

Department of Computer Science, Western University
CS 4482A – Game Programming

APP 2 – Build a Basic Game

1 Introduction

The year is 2016 and you are one of the greatest game programmers of all time. The CEO of your burgeoning company comes to you late on a Friday after noon (as it is always) and tells you that we have to have the prototype of the newest hotness ready to go by Monday or you'll lose your window for greatness.

Your company's team of crack designers has given you a few solid designs to work with. They've given you the choice of which to go with so don't let them down!

Depending on which engine you choose to work with you will have to make concessions, or push it to its limits to really impress these hardcore users.

- **Option 1:** *A 2D Adventure*
- **Option 2:** *Dart Tag: Not Just for Kids*
- **Option 3:** *The Maze Runner: Not a Ripoff*

For any non-code resources, you are free to use anything you find online. You may use any standard assets that come with your game engine or any other resources you find on the internet (just make sure you give credit to whomever built the models you are using). For example, you may find an animated 3D character online. But, you need to write the code which processes user input and triggers the appropriate animation.

Hints

- Choose a game design wisely. It may provide invaluable experience toward building your App 3.
- If the game you decide to make for App 3 has a generous amount of overlap with your App 2 design (e.g. both are 2D platformers), then I will expect *significantly more* mechanics in your App 3 than someone who is switching designs (e.g. 2D platformer for App 2 and 3D combat for App 3).

2 Requirements

Option 1: A 2D Adventure

This option requires you to create a sprite-based game. You have some flexibility here. You can create a game with a top-down view (think The Legend of Zelda, The Binding of Isaac), or a side-scroller (think Super Mario Bros. or Sonic the Hedgehog). You may also create a puzzle-based game or a combat-based game. In either case the goal is simple, get your character from point A to point B.

- **Common Requirements:**

- A single humanoid player, animated and represented by 2D sprites.
- A side-scrolling or top-down camera perspective. Your choice.
- Your game must have at least one level.
- There must be a designated start point and “goal”/end point.
- Player must not be able to pass through floor, walls, obstacles.
- Player must respawn at the start point on death.
- There must be at least one environmental hazard which kills the player (e.g. water, lava, spike pit).
- The game must have a UI to keep track of time to complete the level(s).
 - * Each score will be added to a leaderboard on completion (with (at-least) a 3 character name entry)
- The game must end when reaching the goal point.
- When the game is complete it must *gracefully* restart.
- Create a simple title screen.
 - * Start
 - * Leaderboard
 - * Exit
- Create an in-game pause menu accessible with ESC to restart or exit at any time

- **Combat-Based (hack and slash):**

- Player must have at least one close-range attack (e.g. melee or sword).
- Player must have at least one long-range attack (e.g. magic, boomerang).
- Game must include at least two different types of enemies, each with a unique attack style.
- Player must die when touching an enemy or when hit by an enemy attack.

- **Puzzle-Based (puzzle-platformer):**

- Game must include at least two unique puzzle “obstacles”/“mechanics” (e.g. a door locked by a key that must be found).
- A unique platforming move may be considered a puzzle mechanic. (e.g. rope swing, wall jump)

Option 2 *Dart Tag: Not Just for Kids*

- At least 2 players:
 - A human controlled player.
 - At least one computer controlled player.
 - Players must be represented by 3D meshes.
 - Players must display with an animated run-cycle or walk-cycle when they move. It doesn't have to be fancy, but it has to be animated.
- Each player **must fire a projectile** to tag their opponent. A successful strike will switch who is "it"
- Players must not be able to pass through walls or each other.
- You must indicate which player is currently "it".
- You have to let the human player know if he is currently "it" or not. You must do this in an obvious way.
- You may choose for the tag match to take place either indoors or outdoors.
- Keep a score for each player, and end the game when the timer reaches 3 minutes.
 - Each score will be added to a leaderboard on completion
- When the game is complete it must *gracefully* restart.
- Create a simple title screen.
 - Start
 - Leaderboard
 - Exit
- Create an in-game pause menu accessible with ESC to restart or exit at any time

Option 3: Using Unity - *The Maze Runner: Not a Ripoff*

- A single humanoid player
 - A 3rd person camera view that follows the player
 - Player must be represented by 3D meshes
 - Player must display with an animated run-cycle or walk-cycle when they move. It doesn't have to be fancy, but it has to be animated
- Players must not be able to pass through walls.
- The player's goal is to reach the end of the maze.
 - The maze can be constructed of either a 3D model (created in a 3D modeler like Blender) or by using Unity primitives as the walls. (Or using any assets found online).
 - The maze must have an entrance and an exit
 - The maze must have at least 3 doors that are opened by corresponding keys
- The game must have a UI to keep track of time to complete the maze
 - Each score will be added to a leaderboard on completion (with a 3 character name entry)
- The game must end when reaching the exit to the maze
- When the game is complete it must *gracefully* restart.
- Create a simple title screen.
 - Start
 - Leaderboard
 - Exit
- Create an in-game pause menu accessible with ESC to restart or exit at any time