

**Assignment 1**  
**Deadline: Friday 20th of October**  
**Individual Assignment**

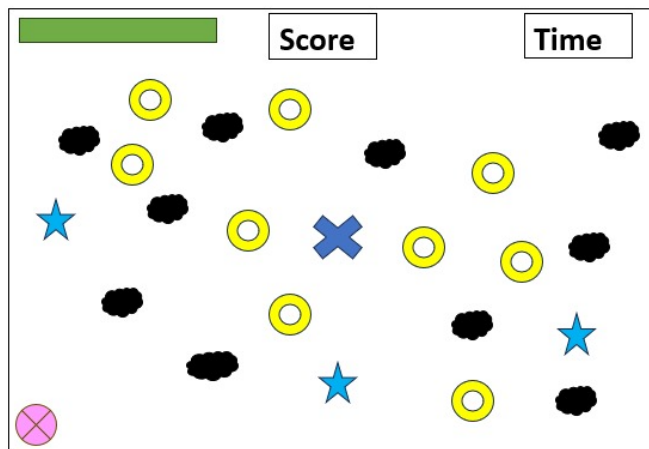
**Description & Game Play:**

**General idea:**

In this assignment, you are required to implement a 2D top-view game of **your own design**. The main idea of the game is that there are **five main models**; the player, the final goal, obstacles that the player should dodge and collectables and power ups that the player can collect.

The player is allowed to move freely in the scene without going outside the game borders or hitting any of the game obstacles. The collectables and power ups are placed **randomly** in the scene and the player should try to **acquire** them.

The player should have **five** lives displayed at the top of the screen as a health bar. In addition, the game score and remaining game time is displayed as well. Player's lives are lost on every collision with a game boundary/obstacles. The game ends either if the player loses all lives or the game **timeout**. The game score increases upon collecting collectables in the scene.



	Player
	Health bar
	collectables
	Power ups
	Obstacles
	Target/ Goal

## The requirements:

- **The environment :**
  - The player is positioned in the middle of the screen at the start of the game,
  - Game obstacles are positioned around in the scene,
  - Collectables and power ups are positioned **randomly for each game run**,
  - Player's health bar, game score and game time are displayed on the screen.
- **The player:**
  - The player starts at a fixed position upon game start,
  - **Controlled** by the keyboard keys to move in **four** directions,
  - The player **rotates** towards the direction of motion.
- **The collectables:**
  - Collectables are placed **randomly for each game run** in the scene,
  - The collectables **disappear** upon collision,
  - Collecting collectables **increases** the game score,
  - The scene should have at least **FIVE** collectable objects.
- **The power-ups:**
  - You are free to choose the functionality of the power up, however you should have **two** different power ups functionality and at least **one** power-up object of each type,
  - Power-ups always appear randomly at various positions and remain fixed in position until acquired by the player, in this case the power up should then disappear,
  - The effect of the power-up should last for a few seconds then **deactivates**,
  - The player should be able to miss the power up, for example, if the player passes next to a power up but does not collide with it, the player does not acquire the power up.
  - The power-ups should not affect the health bar.
- **Player Lives:**
  - The player loses one of the **five** lives whenever collides with any of the obstacles or collides with the game boundary.
  - A health bar should display the player's health and get updated when losing lives.
- **Game End:**
  - Upon game end, an end game screen is displayed to replace the gamescene. A game win or lose is displayed on the screen according to the game state.
- **Sound (BONUS):**
  - Background music that starts playing when the game starts,
  - A sound effect plays when collecting objects from the scene,

- A **different** sound effect plays when colliding with obstacles in the scene,
- A **different** sound effect upon game win,
- A **different** sound effect upon game loss.
- **Texture (BONUS):**
  - Apply textures to everything in the scene except for the small objects where the textures will not be visible.

**NOTE:** The bonus is acquired by **EITHER** sound **OR** texture

**Modeling:** Primitive TYPES include shapes as: point, line, line strip, triangle, quad, and polygon.

### **Submission Guidelines:**

- The assignment should be implemented in **OpenGL**
- This is an **INDIVIDUAL** assignment.
- This assignment is worth 7.5%
- **Follow** assignment 1 checklist to ensure delivering all requirements
- Deadline for the assignment: **Friday 20th of October at 11:59 pm.**
- Cheating cases will lead to a **ZERO**, this **includes**:
  - Copying the code from the **Internet** or **github repo** or a **colleague**.
- Submission guidelines:
  - You should name your .cpp file in the following format PXX-52-XXXX,
  - Submission form: [submit here](#)