

# Shamballs

## Technical Design Document

### File formats:

- Final build – EXE
- Textures – PNG
- Audio – WAV, MP3
- Level files - LEVEL
- Model files – FBX

### Middleware:

- DirectX v11.1 (Refer to appendix A)
- DirectX Tool Kit v2016.9.15.1 (Refer to appendix B)
- PhysX v3.3.4 (Refer to appendix C)
- Raknet v4.081 (Refer to appendix D)

### Target Platforms:

Windows Computers with Directx 11 support will be required to run the game. Visual Studio 2013 will be used to develop the game, and TortoiseSVN will be used as a repository hosting service.

### Risk Register:

Challenge(s)	Solution(s)	Impact
Network collisions	Change game mechanics	High
Memory leaks	Use memory leak detection	Medium
Poor performance	Not rendering not visible elements	High
Poor network performance	Send relevant information to server	High

### Data Structures:

- Stack – The different game menu interfaces will be handled using this.
- Oct Tree - To cull large areas of a 3D to improve performance

## Development Methodologies:

Due to having a small amount of time (4 weeks) to complete this project, and having large scope, we will be using MoSCoW to prioritise our features/user stories. We will use this prioritization technique to ensure we deliver a minimum viable product (Waters, 2009).

The methodology that we will use to undertake this project is SCRUM. We will be using SCRUM due to its flexibility, and adaptability (Rising and Janoff, 2000). The SCRUM board will be managed on trello.com. This will allow for all team members to track progress.

## Acceptance Test Plan:

To see UML class diagram, refer to appendix E.

1. Can a player win?
2. Can we identify the player?
3. Can we identify other players?
4. Do the set controls move the player appropriately?
5. Can players collide with each other?
6. Can players collide with the track and walls?
7. Can a player pick up a power-up?
8. Can a player use the power-up?
9. Can the menu be opened?
10. Can the menu be closed?
11. Can the player use the in-game menu to return to main menu?
12. Does the player get reset onto the track if they come off the track?
13. Does the countdown timer ensure no movement until it reaches 0?
14. Can a player connect to the server to play with others?
15. Can a map be created?
16. Can created maps be played?
17. Can the game be restarted?

## Format & Naming Schema:

- Camel Case
  - Class members' start with "m\_"
  - Static members' start with "sm\_"
  - Enums' start with "e\_"
- Static members will be all uppercase e.g. "MAXSPEED"

## Debug Features:

- Display current frame
- Display FPS
- View all player info
  - Location
  - Player's active state – Alive, dead, or spectating
  - Current power-up
  - Velocity
  - ID
  - Ping
- View level entities
  - Location
  - Type
- View player controller type (Keyboard or Gamepad)

## Cheats:

- Noclip
- Force win
- Give power-up
- Double speed

## Team Sign-off:

Sean:

*SChambers*

Marc:



Mathew:

*MBielby*

## Appendix

### **Appendix A:**

DirectX v11.1

<https://blogs.msdn.microsoft.com/chuckw/2015/01/06/direct3d-win32-game-visual-studio-template/>

### **Appendix B:**

DirectX Tool Kit v2016.9.15.1

<https://github.com/Microsoft/DirectXTK/wiki>

### **Appendix C:**

PhysX v3.3.4

<http://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/Index.html>

### **Appendix D:**

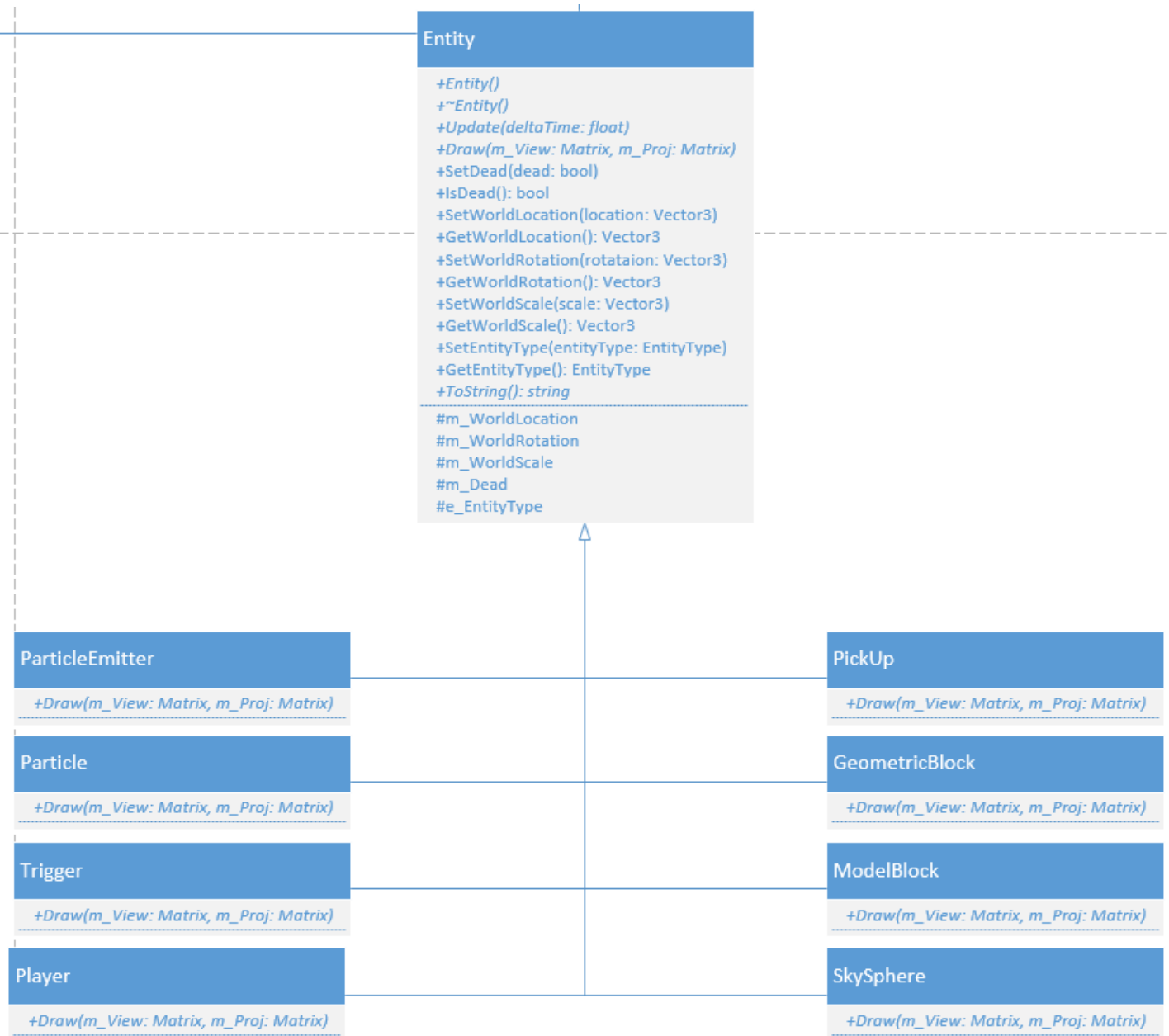
Raknet v4.081

<http://www.jenkinssoftware.com/>

Appendix E:



UML Cont...



## References:

Rising, L. and Janoff, N. (2000). The Scrum software development process for small teams. *IEEE Software*, 17(4), pp.26-32.

Waters, K. (2009). *Prioritization using MoSCoW*. [online] [www.allaboutagile.com](http://www.allaboutagile.com). Available at: [https://cs.anu.edu.au/courses/comp3120/local\\_docs/readings/Prioritization\\_using\\_MoSCoW\\_AllAboutAgile.pdf](https://cs.anu.edu.au/courses/comp3120/local_docs/readings/Prioritization_using_MoSCoW_AllAboutAgile.pdf) [Accessed 24 Sep. 2016].