# W26 Mid-Semester Exam - Answer Packet

| Topic | Score |
|---|---|
| Game Engine Landscape, History & Misc | / 30 |
| C++ and IDE Pragmatics | / 20 |
| Lua and Composition | / 20 |
| Engine Architecture and Lifecycle Functions | / 20 |
| **Total** | / 90 |
| **Total (canvas normalized)** | / 250 |

A harder exam than previous years. The average ended up being 78%-- lower than the typical 85%+. A 20 point scale was applied to account for this. -AY

Alright agent– **focus**. You've totally got this. No big! This mission is all you... but I'll lend a hand when I can.

As the years go by, it becomes harder to make novel questions from a limited set of course topics. This year, I accidentally leaned into "depth" to achieve novelty, but this increased the difficulty in a way I didn't intend. A new approach will be needed to achieve novel questions for next year's exam. -AY

**Good evening. Before we begin, please repeat after me.**
"I have neither given nor received unauthorized aid on this examination, nor have I concealed any violations of the honor code."

name / signature _____

uniqname _____

date _____

# Game Engine Landscape, History & Misc

1

__/1

Off-the-shelf game engines were a rarity in the 80s and 90s, forcing most development teams to build their own engines from scratch. Off-the-shelf engines still existed in this era however. Which of the following was widely available (and accessible without fees / paid licenses) first? **(circle one)**

Unreal Engine  Godot  Unity  UbiArt  Game Maker

2

__/1

In 2026, two off-the-shelf game engines form a plainly-visible duopoly, consuming much of the game development and visualization markets. Which open-source engine is closest to disrupting them? **(circle one)**

Game Maker  Unreal  PyGame  Godot  Unity

3

__/1

The 3D affine transformation matrix is a _____ matrix (rows x columns) **(circle one below)**

2x2  3x3  4x4  5x5

4

__/1

What does the acronym "ECS" stand for?

Entity Component System

5

__/1

How does "ECS" differ from the architecture adopted by our engines throughout homeworks 7 and 8? (briefly mention one substantial difference)

In ECS, components do not contain logic– they are simply small containers of data optimized for the systems that need them. In our engines, components may contain functions and data.

6

__/1

How may we temporarily pause execution of a lua coroutine? **(circle one)**

Pause  Suspect  Yield  Delay  NoOp

7

__/8

**The 2D affine transformation matrix, along with its 3D sibling, are some of the most fundamental matrices in all of computer graphics and game development, providing an efficient representation of common movement, rotation, and scale operations. It is a mainstay of popular engines such as Unity, Unr**

Note : -Sin(θ) for top-center and Sin(θ) for left-center were also accepted due to an error in a lecture slide.

Write the 2D affine
cells to the right, supporting translation (Tx, Ty), Rotation (θ), and scale (Sx, Sy).
Fill every empty cell with something.

Pssst!
Top-left is `SxCos(θ)`.
Center is `SyCos(θ)`.

| | | |
|---|---|---|
| SxCos(θ) | -SySin(θ) | Tx |
| SxSin(θ) | SyCos(θ) | Ty |
| 0 | 0 | 1 |

Use the completed 2D affine transformation matrix above to create a transformation matrix that will translate actors by an (x,y) of (2, -1) while also scaling by a factor of positive 3 in both dimensions.

T =

| | | |
|---|---|---|
| 3 | 0 | 2 |
| 0 | 3 | -1 |
| 0 | 0 | 1 |

Use the transformation matrix "**T**" above to transform the following three position vectors.

$$T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} \qquad T \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \\ 1 \end{bmatrix} \qquad T \begin{bmatrix} -2 \\ -3 \\ 1 \end{bmatrix} = \begin{bmatrix} -4 \\ -10 \\ 1 \end{bmatrix}$$

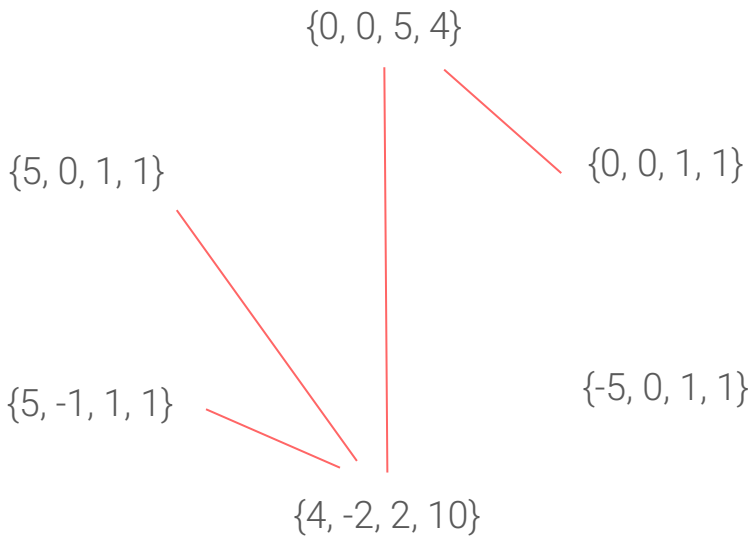
8

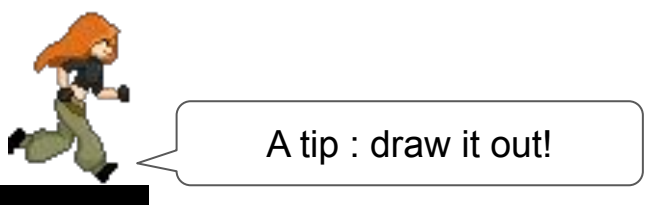
__/ 8

In pseudocode, write the AABB algorithm in the box below. Your algorithm is supplied with an SDL_Rect **a**, and a second SDL_Rect **b**.
SDL_Rect has **{x, y, w, h}**.
If a and b share a corner or border, this is **not considered colliding.**

```
if (a.x + a.w > b.x && a.x < b.x + b.w)
{

        If (a.y + a.h > b.y && a.y < b.y + b.h)
        {
                // It's a collision!
        }
}
```

Utilize the AABB algorithm to determine which of the following rectangles are colliding with which other rectangles. If two rectangles are colliding, draw a line between them. All rectangles are defined in SDL_Rect format {x,y,w,h} and the coordinate system is that of SDL. Multiple rectangles that share borders or corners are not considered to be colliding. The anchor / pivot point for each rectangle is its top-left corner.



# Game Engine Landscape, History & Misc

9

___ / 8

**Match each term below with its single most-reasonable definition at the bottom of the page.** Do so by drawing a line between two boxes. Each box may only be connected to one line. There will be eight lines total.

| | |
|---|---|
| In-House Engine | Zenilib |
| In-House Engine | Adobe Flash |
| In-House Engine  | Retro Engine |
| In-House Engine | Crystal Tools |
| Off-The-Shelf Engine | Clickteam Fusion |
| Off-The-Shelf Engine | Frostbite |
| Off-The-Shelf Engine | Stepmania |
| Off-The-Shelf Engine | UbiArt |

10

___ / 8

**Order the following operations from lowest latency (1) to highest latency (8) according to course lectures, demonstrations, and homeworks.** Write the number / ranking in the empty boxes to the right of the operation.

| | |
|---|---|
| SSD random read | 4 |
| L2 cache reference | 2 |
| Deliver letter via United States Postal Service | 8 |
| L1 cache reference | 1 |
| Local network round trip | 5 |
| Global network round trip | 6 |
| Deliver letter yourself within city (driving) | 7 |
| Main memory reference | 3 |

11

__/2

**`std::shared_ptr`** is popular among developers for several reasons, but receives criticism for having slightly-inferior performance characteristics compared to that of a **`raw pointer`** or **`std::unique_ptr`**.

In the box below, briefly describe what causes this reduction in performance.

std::shared_ptr must store and update a reference counter.

12

__/4

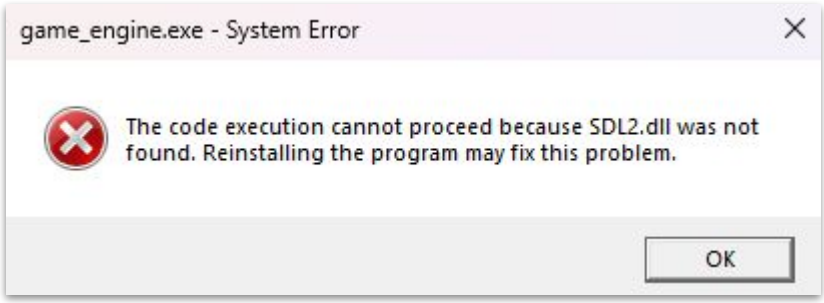
According to the course, what are the four primary "questions" of c++ compilation? (ie, what four pieces of information must be specified to build).

| |
|---|
| Which headers do you want to use? |
| Where should the preprocessor / compiler look for headers? |
| Which libraries / binaries do you want to link? |
| Where should the linker look for libraries / binaries? |

13

__/2

Having provided your compiler and linker with all of the information necessary to build (the four pieces of information listed above), your build completes and your program finally begins to run. Suddenly, your program crashes with the following error--




Note from AY :
I forgot to make problem #13 ask for something, so everyone got 2 points.
Lucky number 13 lol.

14

__/2

This cannot be a compiler issue, as the compiler's job has long since finished. Nor can it be the linker's problem, as the linker successfully created the final binary. Name the technology responsible for this post-build-process error.

Dynamic Linker / Operating System

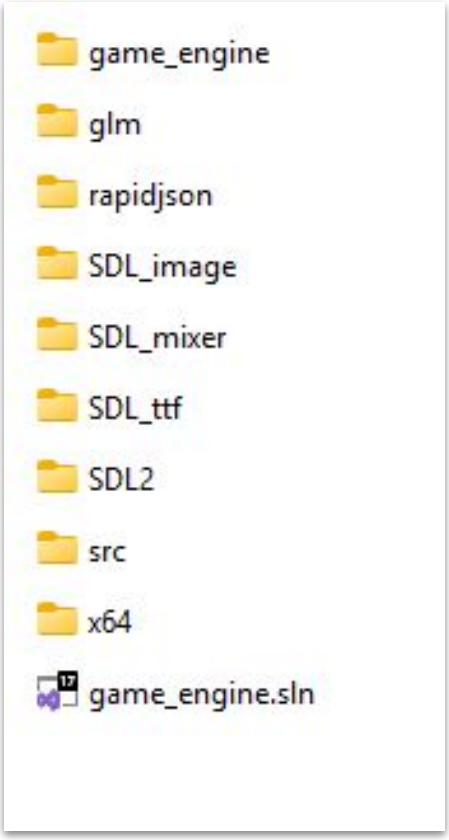
There are multiple correct answers

15

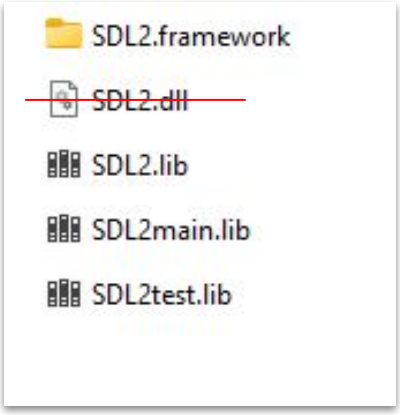
Examine the following file structure and fix the post-build-process issue seen previously. Move a file by crossing it out and writing its name elsewhere. You need only solve the issue seen in the previous problem.

___ / 2

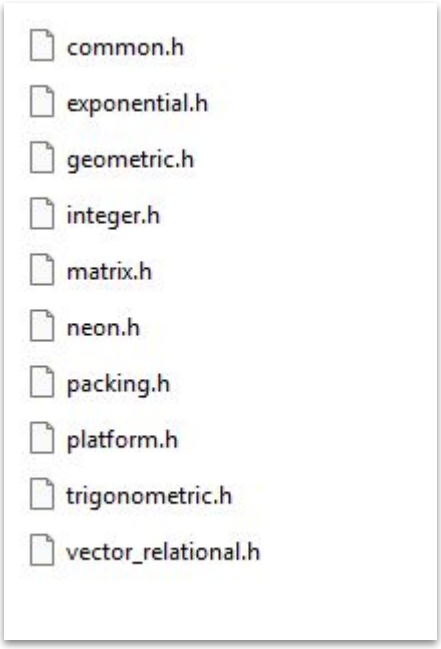
*project/*



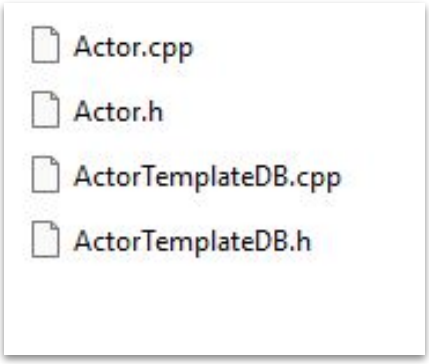

*project/SDL2/lib/*




*project/glm/simd/*

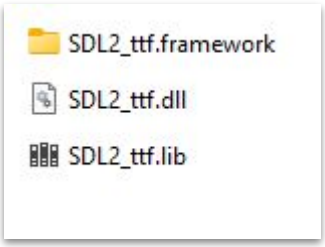



*project/src/*




*project/x64*


*project/SDL_ttf/lib/*




*project/x64/Release/*

# Lua and Composition

16 __/10

resources/component_types/KPManager.lua

```
KPManager = {

	lives = 3,
	intro_frames = 180,
	gameover = false,
	outro_frames = 180,

	OnStart = function(self)
		Audio.Play("intro", 5, false)
	end,

	OnLateUpdate = function(self)

		-- Intro
		self.intro_frames = self.intro_frames - 1
		if self.intro_frames > 0 then
			Image.DrawUI("intro", 0, 0)
			return
		end

		-- Respawning
		local kp_actor = Actor.Find("kp")
		if kp_actor == nil then
			if self.lives > 0 then

				local new_kp = Actor.Instantiate("kp")
				local new_kp_t = new_kp:GetComponent("Transform")
				new_kp_t.x = 3
				new_kp_t.y = 4
				self.lives = self.lives - 1

			elseif self.gameover == false then

				self.gameover = true
				Audio.Play("gameover", 5, false)
			end
		end

		-- Health
		if kp_actor ~= nil then
			local status = kp_actor:GetComponent("status")
			for i=0,status.hp,1 do
				Image.DrawUI("bar", 50+50*i, 50)
			end
			Image.DrawUI("" .. status.hp, 0, 0)
			Text.Draw("x" .. self.lives, 75, 75)
		end

		-- Outro
		if self.gameover == true then
			Image.DrawUI("gameover", 0, 0)
			self.outro_frames = self.outro_frames - 1
			if self.outro_frames <= 0 then
				Application.Quit()
			end
		end
	end
```

# Lua and Composition

```
-- More space for KPManager

}
```

# Lua and Composition

17 __/10

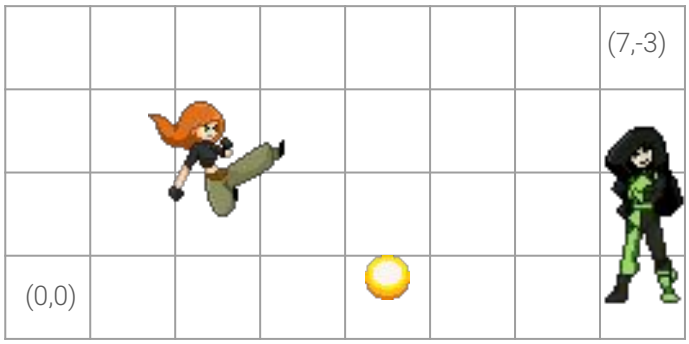
resources/component_types/GlobalHighScoreManager.lua

```
GlobalHighScoreManager = {

	score_objects = {},
	high_score = -1,

	OnStart = function(self)
		Event.Subscribe("global_score_received", self, self.OnGlobalScoreReceived)
	end,

	OnGlobalScoreReceived = function(self, e)

		-- Track new score for this player if new or improved.
		if self.score_objects[e.player_name] == nil then

			self.score_object[e.player_name] = e

		elseif e.score > self.score_objects[e.player_name].score

			self.score_objects[e.player_name] = e

		end

		-- Is new score highest ever?
		if e.score > self.high_score  then
			self.high_score = e.score
			Event.Publish("global_new_highscore", e)
		end
	end,

	OnLateUpdate = function(self)

		local sorted_scores = table.supersort(self.score_objects, function(a,b)
			return a.score > b.score
		end)

		for i,e in ipairs(sorted_scores) do
			Text.Draw("" .. i, 50, 50 + i * 50)
			Text.Draw("" .. e.score, 100, 50 + i * 50)
			Text.Draw(e.player_name, 200, 50 + i * 50)
			Image.DrawUI(e.character, 300, 50 + i * 50)
		end
	end
```

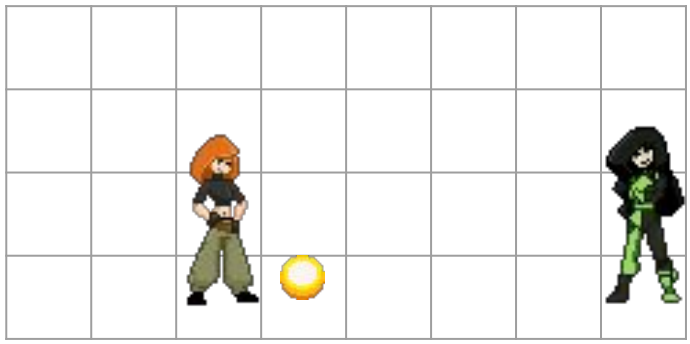
# Lua and Composition

```
-- More space for GlobalHighScoreManager

}
```

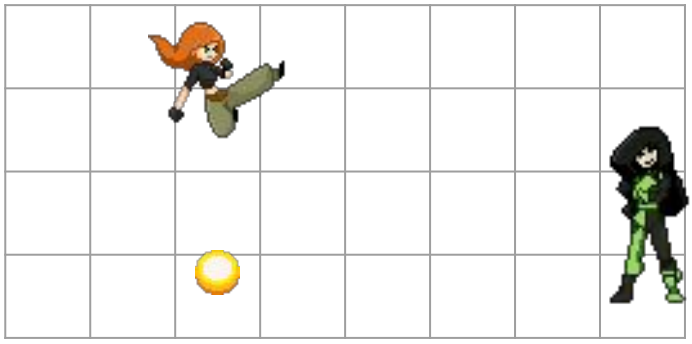
18 __/ 9



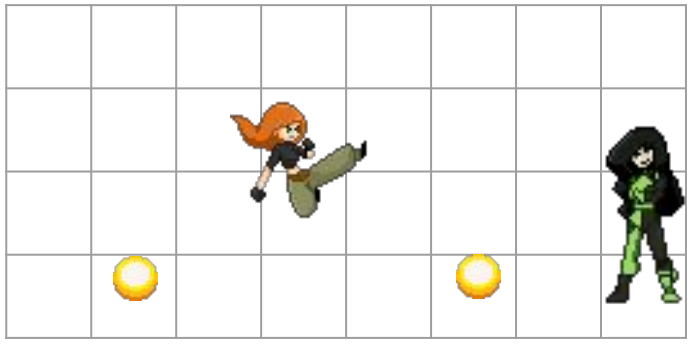

Frame #2 (input coming early frame : **right**)



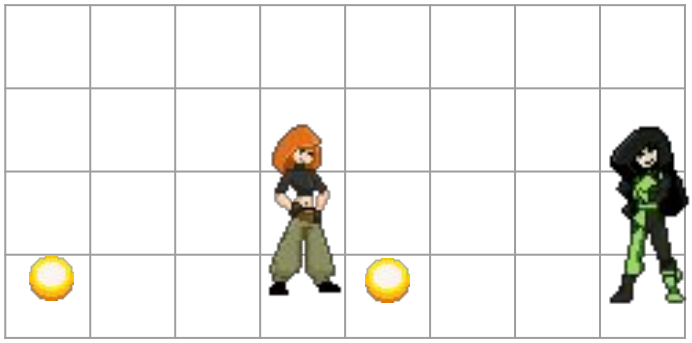
Frame #3 (input coming early frame : **none**)



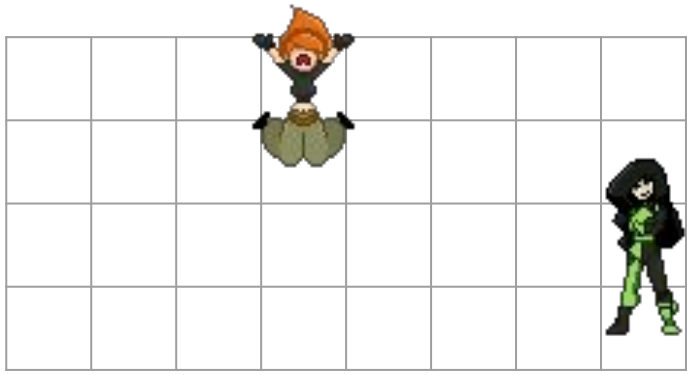
Frame #4 (input coming early frame : **space**)



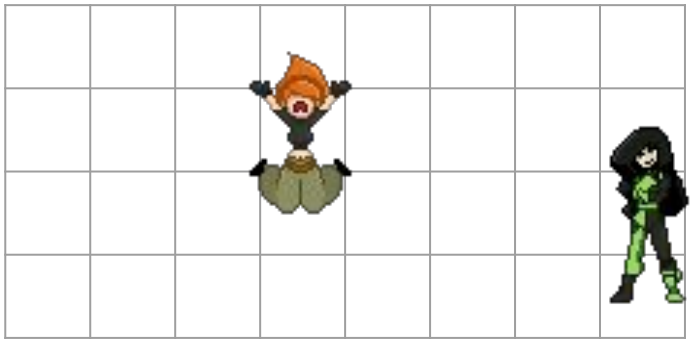
Frame #5 (input coming early frame : **right**)



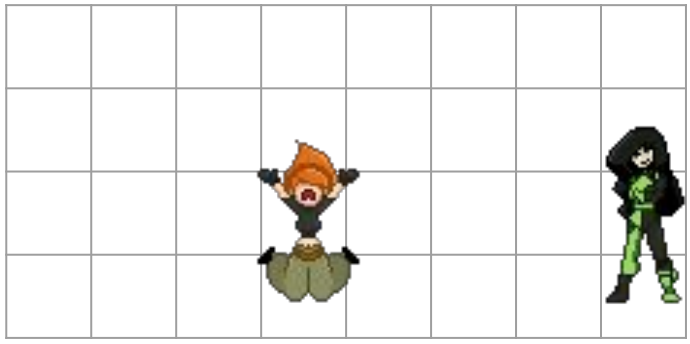
Frame #6 (input coming early frame : **none**)



Frame #7 (input coming early frame : **none**)



Frame #8 (input coming early frame : **right**)



Frame #9 (input coming early frame : **space**)

19 __/10

```
class AudioDB
{
public:
	// Connected to Lua functions Audio.RegisterBeat() & Audio.UnregisterBeat()
	static void RegisterBeat(int channel, LuaRef function);
	static void UnregisterBeat(int channel, LuaRef function);

	// Called every frame automatically.
	static void OnUpdate();

	// Obtain the channels currently playing audio?
	static std::vector<int> GetActiveChannels();

	// Obtain information about a particular audio channel.
	static ChannelInfo GetChannelInfo (int channel);

	// Obtain information about a particular audio file.
	static AudioFileInfo GetFileInfo (const std::string & filename);

	// TODO : Write a data structure below to track your function registrations.
	// Tip : LuaRef has equality operator defined, but is NOT hashable.

	static std::unordered_map<int, std::vector<LuaRef>> registrations;

	// TODO : Write a structure below to track the previous beat number per channel.

	static std::unordered_map<int, float> previous_beat_numbers;

};
```

```
struct ChannelInfo
{
	float seconds_since_play = 0; // This increases every frame.
	std::string filename;
};
```

```
struct AudioFileInfo
{
	std::string filename;
	float seconds_per_beat; // time between beats (1.0 / beats-per-second)
};
```

# Engine Architecture and Lifecycle Functions

```
class EngineUtils
{
public:
      // Use to get the "self" reference of a function (the component it's on).
      // NOTE : You may assume all registered functions are inside a component.
      // NOTE : You may assume all registered functions are member functions.

      static LuaRef GetCompOfFunction(LuaRef function);
};
```

```
// This may be called from Lua via the Audio table.
// Use data structures added within AudioDB (Reminder : this function is static).
// Assume no function will be registered for the same channel multiple times
void AudioDB::RegisterBeat(int channel, LuaRef function)
{
      registrations[channel].push_back(function);
      if registrations[channel].size() == 1 {
            previous_beats[channel] = 0;    // Lacking this did not cause penalty
      }
}
```

```
// This may be called from Lua via the Audio table.
// Use data structures added within AudioDB (Reminder : this function is static).
void AudioDB::UnregisterBeat(int channel, LuaRef function)
{
      int index_to_remove = -1;

      int i = 0;
      for(LuaRef & f : registrations[channel])
      {
        if (f == function)
        {
            index_to_remove = i;
            break;
        }
        i++;
      }

      // Index-based for loops did not need collect-then-alter.
      // My for loop above is range-based, so collect-then-alter is necessary.
      if (index_to_remove != -1)
      {
         registrations[channel].erase(registrations[channel].begin() + index_to_remove);
      }



}
```

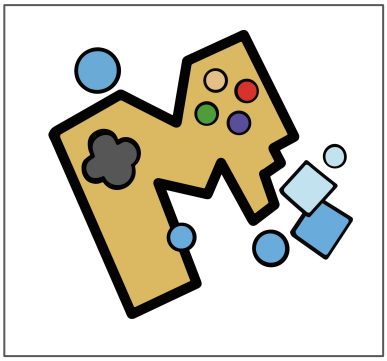
```
// This function gets called automatically every frame.
// Reminder : Calling a lua function on a component requires a "self-reference"
of // the component to be "sent-in" (check Glossary for example).
// NOTE : Do not worry about the state of the actor or component (enabled, etc).
// Reminder : This function is static.
void AudioDB::OnUpdate()
{
      std::vector<int> active_channel = GetActiveChannels();

      for(int & c : active_channels)
      {
            /* Obtain some information about this channel */
            ChannelInfo c_info = GetChannelInfo(c);
            AudioFileInfo f_info = GetfileInfo(c_info.filename)

            float t = c_info.seconds_since_play;

            /* Decide if we have reached a new beat in the song or not. */
            int beat_number = static_cast<int>(t / f_info.seconds_per_beat);

            /* If so, invoke all functions registered on this channel. */
            if (beat_number > previous_beat_numbers[c])
            {
                  for (LuaRef & function : registrations[c])
                  {
                        LuaRef comp = EngineUtils::GetCompOfFunction(function);

                        /* Gotta prevent user's Lua code from crashing us */
                        try {
                            function(comp);

                        } catch(const LuaException & e){}
                  }


                  // And update our previous beat number.
                  previous_beat_numbers[c] = beat_number;
            }
      }




}
```

20

__/1

**Name your new engine and draw its logo ->**
**(you need not keep the name “A2 Engine”)**

A2 Engine




Prepared by Maylen Meguri, (Donna, May, and Mags’ artist) specifically for the students of 498