on its human creators. Connor then chimed in mentioning how that is a great idea, we could do that, we then begun a slow start with how we could go about making all these things that would be needed of us.

Slowly over the next couple of weeks after the reality of being in a group had finally set in and we began to divvy roles out to each other so we could figure out who would be doing what and what that would entail. Michael, Ross, and Hamilton received most of the coding roles, with Leo being a little bit of an in-between guy, doing a little of everything, but also working to implement graphics. Connor would be working on most of the graphics and designs for the game, while Corbin was the Team leader, he was there to keep everyone in check and up to date and made sure things were going smoothly as well as creating and coding the programs Audio requirements.

Now that we had our roles clearly defined, we set about slowly evolving what we wanted included in the game play. We had started out with wild ideas, compared to the hidden Wumpa fruit from crash, or hard to reach or spot easter eggs. It started out pretty wild with our ideas. We eventuated into using a scrap yard for a background for the game, as it would be easy to animate and we slowly came to realize that may be too bleak and colorless, so we produced a rough plan to start working on what the group felt was right for that person by having a weekly meeting on a Tuesday, and then a follow up meeting on the following night. We had an estimate of how much we wanted to get done by the end of the first assignment, we were right on track by the time the first assignment rolled around.

After a couple of weeks of everyone learning that platform of choice, work had begun on designing the game, Michael, Ross, and Hamilton all worked on perfecting the code the game would be based on, while Corbin, Leo, and Connor, continued to work on their separate part and given MVFs and EVFs. All the graphics were getting a rough design drawn, and then a final sketch, before finally being transferred into a digital medium, and uploaded into the groups Team’s files server.

By the time Week 10 had come to upon our group rather suddenly, we had what felt like something awfully close to our initial goal that we had set upon from the start of the whole project. We decided to omit a couple of things like some EVF’s, for example: The Boss fight at the end had to be cut due to time constraints, so we could not implement everything we wanted, but we were still quite proud of what we had come to accomplish since the beginning.

Outcomes to Date.

Our Outcomes for the project was very close to our target from the beginning, we had to omit a couple of things to shorten the scope of the game and make it achievable within the timeframe. Below is a brief summary of the Minimum Viable Features, if they were achieved, what shows the viable feature, and the Extended Viable Features and whether or not they were included in the final scope of the game.

Minimal Viable Feature 1 – Player Movement.
The player movement was achieved by the time of writing this, the player moves left, right and jumps with accompanying animations while not weighing down the system with complicated code, the player successfully moves across the screen.

Minimal Viable Feature 2 – Enemy Movement.
The enemy movement was based upon the player movement code, just put into a loop that the enemy walks along and attacks the player if nearby.

Minimal Viable Feature 3 – Side Scroller / Platformer.
The side scrolling platformer MVF was achieved based upon the drawn pixel art style, the level design, we did not achieve the full scope of the levels we wanted, albeit close, but we got to having our first full level by the end of the semester.

Minimal Viable Feature 4 – Projectile Weapons.
Projectile weapons were completed by having the resource called scrap that you fire at enemies, we wanted to achieve having multiple weapons to switch from but chose not to implement it due to time constraints.

Minimal Viable Feature 5 – Pixelated Graphics.
Pixelated graphics was achieved by the way of separate sprites for each different mode, the player character had 3 separate states to show when the different states of full on for the player without looking, the graphics were kept to a strict 16-bit style to keep the overhead low.

Extended Viable Feature 1 – Character Speed alters based on ammo/Health.
For this EVF while trying to implement the feature, we found it was too complex to configure properly while in constant conjunction with the player health and ammo, so it was not included in the final build of the game.

Extended Viable Feature 2 – Ammo/Armor Adds Effects.
The ammo and armor effects were added, as the player can collect scrap during the game and add extra health. While at low health the player has a dash ability, while the enemy drops scrap for the player once defeated for the player to pick up and reuse as ammo.

Extended Viable Feature 3 – Sound Effects/Soundtrack.
The sound effects for this game were successfully implemented, all player sounds executed on command, as well as enemy sounds, but automatically activated. Both classes have a jump and attack animation to accompany them.

Extended Viable Feature 4 – HUD/Menu.
The HUD and menu were configured correctly, the menu show after the player is killed so the player can start the game once again, and the HUD was implemented to show the players health.

Scope Creep.

Weeks 1-3
Planned tasks:
- Nothing planned.

Tasks complete:
Week 3 was the week we formed our team, we spent this week discussing what it is we want our project to be and then once we decided on a 2D game, we began planning the tools and programs we will be using to create our game.

Week 4:
Planned tasks:
- Set up our GitHub and Java project.
- Research our assigned fields

Tasks complete:
We spent this week discussing what our game will be about and getting ready by researching what it is we will be implementing and making sure we are attempting a project that is going to be a challenge but can also be finished in the following 8 weeks. In the back half of week 4 we had been setting up all the ‘tools and technology’ that we will be using to create our game, like GitHub and Java.

Week 5:
Planned tasks:
- Creation of the Java classes we will be using and start working on our assigned classes.
- Begin work on graphics.

Tasks complete:
We did make a start on all assigned classes this week and made a start on our game's graphics.

Week 6:
Planned tasks:
- Continue work on our assigned classes.
- Implement more classes like health and projectile.
- Begin working on a basic sandbox level design.

Tasks complete:
Week 6 we continued working on our parts of the project, further developing the classes we were assigned. We also made starts on all other classes that were planned. A sandbox level design was started, it did not include hazards or enemy's as they were still being developed.

Week 7:
Planned tasks:
- Continue all allocated classes.
- Begin assignment 2 work.

Tasks complete:
We did continue our allocated parts of our project this week however progress seemed to slow down a bit around week 7, most likely due to assignment 2 being discussed and broken into individual sections to complete. We did start on assignment 2 this week.

Week 8:
Planned tasks:
- Continue all allocated classes.
- Complete assignment 2.
- Have graphics artefacts ready.
- Begin work on final level design.

Tasks complete:
We did not make too much progress towards our project this week, we did however complete our assignment 2. We did not have graphics ready only early-stage artefacts and we did not start on our final level design either.

Week 9:
Planned tasks:
- Continue all allocated classes.
- Continue work on final level design.
- Begin to work on soundtrack.
- Begin work on a testing document.
- Finalise work on primary classes, Main and Physics.
- Finalise work on secondary class, Projectile.

Tasks complete:
We all continued working on our portions of the project. We did not make any progress towards a final level design, game soundtrack or a testing document. We did have complete the Main class and the Physics class; the Main class would however be altered slightly each week as our project came together. We started on the Projectile class this week.

Week10:
Planned tasks:
- Continue all allocated classes.
- Finalise level design.
- Finalise Sound classes and begin testing sound.
- Finalise graphics.
- Implement graphics into game.
- Finalise work on secondary classes like Player, enemy and GameObject.
- Begin work on Boss class.

Tasks complete:
We did not have a finalised level in week 10 nor had we started work on one. Sound classes were not finalised, but a lot of progress was made on them. Graphics were not complete so implementing them into the game was also not complete. All secondary classes that were scheduled to be complete by week 10 were complete asides from the Enemy class, which was about halfway through its completion. The Boss class did not get started this week.

Week 11:
Planned tasks:
- Continue all allocated classes.
- Begin work on collectable player power-ups.
- Begin designing graphics for power-ups.
- Finalise Graphics, Scrap and Health classes.
- Finalise soundtrack.
- Start assignment 2.

Tasks complete:
Week 11 the work towards our project started to ramp up, we completed most of our games classes and game graphics started to be complete. Level design began and game sounds were being produced as well. The games soundtrack was not complete this week nor was all graphics complete. We did make a start on a speed boost power-up and did not start on graphics for them. We also had not started or complete any testing but did start assignment 2.

Week 12:
Planned tasks:
- Complete all our allocated classes, finalise our game.
- Complete all graphics.
- Complete all sound effects and the games soundtrack.
- Complete assignment 2.
- Finalise testing.

Tasks complete:
Our game was coming together fast in week 12, we had completed all our MVF's and most of our EVF's in time. The final level design was complete (it was not as large as initially planned), we had most of the graphics we wanted in our game complete, the sound effects were complete along with the soundtrack. The end game boss was complete and implemented. We did not complete our collectable player power-up EVF but did have a single power-up implemented. We were not able to extensively test our game but did get a lot of testing done in the final week of development.

Testing.

Player Movement.

Test	Expected Outcome	Actual Outcome	Correct?
Keyboard input is taken from user and applied to the player sprite for movement.	The player sprite will move according to the input of the user. A= move left, D= move right.	The player moves left when A is press and right when D is pressed.	Yes.
Player can “jump” when the spacebar is pressed.	The player sprite will jump into the air then slowly fall back to the ground due to gravity. Jumping should not be permitted in the air.	When spacebar is pressed, the player sprite moves upwards. The player sprite then slowly moves back down. Jumping in the air is not permitted.	Yes.
Player sprite cannot move into platforms or other objects.	Movement should stop once contact with a platform is made.	When player collides with platforms movement is stopped.	Yes.

Projectiles

Test	Expected Outcome	Actual Outcome	Correct?
Mouse input in taken and when mouse is clicked a projectile will fire towards the point the mouse cursor was clicked.	Projectiles will only be created if player has available ammo. Once clicked the projectile will move in a straight line towards the point the mouse was clicked.	Projectiles only created when player has ammo. The projectile moves towards the point the mouse was clicked.	Yes.

Projectiles will turn into scrap once it has collie with another object.	The projectile will follow its path towards the mouse point and if it collides with anything movement will stop, and a scrap will be created in the same spot.	Scrap is created in the same spot that the projectile has collided with another object. The projectile is then deleted.	Yes.

Pixelated Graphics

Test	Expected Outcome	Actual Outcome	Correct?
Pixel graphics are overlayed on the position of the object to create graphics.	A graphical image should be shown in the same position as the object created in the code. For example, the player graphic should overlay exactly where the player object is located in game.	Graphical images are in the locations of the objects.	Yes.
Graphics that no longer need to be shown are removed.	Once an object is deleted the graphical representation of the object should also be removed to represent that to the user.	Graphics are deleted at the same time the object is deleted.	Yes.

Armour Abilities.

Test	Expected Outcome	Actual Outcome	Correct?
Abilities of the player are changed by collecting more scrap and ammo.	The player will be able to gain abilities depending on how much scrap/ammo is picked up.	Scrap/Ammo has no effect on abilities of player.	No. This is not implemented in the current build of the game.

Sound

Test	Expected Outcome	Actual Outcome	Correct?
Sounds are played to reflect the actions of the player.	When the player is shot, jumps, picks up scrap, or shoots an enemy, a sound is played.	Sound effects play at times appropriate to the actions of the player.	Yes.
Sounds are stopped and not looped when they have ended.	When a sound is finished playing it should be stopped and not loop or create a new sound effect.	Sound is stopped at appropriate times.	Yes.

Enemy Movement

Test	Expected Outcome	Actual Outcome	Correct?
Enemy can move around the level.	The enemy should be able to wander around the level nearby where the enemy has spawned.	The enemy moves in a direction that is initially chosen at random, upon collision on world objects the direction is inverted, and the enemy continues to move normally without getting stuck.	Yes.
Enemy can jump.	The enemy should be able to jump.	The enemy can jump decided upon random logic and an interval timer to prevent ‘spamming’ jump behaviour.	Yes.
Enemy can target player.	The enemy should begin to target the player once the player is close enough.	The enemy only targets the player once the player is within a hardcoded range.	Yes

Enemy can attack player.	The enemy should attack the player with multiple attacks, a short-range attack, a medium range attack and a long-range attack.	Functionality not fully implemented, currently the enemy will always perform a 'long-range attack' by shooting projectiles at the player once the player is within a fixed range.	Partially. Some aspects implemented correctly however feature is not fully functional – not all aspects work as intended.
Enemies do not move off-screen unnecessarily.	The enemy should not move off-screen 'randomly.'	Enemies move in a random initial direction and the direction changes when collision happens. Enemies may be able to roam off the map, but this should not be possible due to level design.	Partial. Enemies have the low possibility they may roam off the map, but this has not occurred during testing.
Enemies can move with and without relation to the player.	Enemies should be able to move freely and move relative to the player when within an acceptable range.	Enemies move based on a random initial direction and the direction changes upon collision with platforms.	Partial. Enemies currently always move without relation to the players location.

Side Scroller/Platformer

Test	Expected Outcome	Actual Outcome	Correct?
Camera follows player.	Camera correctly follows the player.	Camera is centred in the middle of the window and the world moves around the player, making it so that the camera follows the player.	Yes.

Platforms can be collided with.	Entities (Players and Enemies) can collide with the platforms.	Collision works as expected. Entities do not pass-through platforms and they can walk on them / against them as expected.	Yes.
World can move left/right and 'scroll' in/out of view.	Camera follows player and can move out of fixed window dimensions.	Camera correctly follows players location and player can move out of the defined window dimensions with the camera still following.	Yes.

Character Speed Alters Based on Ammo/Health

Test	Expected Outcome	Actual Outcome	Correct?
Character speed increases/decreases based on the amount of 'ammo' or 'health' the character has.	The players movement speed should increase or decrease relative to the amount of 'scrap' that the player has.	The players movement speed correctly scales based on the amount of 'scrap' the player has.	Yes.
Scrap is lost when firing projectiles causing the player to increase movement speed.	When the player is shooting scrap, the scrap count is decreased, and the players movement speed is relative to the amount of the scrap the player has remaining.	Player movement speed scales accordingly.	Yes.

HUD/Menu

Test	Expected Outcome	Actual Outcome	Correct?
Menu displays when starting game.	When the game starts the HUD/Menu displays.	When the game is started the HUD is initialized and begins to draw relevant information to the game such as performance metrics and relevant player information.	Yes.
Game can be paused, and menu shown.	The game can be paused and when paused a menu is shown to the player to indicate the paused state.	When the player presses the ‘escape’ key the game pauses all input, and an overlay is shown to indicate the game is in a paused state.	Yes.
Death screen shows when player ‘dies.’	A death screen should be shown once a player has ‘died.’	When the player runs out of scrap by either wasting it or taking damage, they are in a ‘dead’ state and the game ends. Upon the game ending a screen/overlay is shown to indicate this.	Yes.

Tools and Technologies.

Microsoft Teams.

This is our primary method of communication. We use it regularly for communication, and collaborative work on documents required for our project, as well as sharing documents related to our project.

GitHub.

Our team performs regular uploads of our program code to GitHub to collaboratively work on our program. It allows us to dynamically accept and merge work done by our team mates, update others work to our local files, and revert to previous versions should we experience issues with any changes made to our build.

Eclipse IDE.

We have chosen to use Eclipse IDE as the workspace that we will use to program our game in. The Eclipse IDE has built in support for GitHub and since we are using Java for our main programming language, Eclipse is the ideal choice as it will support everything we will be needing.

Trello.

Trello is a task allocation and tracking collaboration tool that our team uses quite significantly. We use it to separate the tasks into our own personal areas, collaboratively track how each other is going, and upload assets and documents so they can be accessed and viewed by one another.

Java SE 16.

We are using the latest Java release as the language to write our program in. It is a powerful, highly flexible language that is easily deployable across multiple platforms.

LWJGL (Light-Weight Java Gaming Library).

We are using the highly complex and powerful LWJGL to assist us in creating our project. It is a library that contains multiple API's geared towards producing games.

The main features we have chosen to use include:

- OpenGL.

OpenGL is a popular library with a lean API that exposes functions allowing programmers to buffer data to the GPU for rendering.

- OpenAL.

An audio library that handles loading audio data from file, storing the data, and playing sounds dynamically in a 3D space.

- NanoVG.

A vectorized rasterization library for 2D graphics. This library is built around OpenGL and exposes an API for programmers to utilize, allowing them to easily create basic geometry shapes such as rectangles, circles, and triangles. This library rasterizes these shapes into vectors which OpenGL will interpret and buffer for the GPU to render.

- NanoSVG.

A library that will parse a supplied SVG coordinate list and rasterize into geometry shapes that are supported by the SVG standard, much like NanoVG this library exposes an easy API for programmers, this library simply parses supplied data, rasterizes and vectorizes it.

- STBVorbis.

STB Vorbis is a small library created by Sean Barrett, also known as “nothings”, it is originally a single-header library for the programming language C and Vorbis is designed to parse the .ogg file and read multiple segments into memory to be used alongside other libraries such as OpenAL.

Photoshop.

We have chosen to use Photoshop to create our graphics for a few reasons, with the primary reason being the popularity of Photoshop and its ease-of-use. When making pixel art Photoshop has a lot of features to ease these processes, things such as using grids to keep art consistent and many other features.

Piskel.

Alongside Photoshop, we have also utilized a free online editor for animated sprites and pixel art, this tool is entirely web based which means no additional downloads for our team members.

Ableton Lite.

Ableton produces a light version of their audio recording, mixing and mastering software – Ableton Lite. This will be the tool that we use to produce all the sounds for the game. It is extremely powerful, even as a light version, and will enable us to create complex, specific audio relevant to the actions in game.

Challenges and Learning.

Corbin.

Group Challenges.

Mostly our group has worked well together. The primary difficulty many of us faced was learning a new Java library that we would use to program our game in. LWJGL has a steep learning curve, especially for beginners to programming. We have spent many hours researching it and are now satisfactorily skilled in its used to achieve our goals.

One of our members uses a Mac, and OpenAL (Open Audio Library) was deprecated on Mac systems a few years ago. This has been unfortunate, with no real solution as there is no longer support for it unless using an obsolete version which is difficult to get now.

Outside of technical issues, our personal lives have gotten in the way at times, with me moving to a new house and changing jobs towards the end of the course, and Connor's house flooding close to the beginning.

How Challenges Were Addressed.

We have worked together to overcome these challenges and have been understanding of the difficulty each other must face at times. It has been important for us to reconsider each member's workload, and to reshuffle tasks accordingly. Open communication creates a sense of confidence in knowing at least where people need help, prioritising tasks based on where they are in their life, and the nature of their interaction with the tools and technology used.

Learning from Challenges.

These challenges provided us, not only with a greater sense of the tools used in our project, but also with a better understanding of how to accommodate one another's personal experience into our planning. We are only as strong as our weakest member, so at times others must pick up the slack to maintain productivity. Our understanding of LWJGL, and programming skills has increased exponentially, and we all come out of this course as more sound Java programmers.

Changes.

We decided to forego some of the original viable features we set out to achieve, specifically the "Scrap adds additional effects" feature, as it was getting much too close to the end of the course for us to include such a complex level of functionality. We also moved some of the graphics requirements from Connor on to Hamilton as he had experience in animations and put his hand up to help.

Project Plan Refinements.

If we had been aware of the challenges that we would face, we may have decided to accomplish a less complex game, one that was based on a premade engine. We also would have spent more time planning the specifics, rather than allowing for team members to make their own calls disparate of the rest of the team.

Timeline Refinements.

We would have spent more time in the beginning working on the fundamentals, rather than expecting to quickly move from one part to another. We could have focussed more closely on key features that needed a lot of work, rather than being forced to move on to the next due to time constraints.

Risks and Unexpected Events.

Mostly the risks and unexpected events portrayed in Assessment 1 went unrealized. We worked together, and mostly had a clear understanding of what was required of us. The only risk mentioned that came real was, at times, communication was lacking from some members leading to them being confused as to their expectation, though the information was readily available to them.

Connor.

Group Challenges.

Some of the group challenges was small things like communicating effectively between the 6 of us, the divide of jobs as most people were doing the coding side of the project. Towards the end of the project, we discovered that Leo had issues with not being able to play the music due to an unforeseen issue with Apple Macs and LWJGL, which is a known user issue, there were small graphics issues I had encountered over time, but I just put those down to rookie errors. Some of the errors I encountered was the sizing of the files, how big the image was, whether the background of the image was erased and see through, having to shift a differing 72 slides for player character and enemy sprites, as generated sprite sheets were harder for the programming side to handle.

How the Challenges Were Addressed.

Starting from the top of the last paragraph and moving downwards, some of the ways we combatted the communication issues, we put a time sheet of sorts in so that each group member can list down times they are not available, but also meeting times can work around those times so it is easier to structure, the issues with Mac and LWJGL is a known problem and isn't fixable yet, so we had another group member do the recording of the project for Part 2 of this assignment. The graphical issues I had were easy to fix, it was just more tiny issues, like having the 16 bit sprites not sized up so that when implemented that can be scaled down and still keep their shape and not distort, I had to change some of the positioning of the player character to lining up properly with the jump animation, or not deleting the background layer to make them transparent, while also missing some slight squares, but those were cleaned up immediately after I was notified about them.

Learning from the Challenges.

Most of my side of this project had been dealing with new skills and experiences involving photoshop, as I do not have any background in Art or Animation, designing characters, character sprite sheets, background, foregrounds. There was a lot on my

plate that I did not realise when too much was too much. In the later weeks, Hamilton jumped on to help with some of the art aspect, that being the background and fore ground for the game. I learnt a lot during the character design phase of this assignment, as learning how to try to move the character for the animations was a little hard to figure out. Configuring the enemy sprites so white boarders were not showing past the black board I had set out for the characters.

Project Plan Refinements.

The scope we had for the project was quite large at first, but that may be due to 5 of the 6 did not have any game development experience. We had a slightly larger scope for the assignment, but over time and figuring out just how much work was head of us, we narrowed the scope of our assignment, leaving mostly just the main core of the product that we had already discussed. It left us with a good framework to get a product out in a short amount of time, with limited knowledge and developing skills along the way.

Timeline Refinements.

The changes I would have made to the timeline, I would have kept it very much the same as to how we had it, I would have added someone else to the design team, so the work could be split a little more evenly, allowing for more time to fix other things wrong with the project, due to the possibility of the design work for the animation side would have been done a little earlier making it easier for the rest of the group. other than that minor adjustment, I would not have changed much in timeline for this project.

Risks and Unexpected events.

A very unexpected event was the flooding of one of our members houses (Connor) that set us back quite a bit as we waited on graphics a little longer than what the group really wanted, the biggest risk I had was along the lines of learning Photoshop by myself, trying to design characters and then create them by myself was hard work, the biggest risk I took there was not really knowing what I was doing week in and week out, I had a rough idea but I’m generally better with someone to bounce ideas off until an idea comes to fruition, or whether the team would accept my low levels skills with no background in the subject

Hamilton.

Group Challenges.

Often after commits to gitHub, team members code will sometimes break after pulling. This happened quiet often to start with but slowed down towards the end of the build once the code started to come together some more. This issue was caused mostly by the LWJGL library we used while creating our game.

Home life caught up with some team members, making it difficult for some of us to finish their allocated portions of the assignments. This became an issue towards the end of the study period when our game was starting to be constructed and all our individual

work was being combined, finalising our game. Some tasks assigned were started but not complete.

Some team members struggled with coding and working with the LWJGL library. Using this library did help in a lot of ways but also made coding difficult at times, specifically because it is difficult to find online resources related to LWJGL.

How Challenges were Addressed.

Ross was most helpful with addressing the issue of people's code breaking after commits, providing video captures and staying back after meetings to help those who had issues fix them. Because of these video captures and walk-through's this issue was resolved around week 9, we all became more familiar with fixing our own code and using the 'build path' option within Eclipse to link the LWJGL library which was the root cause of this issue most of the time.

The issue of work not being complete on time was solved quickly by redistributing and shuffling the workload around. We successfully overcame this issue by taking some coding away from me and giving it to the more capable team member, Ross. I then picked up some of the graphics MVF to ensure we were able to complete our MVF's on time.

The problem of struggling with the library LWJGL (and coding a 2D game in general) for some team members was tough to work around but was done so by sharing what we have found in relation to other people's sections while trying to research our own sections. We also kept up strong weekly communication where those who did have issues to do with code could raise with the others and a solution would be found, or at the very least the team member who is struggling will be pointed in the right direction.

Learning from the Challenges.

Everyone has learnt something new this study period regarding GitHub, especially those of us who have had limited exposure to the world of IT. We decided to link our Java project up to GitHub which allows us to push and pull our progress from Eclipse, something a lot of us did not know about. For those of us who have not used GitHub much previously, this was a good learning curb, highlighting what can be done with GitHub.

Our biggest take away from this as a group was team-oriented thinking, whenever a team member seemed they cannot finish a portion of their work we were able to sort it out quickly, allocating new roles and offering help.

We have all taken something away this study period to do with coding, building this game has been particularly challenging for most of us but at the same time a good opportunity to learn how game development works and how the code behind it is written.

Changes.

The biggest change to our group structure was redistributing workload after team members either could not finish their portion or could not work out how to code their allocated task. This changed the timeline a small bit forcing us to focus more on

the parts that needed it. The timeline for our project always remained the same, this rigid structure was the key to our successful completion of our project.

Project Plan Refinements.

We had a fairly strict schedule which kept most MVF's and EVF's up to date and complete on time, the only thing I would change is to make more deadlines, preferably weekly deadlines where progress needs to be shown. Doing this ensures the project and its MVF's get done on time, (Basically, the complete opposite to working agile).

Timeline Refinements.

The changes to our timeline that I would suggest would be completing the graphics earlier, in particular the platforms. Doing this allows us to start working on the level layout and design earlier. Level building has turned out to be a very lengthy process and it would have been beneficial to have made a start earlier.

Risks and Unexpected Events.

Our group did well to avoid any major risks while completing our project and for the most part avoided the risks we had predicted back in our first assignment. We did have an issue while completing our second assignment, we had not set deadlines creating a 'crunch' period in the last couple of days before we were due to hand in assignment 2. This risk was our first risk raised, 'proper planning.'

Leonard.

Group Challenges.

There were several challenges faced within the groupwork. The main problem I had encountered was specific to my hardware, using a MacBook laptop to develop a java program seemed fine at first yet it turned out that Apple had depreciated support of OpenGL and OpenAL, the external libraries of which we based our entire project on. Another challenge I had was a situation where I could not test my Texture class, nor the other classes could render Textures as for most of the project we had no images to load. Furthermore, meeting times were not always clear as team members would have different schedules quite often and the next meeting time would be confused. Lastly, I also found a challenge in my ability to complete my role without assistance.

How Challenges were Addressed.

Ross and I discussed the MacBook problem, I suggested that I could run a windows OS while Ross took it into his own hands to alter the program to work with an older version of OpenGL which allowed me to continue using my standard OS. After multiple communications to Connor, I had decided to spend a few hours outside my role making temporary player graphics to test my class, while later Hamilton picked up a portion of the graphics development and we re-discussed the project roles. Meetings were sometimes changed and communicated efficiently where most team members could attend or would comment on the meeting afterwards. To continue working in my role I received some assistance from Ross who was able to help an amount that would allow me to continue.

Learning from Challenges.

Communication and mutual respect are huge factors in how teams can work together, I believe the team was able to produce what we have due to this and the overall teamwork. We improved our generic groupwork ability due to these challenges and how we handled them.

Changes.

As a team of newly taught developers, we were unable to be as accurate as necessary in our project goal and had to lower the scope of some deliverables. Due to outside life circumstances effecting team members we changed roles and workloads.

Project Plan Refinements.

I like that we worked in Java and OS trouble would never have been foreseen from our research. That being said, I am not sure if it would have been better, but I would consider testing out a more basic library in Java to develop a game so as to remove the workload from the more skilled programmers and have access to numerous tutorials to make the project fairer and avoid these challenges.

Timeline Refinements.

I would speed up the necessary portions of the programming that are relied upon (my texture class included) by assigning a skilled team member the role of assisting others, to avoid roadblocks in the development process and to stick to our original timeline as I consider our timeline to be quite fair.

Risks and Unexpected Events.

The late creation of graphics resources was not expected and hindered portions of the project while my texture class was a large undertaking that caused rendering of textures to be later than expected. Communication was normally consistent though there were some moments that caused misunderstanding from members not participating in meetings, these situations were mitigated by the team leader or by commenting in MS Teams.

Michael.

Group Challenges.

The team worked well together. There was an initial challenge with experience in creating a game engine. The process of making a game engine takes a long time and because the final game polish (graphics, sounds, etc.) are not in the game it can start to feel like no progress is being made.

Because I work in the healthcare sector my work timetable was very chaotic because of COVID-19 outbreaks. Other member also had changes in their personal lives that made communication and planning especially important.

How Challenges Were Addressed.

Both Ross and I have industry programming experience. At the beginning of the project, we worked together to lay out a design for the game engine that the team

would work toward. We also would be available to talk with team members on any problems they had and help guide them towards a solution.

Due to changes in our personal timetables we had to occasionally change the time for the weekly meetings. The team was flexible and understanding and that helped us overcome these challenges.

Learning from Challenges.

I personally had to learn how to better communicate my thoughts and ideas while trying to explain a concept (such as how a game engine works) to people still learning about the concept. Up to this point I was usually communicating with other people in the game industry so when talking with people outside the gaming industry I found myself skipping or glossing over important parts of the explanation because I had made a wrong assumption that its common knowledge.

The team has grown from the experience of the project as the level of coding knowledge has seen a drastic improvement since the beginning of the project. The team has had discussions about how other coding-based courses are teaching them thing they have been doing in this project, so it gives them a better understanding.

Changes.

The main changes were a couple of the EVF's were cut from the game due to time constraints. The ammo abilities EVF was cut as we ran out of time. Some of the tasks were reallocated to help move the project along.

Project Plan Refinements.

We had a simple plan set out to create the project. All the MVF's were completed but some the EVF's were cut from the game. If we decided to use a pre-made game engine, we could have created a more polished game with a bigger scope but because we created the engine ourselves the game had to be simpler.

Timeline Refinements.

It would have been better to implement graphics earlier than we did. As graphically seeing something on the screen can help make it easier to visualise the final product. This would also allow more time for level designing as level design is long process.

Risks and Unexpected Events.

Most of the risks outlined in our first assignment did not happen. The only risk that happened was a lack of proper planning in assignment 2. Some team members (including myself) had multiple assignments due the same week as assignment 2 so this caused a rushed crunch period.

Ross.

Group Challenges.

I encountered only a few challenges during the project, the first of which was that our meeting times needed to be reshuffled multiple times to accommodate for everyone. The next challenge was GitHub merge conflicts, specifically with the .classpath file, I had not anticipated that this file would cause issues when added to the .gitignore file

and that other members may not have had the exact same build path setup as I did, though I documented how to setup the project, this was still an issue that was encountered. The final issue was that my progression on the project was halted due to others' personal issues, some of my progression was halted due to the delay on graphics being created.

How Challenges were Addressed.

The meeting times were addressed by finding a time where most of us were able to attend, most of the time we found a time where everyone could make it.

GitHub merge conflicts were resolved eventually by adding the .classpath file into the .gitignore file which made it so it would not get committed to the remote repository. I had initially attempted to do this myself, but I also removed the existing .classpath file from the remote repository which caused issues, thus I had to revert the change. It seems sometime later this issue was fixed.

The way we addressed the graphics not being done on time is by assigning two other people to work on them together.

Learning from Challenges.

The group has learnt to adapt better to unexpected situations. I personally have learnt to manage my time better and throughout the course of this project I have learnt more about working as a group and the expectations that come with it.

Changes.

Due to the unexpected circumstances of Connors' situation, we had to assign other people to work on the graphics so that we would proceed further with completing our project. Furthermore, we were a bit ambitious with our MVF and EVF completion estimations, with the experience gained I believe we would better estimate when these features would be completed in future.

Project Plan Refinements.

I would change the deadlines for each MVF and EVF to give more time, we initially set these deadlines under the assumption we would be working on the project from week 2 of our course, however, we started to really pick up pace much later as we overcame the other challenges we had. I would also better assist others with setting up GitHub to avoid the continuous issues we had with the .classpath file changing which affected other people's IDE and required them to fix the build path to run the project.

Timeline Refinements.

I would adjust the deadlines for each MVF and EVF to give more time to complete these features. The deadlines we set for these MVF and EVF's are achievable if we were to adjust sooner and if we had the experience, we gained prior to beginning the project. I do believe if we were to do this same project, or similar, in future, we could keep similar deadlines and have them achievable.

Risks and unexpected events.

I believe we outlined the project risks well in Assessment 1, and some were actualized. I believe that we had some miscommunication when it came to the graphics portion being completed, I believe it was left far too late for the mitigation outlined in Assessment 1 to be useful. I do not believe any of the other risks outlined in Assessment 1 were actualized throughout the lifespan of our project.

Marketing Pitch.

Our name is Code Name Bricks, and we are a group of young individuals studying Building IT Systems together and, as part of our course, creating a 2D game called Trash Unit Response Droid, or T.U.R.D for short. We all have extensive experience in the different required areas of game design, with brilliant team dynamics and an exemplary sense of innovation.

Our game is based on a scrap yard wherein, due to a supernatural event, the trash that occupies the yard has come to life and is forced to battle against the other trash-creatures to survive. The Player controls one of these pieces of scrap and views against other scrap-enemies, attacking with the trash it collects, and collecting the scrap left from their corpses. The characters speed fluctuates depending on the trash it has on hand, forcing the player to be conscious of managing their scrap levels to simultaneously have enough on hand to attack the enemy with and maintain their health, but not so much as to slow themselves down, rendering them unable to avoid enemy attacks. The game involves complex level design comprised of hazards, platforms, drop-pits, enemies, collectable scrap, and a boss fight.

Have you ever wanted to join a game start-up and be part of something creative? If so, contact us to organize a time to meet or to have an informal chat on the phone today!

Skills and Jobs.

Publishing Manager Job Description.

About us.

Code Name Bricks is a new startup company comprising a small team of like-minded individuals that develop medium-fidelity games for the online sector. We are currently nearing the end of the development cycle for our first release and our team needs to grow to push our product to the next stage.

Our game is called T.U.R.D (Trash Unit Response Droid). It is about a piece of scrap in the form of a toilet that, through some paranormal event, has come to life and must battle against other scrap-monsters to achieve dominance. The format is a moody, tense 2D side-scrolling platformer with complex combat features and level design. The game is fun, and addictive to play, with a high reward system.

With the success of our first game, our company will need to grow with it, diversifying into developing multiple games at any point in time, so there is a huge amount of possible career progression for the right individual.

We have a strong belief in giving each team member independence and allowing them to decide how they will spend their time, as long as the work is being done. We provide a happy, engaging and rewarding workplace for our staff and offer great remuneration and benefits packages.

About the Role.

We require a full-time publisher to handle all aspects of publishing our game including hosting on web services like steam, managing user access to the server, pushing updates and downloadable content, and producing a process to include our game on more host servers over its lifeline and as interest increases. Ideally, we would like T.U.R.D to be downloadable on PC, Xbox, and PS4, with new levels, enemies, and features made open for download each month.

About you.

A successful applicant will have:

- Experience in publishing digital media in an online capacity, and preferably with experience in console games.
- Experience pushing updates and downloadable content to live-programs, minimizing downtime and post-update bugs.
- The ability to work independently as part of a small team.
- An innovative attitude, and willing to think freely to expand as the company does.
- The willingness to move into a managerial position as the publishing team expands.
- The right to work in Australia.

Remuneration and Benefits.

We believe in providing quality remuneration packages to our staff to ensure they are satisfied in their position. Our company's benefits are above industry expectations and have high growth potential.

The remuneration is as follows:

- $95,000AUD base salary.
- 16% superannuation contribution.
- Company provided laptop.
- Quarterly KPI bonuses.
- Company social events like concerts, karaoke, Christmas events etc.

Please send your resume, and cover letter to management@cnb.com. Successful applicants will be contacted by our management team within a week of the close of applications.

Code Name Bricks believes in an equal opportunity workplace and is open to a diverse range of applicants. We are committed to providing a workplace that is free of discrimination and prejudice.

Marketing Manager Job Description.

About us.

Code Name Bricks is a new startup company comprising a small team of like-minded individuals that develop medium-fidelity games for the online sector. We are currently nearing the end of the development cycle for our first release and our team needs to grow to push our product to the next stage.

Our game is called T.U.R.D (Trash Unit Response Droid). It is about a piece of scrap in the form of a toilet that, through some paranormal event, has come to life and must battle against other scrap-monsters to achieve dominance. The format is a moody, tense 2D side-scrolling platformer with complex combat features and level design. The game is fun, and addictive to play, with a high reward system.

With the success of our first game, our company will need to grow with it, diversifying into developing multiple games at any point in time, so there is a huge amount of possible career progression for the right individual.

We have a strong belief in giving each team member independence and allowing them to decide how they will spend their time, as long as the work is being done. We provide a happy, engaging and rewarding workplace for our staff and offer great remuneration and benefits packages.

About the Role.

We are looking for someone who is an experienced and driven Marketing Manager to join our rapidly expanding team, overseeing the marketing of our current and future projects. In this role you will:

- Lead our marketing division which will provide a terrific challenge.
- Lead and oversee the delivery of marketing initiatives including campaigns and customer relations initiatives.
- Work within a supportive business that value marketing achievements
- Identify new opportunities from understanding the game industry market and customer trends.
- Work alongside our web development team to increase user retention and product impact.

Successful applicants will have a background in team leadership positions regarding roles in marketing or similar fields, working within roles in marketing games is a bonus though this is not necessary. The correct applicant will be someone who can work both independently and alongside other team leaders to meet deadlines without issue.

About you.

- 5+ years' experience in marketing or similar fields with 2+ years leading a marketing division.
- Tertiary qualifications in marketing or business fields
- High level interpersonal, negotiation, verbal, and written communication skills
- Strong skills in problem-solving and analysis
- Organisational and with a tendency to seek detail.
- Ability to interpret research results and data.
- Strong developed skills in teamwork, leading team agency, and leading team urgency.
- Champion data driven marketing initiative programs created from data insights.
- The right to work in Australia.

Remuneration and benefits

We offer quality packages to our staff here at CNB, we believe a happy work force produces a better final product. A Marketing manager can enjoy:

- $78,000AUD base salary.
- 10% super contribution
- Re-location bonus, (up to $4000)
- Quarterly KPI bonuses
- Paid vacation and sick days
- Full medical and dental package
- Use of company car.
- Use of company mobile phone.
- Company social events, paid for.
- Opportunity for growth and income increase within company.

Please send your resume, and cover letter to management@cnb.com. Successful applicants will be contacted by our management team within a week of the close of applications.

Code Name Bricks believes in an equal opportunity workplace and is open to a diverse range of applicants. We are committed to providing a workplace that is free of discrimination and prejudice.

Website Developer Job Description.

About us.

Code Name Bricks is a new startup company comprising a small team of like-minded individuals that develop medium-fidelity games for the online sector. We are currently nearing the end of the development cycle for our first release and our team needs to grow to push our product to the next stage.

Our game is called T.U.R.D (Trash Unit Response Droid). It is about a piece of scrap in the form of a toilet that, through some paranormal event, has come to life and must battle against other scrap-monsters to achieve dominance. The format is a moody, tense 2D side-scrolling platformer with complex combat features and level design. The game is fun, and addictive to play, with a high reward system.

With the success of our first game, our company will need to grow with it, diversifying into developing multiple games at any point in time, so there is a huge amount of possible career progression for the right individual.

We have a strong belief in giving each team member independence and allowing them to decide how they will spend their time, as long as the work is being done. We provide a happy, engaging and rewarding workplace for our staff and offer great remuneration and benefits packages.

About the role.

We are looking for someone who is proficient in website design and website content creation to be a part of our rapidly expanding team. In this role you will,

- Design and implement our backend and frontend features.
- Moderate and provide support for our website, both front and backend.
- Maintain and develop existing software.

A successful applicant will have extensive knowledge in website design and website content creation, having a background in sales and marketing is a bonus but not a necessity. The correct applicant will be able to take initiative, work independently and meet deadlines.

About you.

To be successful in your application, the following requirements are a must:

- Vast prior experience in website development.
- Experience in website content creation and design.
- Extensive knowledge in all relevant website development languages (HTML, CSS, JS, etc.)
- A background in the video game industry.
- Ability to follow a schedule and meet deadlines.
- Good written grammar.
- Strong communication skills.
- The right to work in Australia.

Remuneration and benefits.

We offer quality packages to our staff here at CNB, we believe a happy work force produces a better final product. A website developer can enjoy:

- $92,000AUD base salary.
- 14% super contribution.
- Company provided laptop.
- Re-locating bonus, (up to $10,000).
- Quarterly KPI bonuses.
- Company social events, paid for.
- Opportunities for growth within the company.

Please send your resume, and cover letter to management@cnb.com. Successful applicants will be contacted by our management team within a week of the close of applications.

Code Name Bricks believes in an equal opportunity workplace and is open to a diverse range of applicants. We are committed to providing a workplace that is free of discrimination and prejudice.

Art Assistant Job Description.

About us.

Code Name Bricks is a new startup company comprising a small team of like-minded individuals that develop medium-fidelity games for the online sector. We are currently nearing the end of the development cycle for our first release and our team needs to grow to push our product to the next stage.

Our game is called T.U.R.D (Trash Unit Response Droid). It is about a piece of scrap in the form of a toilet that, through some paranormal event, has come to life and must battle against other scrap-monsters to achieve dominance. The format is a moody, tense 2D side-scrolling platformer with complex combat features and level design. The game is fun, and addictive to play, with a high reward system.

With the success of our first game, our company will need to grow with it, diversifying into developing multiple games at any point in time, so there is a huge amount of possible career progression for the right individual.

We have a strong belief in giving each team member independence and allowing them to decide how they will spend their time, as long as the work is being done. We provide a happy, engaging and rewarding workplace for our staff and offer great remuneration and benefits packages.

About the Role.

We require an Art assistant, or technical Artist, that would act as the bridge between most of the Team, between various artists for the game, sound engineers to help with successful implementation of the graphics to match with the sounds, the software engineers and game designers to make sure the vision to coming towards the aforementioned goal.

It would be preferred by Code Name Bricks that you would have a sense of technical expertise, prior experience in Teamwork and working effectively within a team of people with different jobs. Leadership and management techniques would be welcome, but not highly sought after at this moment, this job will also require a lot of innovative thinking as we are making a game from the ground up.

About you.

- A steady knowledge of composition work, where you will but working within the team to gain an understanding of the design of the games, the needed video assets that will be detrimental to the work and the way the assets co-exist and work with the proprietary software.
- Video Graphic management, keeping a version control by understand where the team is up to at that current period, access control – which will be releasing the

final version of a graphic to the team, the quality control for all assets and optimisation.
- One of your other duties will be around trouble shooting technical video graphic issues that the team may come upon during the workload that will need to be rectified before continuation of the project.
- Your final task within the team will be the pipeline process management, you will be overseeing the optimisation, administration, and the support for anything graphically for the project.
- You will need excellent understanding of Photoshop and other video editing software (Photoshop, Premiere, After Effects and Final Cut Pro)
- Strong Multi-tasking skills, you will be moving between projects are various stages of development.
- Strong knowledge and experience with version control software.
- Good problem-solving skills to analyse issues and find solutions and workarounds to issues that are not documented.
- Bonus: LWJGL Experience
- Strong Communication skills as you will be the bridge between the video graphics, software, and design.

Remuneration and Benefits.

We believe in providing quality remuneration packages to our staff to ensure they are satisfied in their position. Our company's benefits are above industry expectations and have high growth potential.

The remuneration is as follows:

- $58,000AUD base salary.
- 13.5% superannuation contribution.
- Company provided laptop.
- Commission per image bonus.
- Paid company events.

Please send your resume, and cover letter to management@cnb.com. Successful applicants will be contacted by our management team within a week of the close of applications.

Code Name Bricks believes in an equal opportunity workplace and is open to a diverse range of applicants. We are committed to providing a workplace that is free of discrimination and prejudice.

Appendix.

References.

- COG Branding, *Website Developer*, Seek, viewed May 19th, 2021, <https://www.seek.com.au/job/52351241?type=standard#searchRequestToken=cd3058d6-c9e1-4685-b71c-83af66b90156>
- Private Advertiser, *Website Developer*, Seek, viewed May 19th, 2021, <https://www.seek.com.au/job/52275704?type=standard#searchRequestToken=cd3058d6-c9e1-4685-b71c-83af66b90156>
- RELX, *Digital Manager*, Seek, viewed May 18th, 2021, <https://www.seek.com.au/job/52285462?type=promoted#searchRequestToken=d0769e6a-4e86-4d74-9bb0-3a5f0ca8dc9f>
- Preacta Recruitment, *Digital Product Manager*, Seek, viewed May 20th, 2021, <https://www.seek.com.au/job/52296778?type=standard#searchRequestToken=d0769e6a-4e86-4d74-9bb0-3a5f0ca8dc9f>
- IGT Australia, *Game Artist (Multiple Roles)*, Seek, viewed May 20th, 2021, <https://www.seek.com.au/job/52150466?type=standard#searchRequestToken=18e11f24-9d69-46b1-8063-6324d5c8d0bf>
- 200Pixels Creative Agency, *3D Artist*, Seek, viewed May 20th, 2021, <https://www.seek.com.au/job/52207656?type=standard#searchRequestToken=18e11f24-9d69-46b1-8063-6324d5c8d0bf>
- Mensatech, *Website Developer & Designer*, Seek, viewed may 19th, 2021, <https://www.seek.com.au/job/52267630?type=promoted#searchRequestToken=a19817c9-e982-4b11-875e-ca88b78203c2>
- Zimple, *Website/Digital Developer*, Seek, Seek, viewed 20th May, 2021, <https://www.seek.com.au/job/52109182?type=promoted#searchRequestToken=a19817c9-e982-4b11-875e-ca88b78203c2>
- Test Bytes, *Different Types of Game Testing Techniques*, Test Bytes, viewed May 15th, 2021, <https://www.testbytes.net/blog/types-of-game-testing/>
- Software Testing Help, *Types Of Software Testing: Different Testing Types With Details*, Software Testing Help, viewed May 15th, 2021, <https://www.softwaretestinghelp.com/types-of-software-testing/>