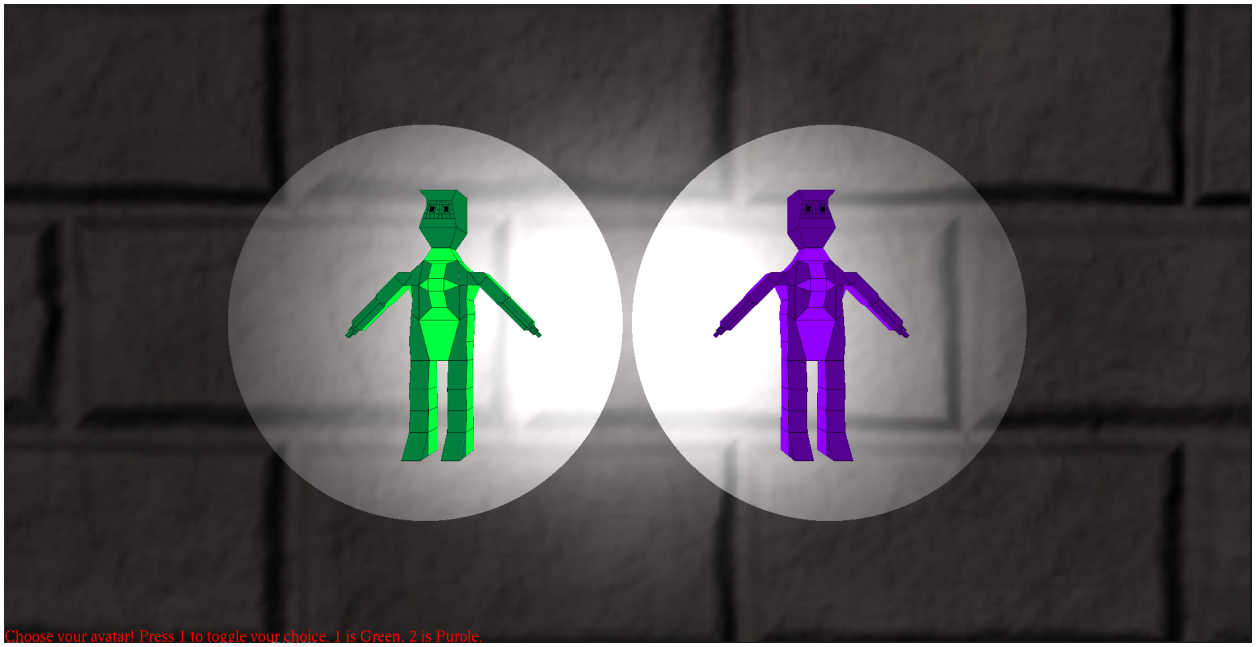
Coin Collector

Brandon Doney and Peter Metz
Section 01 and Section 01 respectively



3.) Instructions for compiling:

1. For the server:
    a. Run the compile.bat file.
    b. Run the run.bat file.
2. For the client
    a. Run the compile.bat file.
    b. Run the run.bat file.

4. Any special device requirements, such as particular input device(s)

No special device requirements.

5. How to play your game, including what things happen and how the scoring works

To play the game, you first select a character. You are then moved to a lobby. In this lobby, you can follow terrain and bounce around a small room as you are unable to leave. When you are ready, you can move outside to there the collecting begins. Move forward to collect coins and increase your coin count.

6. What player controls are available (what keyboard/gamepad buttons do, etc.)

1. Keyboard
    a. grave/tilde key - move from character select to lobby, move from lobby to outside
    b. 1 - toggle selection of character
    c. Space - toggle lights in character select
    d. W - move forward
    e. A - move left
    f. S - move right

    g. D - move backward
    h. V - toggle visibility of world axes
2. Gamepad
    a. Y Axis (left thumbstick) - move forward and backward
    b. X Axis (right thumbstick - move left and right
    c. A (button _0) - toggle avatar select
    d. B (button _1) - move from character select to lobby, move from lobby to outside
    e. X (button _2) - toggle lights in character select

7. A list of initializations done in the scripting

1. World axes
2. One light source
3. The player avatar's translation and scale
4. The "menu" avatars' translation and scale

8. A brief summary of any changes (or none) that you made to the network protocol

No changes made to the networking protocol.

9. A list of changes and additions that you made to TAGE

Added a method in the PhysicsObject class to get the rigid body of a PhysicsObject.

10. A statement indicating the (1) genre, (2) theme, (3) dimensionality, and (4) activities utilized in your game (see week 1 notes [chapter 00] for examples)

This is a (1) racing game, (2) with a sci-fi theme as the player is a flying robot, (3) in 3D space, as the player can move in all 3 axes, taking place on the ground, (4) involving exploration and collection**.**

11. An explanation of where (in the game, not the code) each project requirement is satisfied

1. Input controls are satisfied in all stages of the game
2. Networking works but is not fully functional/stable. Players can see each other and their movement.
3. There are two models in the game (menu, behind the player when outside). They each have custom textures (uv-unwrapped from Blender).
4. The game works fine in singleplayer
5. Avatar selection is at the beginning of the game
6. Scripting is used to initialize some objects
7. The skybox is visible when outside
8. Terrain is visible in the lobby, as well as the terrain-following
9. Lighting is visible on character selection (2 spotlights) and in the lobby (1 positional). Spotlights can be turned on and off.
10. Background music plays constantly, there is a sound effect when selecting a character, and the NPC makes a noise when near it.
11. The bottom left of the screen has various text depending on stage of the game
12. The player has a miniature coin above them when they collect a coin
13. The player has animations, and they are in the game
14. There is one NPC outside, behind where the player is moved to (must move backward to see it)
15. No AI for NPCs.

16. Player is controlled by physics, and can collide with NPC. Everything is affected by gravity.
17. JavaDocs are updated. Code has comments where it is confusing, otherwise variable/method/class names should be sufficient.
18.

12. A list of the requirements that you weren't able to get working

1. Networking is not fully working. A client will sometimes crash. Most things are not synced between clients. A client will not correctly see another client's chosen character.
2. There is no AI for the NPCs.
3. Could have done more with scripting.
4. Could have done more with physics.

13. Any technique you used in your game that goes beyond the requirements

Nothing notable.

14.The contributions of each team member, including who designed which model(s)

1. Brandon
    a. Designed the original creature and texture (later fixed by Peter due to errors with the model's faces)
    b. Added audio to the game (later replaced by music)
    c. Worked on AI, but neither Brandon/Peter could get it working
    d. Added a bush (not present in final version)
    e. Added initial physics (professor's example)
2. Peter
    a. Designed the player avatar. Textured all versions of player avatar. Fixed creature's design and made a new model/texture.
    b. Fixed most of the compilation issues that we ran into. Fixed audio issues we ran into.
    c. Created animations for the player avatar.
    d. Applied physics system to the player.
    e. Made sure networked movement was applied to the player when moved with physics.
    f. Created character select, lobby, and racing stages of the game.
    g. Created coin model and texture.

15. A list of items that you created yourself (models, textures, heightmap, etc.)

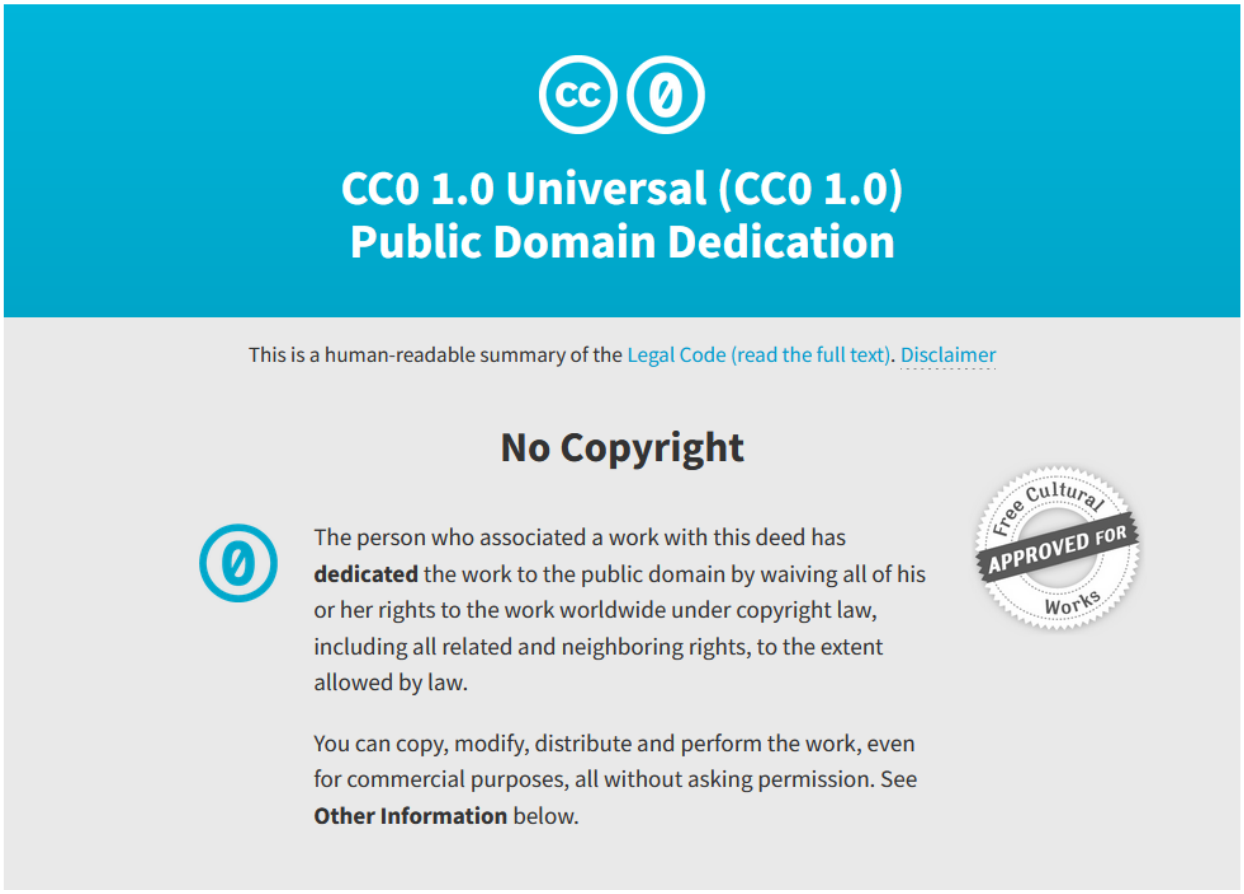
1. Player model
2. NPC (creature model)
3. Textures for the player and creature
4. All heightmaps
5. All animations
6. All object models (excluding default TAGE models)
7. All scripts
8. All textures involving the player (simpleChar) or creature.

16. Evidence of permission to use any item (models, textures, etc.) not listed in #15

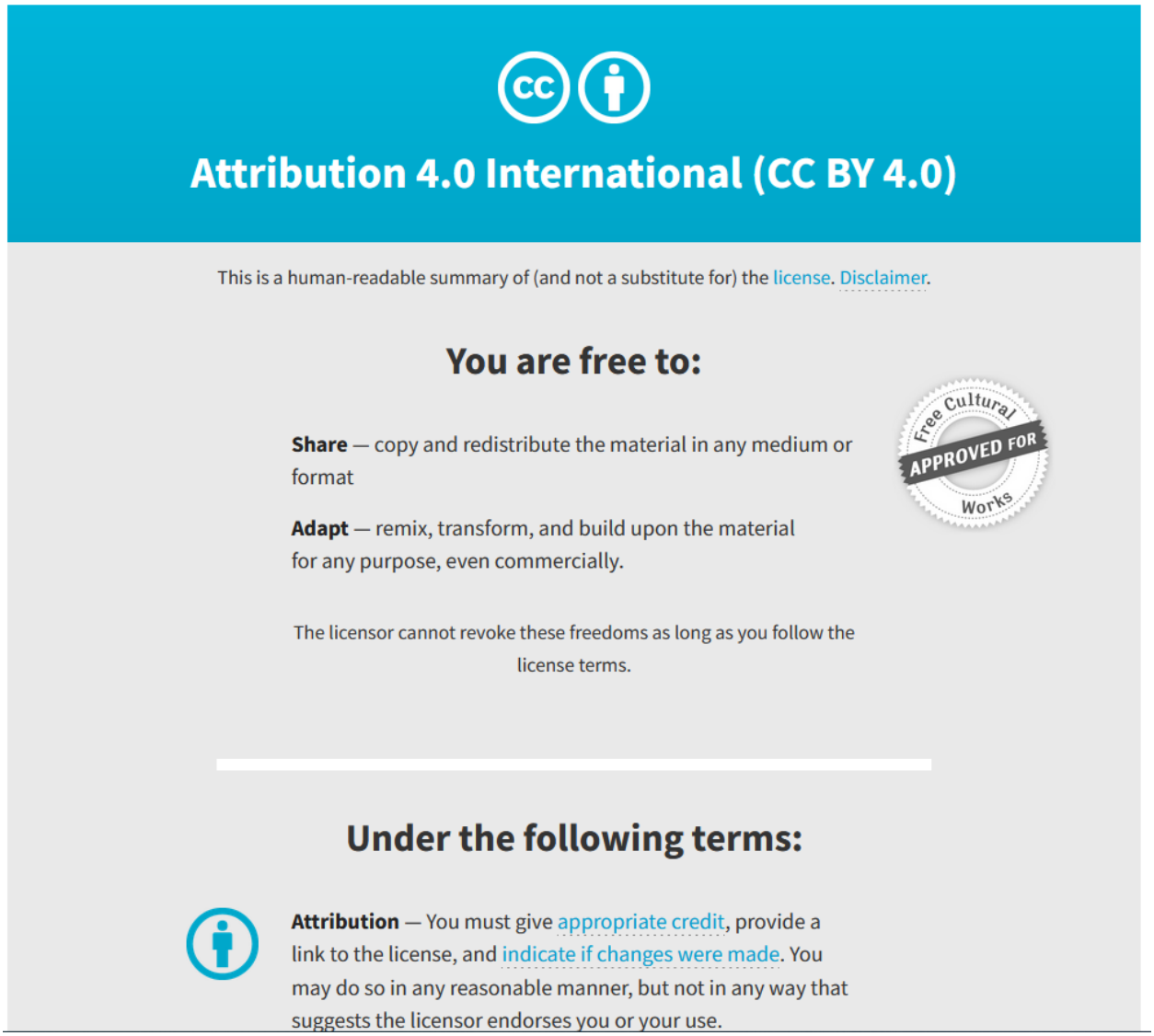
1. Grass texture - public domain, author is athile, opengameart.org/content/seamless-grass-texture-ii. "You can copy, modify, distribute

and perform the work, even for commercial purposes, all without asking permission." creativecommons.org/publicdomain/zero/1.0/

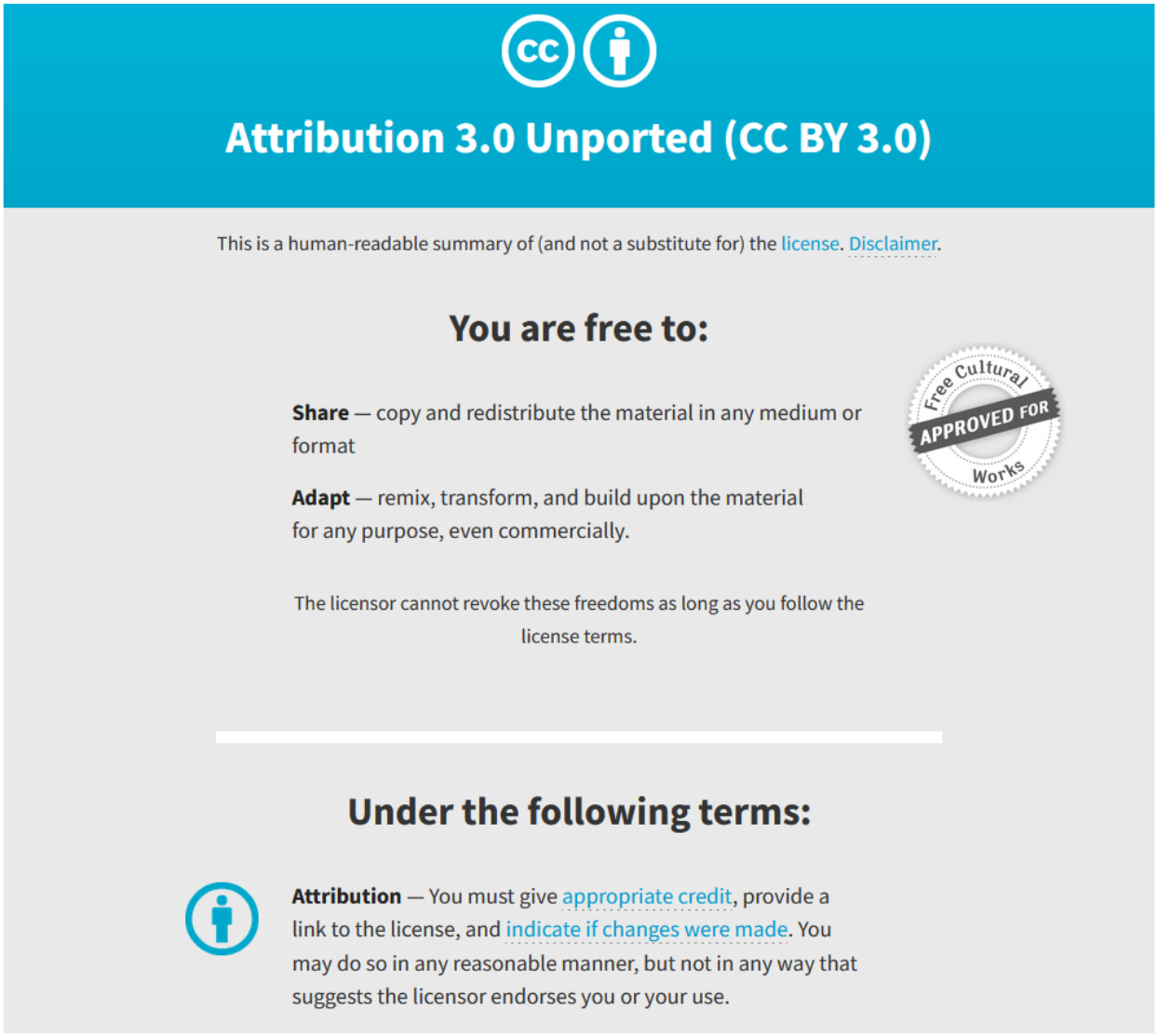
2. Night Sky skybox - public domain, author is StumpyStrust, opengameart.org/content/space-skyboxes-0. "You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission." creativecommons.org/publicdomain/zero/1.0/



3. Signal Pursuit (background music) - attribution, author is Vitalezzz, opengameart.org/content/signal-pursuit. “You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.” creativecommons.org/licenses/by/4.0/. Modified to be 16-bit Mono .wav format.



4. Goblin1 Sound Effect - attribution, author is Little Robot Sound Factory (www.littlerobotsoundfactory.com). “You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.” Modified to be 16-bit Mono .wav format. creativecommons.org/licenses/by/3.0/.



Attribution 3.0 Unported (CC BY 3.0)

This is a human-readable summary of (and not a substitute for) the license. Disclaimer.

You are free to:

Share — copy and redistribute the material in any medium or format

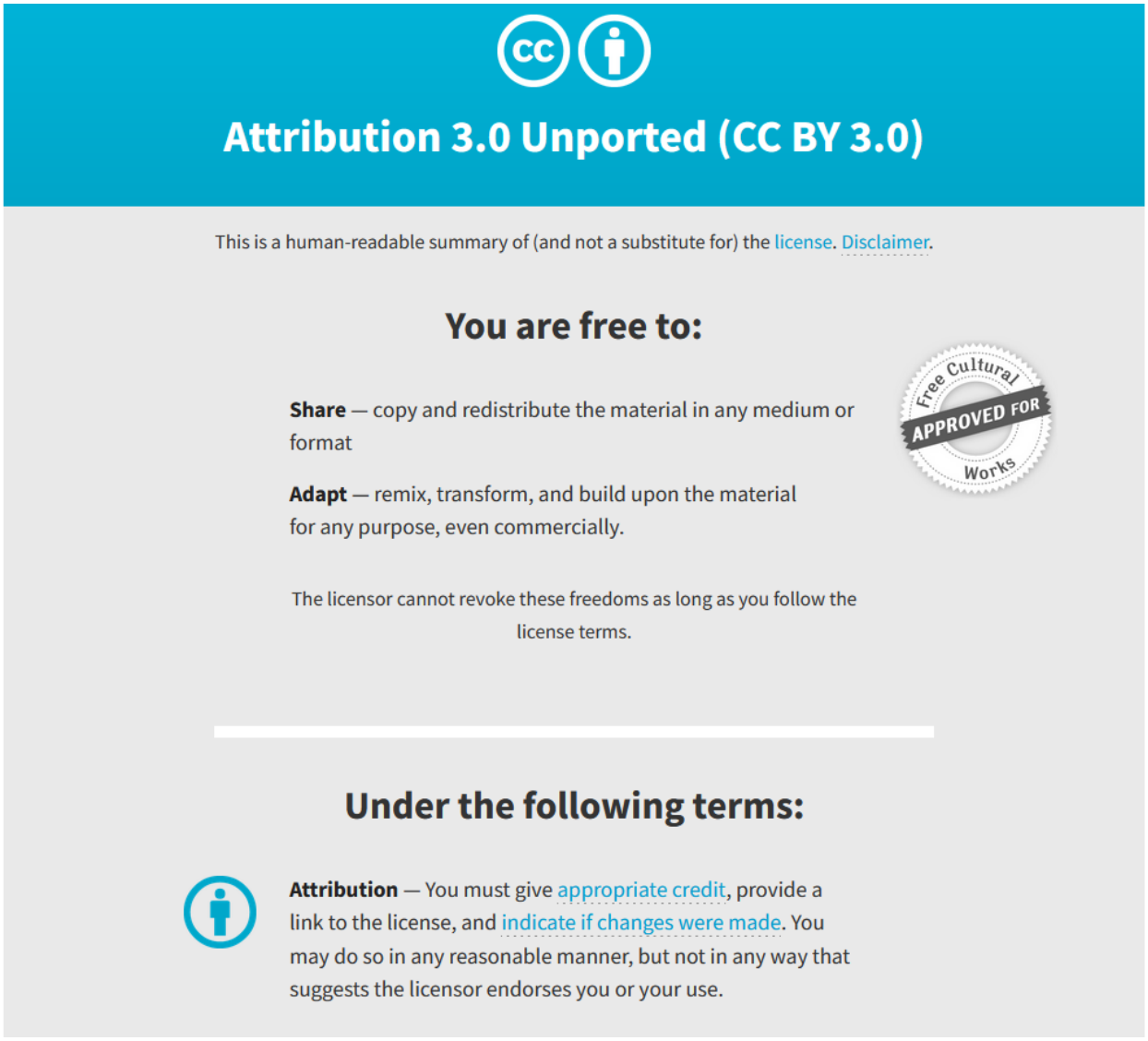
Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

5. MENU_Pick - attribution, "Some of the sounds in this project were created by David McKee (ViRiX) soundcloud.com/virix." opengameart.org/content/ui-sound-effects-pack. You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use." Modified to be 16-bit Mono .wav format. creativecommons.org/licenses/by/3.0/



6. Items not listed in 15 and 16 were provided by TAGE or by the specific code-example download (like the brick texture, dolphin textures, earth).

17. Which RVR-5029 lab machines (at least two – it's networked!) on which your program was tested and is known to work correctly on.

1. Tested on SPACEQUEST and MARIO.