Chapter 1: Introduction to Design Patterns

```
Enemy
- currentHealth: int
- maxHealth: int
+ GetHealth(): void
+ SetHealth(int): void

FlyingEnemy
- flySpeed: int
+ Fly(): void

Enemy
- currentHealth: int
- maxHealth: int
+ GetHealth(): void
+ SetHealth(int): void
```
Graphic Bundle
Thank you for downloading Visual Studio
If your download does not begin click here to retry.

Get started
Find tutorials, samples, and documentation to help you start coding.

Watch videos
Watch short deep-dive videos to learn about Visual Studio features.

Collaborate now
Open your own Team Services account to plan projects, share source code, and build solutions with your team.
Choose your installation location
C:\Program Files (x86)\Microsoft Visual Studio 14.0
Setup requires up to 8 GB across all drives.

Choose the type of installation
- Default
  Includes C#/VB, Web and Desktop features
- Custom
  Allows you to customize features for your installation

You can add or remove additional features at any time after setup via Programs and Features in the Control Panel.

By clicking the "Next" button, I acknowledge that I accept the License Terms and Privacy Statement.
Graphic Bundle

Visual Studio

Community 2015
with Updates

Select features

- Programming Languages
  - Visual C++
  - Visual F#
  - Python Tools for Visual Studio (January 2017)
- Windows and Web Development
  - ClickOnce Publishing Tools
  - Microsoft SQL Server Data Tools
  - Microsoft Web Developer Tools
  - Silverlight Development Kit
  - Universal Windows App Development Tools
  - Windows 8.1 and Windows Phone 8.0/8.1 Tools

Select All

Setup requires up to 10 GB across all drives.

Back

Next

[7]
Graphic Bundle

Security Warning for EngineTest

You should only open projects from a trustworthy source.

The project file EngineTest may have come from a location that is not fully trusted. It could present a security risk by executing custom build steps when opened in Microsoft Visual Studio that could cause damage to your computer or compromise your private information.

Would you like to open this project?

[ ] Ask me for every project in this solution

[OK] [Cancel]
Chapter 2: One Instance to Rule Them All - Singletons

This is a demo of the different things you can do in the Mach 5 Engine. Play with the demo but you must also inspect the code and comments.

If you find errors, report to lazersquad@gmail.com

Value of enemyCount: 10
Value of enemyCount: 20
This is a demo of the different things you can do in the Mach 5 Engine. Play with the demo but you must also inspect the code and comments.

If you find errors, report to lazersquad@gmail.com

A I can be called anywhere! classVariable value: 2.500000
B I can be called anywhere! classVariable value: 5.000000

Singleton

- instance: Singleton *

+ GetInstance(): Singleton *
This is a demo of the different things you can do in the Mach 5 Engine. Play with the demo but you must also inspect the code and comments.

If you find errors, report to lazersquad@gmail.com

Constructor!
Value: 5
Constructor!
Value: 2.5
Constructor!
Value: Testing
Deconstructor!
Deconstructor!
Deconstructor!
Chapter 3:
Creating Flexibility with the Component Object Model
Program to an interface, not an implementation

Object (Client) Uses

FlightAI (Interface)
  +Update(float)

Chase
  +Update(float)

Intercept
  +Update(float)

Patrol
  +Update(float)

AttackAI (Interface)
  +Attack()

Open for extension, closed for modification.

Bullet
  +Attack()
  +ShouldAttack()

Bomb
  +Attack()
  +ShouldAttack()

Missile
  +Attack()
  +ShouldAttack()
m_comp->Describe();
//Add more description here
Chapter 4:
Artificial Intelligence Using the State Pattern
This is a demo of the different things you can do in the Mach 5 Engine. Play with the demo but you must also inspect the code and comments.

If you find errors, report to lazersquad@gmail.com

Idle: Enter
Idle: Exit
Follow: Enter
Follow: Exit
Idle: Enter
Chapter 5: Decoupling Code via the Factory Method Pattern
Stage (abstraction)

StageFactory (abstraction)
+Stage* Build() 

Level2Factory
+Stage* Build() 

return new Level2;

StageManager
-m_StageFactory

StageFactory
- m Builders: map 
+ Build(StageTypes type): Stage*

StageBuilder
«interface»
+Build(): Stage*

return new Level1;

Level1Builder
+Build(): Stage*

MainMenuBuilder
+Build(): Stage*

OptionsMenuBuilder
+Build(): Stage*
Chapter 6: Creating Objects with the Prototype Pattern
Graphic Bundle

Client
+Shape* Operation()

return someShape->Clone();

Shape (Interface)
+Shape* Clone(void)

Circle
+Shape* Clone(void)

return new Circle(*this);

Square
+Shape* Clone(void)

return new Square(*this);

Triangle
+Shape* Clone(void)

return new Triangle(*this);
Chapter 7: Improving Performance with Object Pools
Chapter 8: Controlling the UI via the Command Pattern
Graphic Bundle

```
5 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 4 3 3 3 1 0 0 3 4
Press any key to continue . . .
```
Graphic Bundle

Client

Command (abstraction)
+Execute()

Square5Command
+Execute()

OneArgCommand
+Execute()

TMethodCommand
+Execute()

Client Doesn't know which function is called

Hard-coded Function

Use Function Pointer

Uses Pointer to class Method

Astro Shot

[ Play ]
[ Options ]
[ Quit ]
Chapter 9: Decoupling Gameplay via the Observer Pattern
Chapter 10:
Sharing Objects with the Flyweight Pattern
Chapter 11: Understanding Graphics and Animation
800 Pixels Wide

800 Pixels High

Options

32\%\text{Infowed}  \quad \text{Full Screen}

800 \times 600  \quad 1024 \times 768  \quad 1280 \times 768

Back
<table>
<thead>
<tr>
<th>Time (in Seconds)</th>
<th>Back Buffer</th>
<th>Front Buffer/Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>Frame 1 (Working)</td>
<td>Blank</td>
</tr>
<tr>
<td>1/60 = 0.0167(Monitor Refresh MISS)</td>
<td>Frame 1 (Working)</td>
<td>Blank</td>
</tr>
<tr>
<td>1/50 = 0.02(Game Update)</td>
<td>Frame 1 (Done)</td>
<td>Blank</td>
</tr>
<tr>
<td>2/60 = 0.0333(Monitor Refresh OK)</td>
<td>Frame 2 (Working)</td>
<td>Frame 1</td>
</tr>
<tr>
<td>3/60 = 0.0500(Monitor Refresh MISS)</td>
<td>Frame 2 (Working)</td>
<td>Frame 1</td>
</tr>
<tr>
<td>2/60 + 1/50 = 0.0534(Game Update)</td>
<td>Frame 2 (Done)</td>
<td>Frame 1</td>
</tr>
<tr>
<td>4/60 = 0.0666(Monitor Refresh OK)</td>
<td>Frame 3 (Working)</td>
<td>Frame 2</td>
</tr>
<tr>
<td>5/60 = 0.0833(Monitor Refresh MISS)</td>
<td>Frame 3 (Working)</td>
<td>Frame 2</td>
</tr>
<tr>
<td>4/60 + 1/50 = 0.0888(Game Update)</td>
<td>Frame 3 (Done)</td>
<td>Frame 2</td>
</tr>
<tr>
<td>6/60 = 0.1000(Monitor Refresh OK)</td>
<td>Frame 4 (Working)</td>
<td>Frame 3</td>
</tr>
<tr>
<td>Time (in Seconds)</td>
<td>Tertiary Buffer</td>
<td>Back Buffer</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>0.0</td>
<td>Idle</td>
<td>Frame 1 (Working)</td>
</tr>
<tr>
<td>1/60 = 0.0167(Monitor Refresh MISS)</td>
<td>Idle</td>
<td>Frame 1 (Working)</td>
</tr>
<tr>
<td>1/50 = 0.02(Game Update)</td>
<td>Frame 2 (Working)</td>
<td>Frame 1 (Done)</td>
</tr>
<tr>
<td>2/60 = 0.0333(Monitor Refresh OK)</td>
<td>Frame 2 (Working)</td>
<td>Idle</td>
</tr>
<tr>
<td>2/50 = 0.04 (Game Update)</td>
<td>Frame 3 (Working)</td>
<td>Frame 2 (Done)</td>
</tr>
<tr>
<td>3/60 = 0.05(Monitor Refresh OK)</td>
<td>Frame 3 (Working)</td>
<td>Idle</td>
</tr>
<tr>
<td>3/50 = 0.06(Game Update)</td>
<td>Frame 4 (Working)</td>
<td>Frame 3 (Done)</td>
</tr>
<tr>
<td>4/60 = 0.0667(Monitor Refresh OK)</td>
<td>Frame 4 (Working)</td>
<td>Idle</td>
</tr>
<tr>
<td>4/50 = 0.08(Game Update)</td>
<td>Frame 5 (Working)</td>
<td>Frame 4 (Done)</td>
</tr>
<tr>
<td>5/60 = 0.0833(Monitor Refresh OK)</td>
<td>Frame 5 (Working)</td>
<td>Idle</td>
</tr>
<tr>
<td>5/50 = 0.10(Game Update)</td>
<td>Frame 6 (Working)</td>
<td>Frame 5 (Done)</td>
</tr>
<tr>
<td>6/60 = 0.10(Monitor Refresh OK)</td>
<td>Frame 6 (Working)</td>
<td>Idle</td>
</tr>
<tr>
<td>7/60 = 0.117(Monitor Refresh MISS)</td>
<td>Frame 6 (Working)</td>
<td>Idle</td>
</tr>
<tr>
<td>6/50 = 0.12(Game Update)</td>
<td>Frame 7 (Working)</td>
<td>Frame 6 (Done)</td>
</tr>
<tr>
<td>8/60 = 0.133(Monitor Refresh OK)</td>
<td>Frame 7 (Working)</td>
<td>Idle</td>
</tr>
<tr>
<td>7/50 = 0.14(Game Update)</td>
<td>Frame 8 (Working)</td>
<td>Frame 7 (Done)</td>
</tr>
</tbody>
</table>

\[
\frac{dx}{dt} = v
\]

\[
\frac{x_1 - x_0}{dt} = v
\]

\[
x_1 - x_0 = v \cdot dt
\]

\[
x_1 = x_0 + v \cdot dt
\]
\[ \frac{dv}{dt} = a \]

\[ \frac{v_1 - v_0}{dt} = a \]

\[ v_1 - v_0 = a \cdot dt \]

\[ v_1 = v_0 + a \cdot dt \]

\[ F = ma \]

\[ a = \frac{F}{m} \]

\[ x_1 = x_0 + v \cdot dt \]

\[ v_1 = v_0 + a \cdot dt \]

\[ a = \text{Force/mass} \]

\[ p = \frac{1}{2} at^2 + v_0 t + p_0 \]

\[ \text{distance} = \frac{1}{2} at^2 + 0 + 0 \]

\[ \text{distance} = \frac{1}{2}(10)(10 \cdot 10) \]

\[ \text{distance} = 10 \cdot (100) \]

\[ \text{distance} = 1000 \]
Chapter 12: Best Practices