

Gama Network Presents:

# Gamasutra.com

---

## Developing Your Own Replay System

( )

By *Cyrille Wagner*

Gamasutra

February 4, 2004

URL: [http://www.gamasutra.com/features/20040103/fristrom\\_01.shtml](http://www.gamasutra.com/features/20040103/fristrom_01.shtml)

, AI 가

*Cyclone Circus* PS2  
Playlogic Games Factory

.( 1 )

가

AI

가

가

가  
가  
가

가 PS2

가 가

가 가

,





:

,

가

가?

PS2

300

가 60Hz

가

,

18,000

가

PS2

32MB

1MB

가

, 가

500K

30

가

, 18,000

540,000

가

500K

( )

가

,

,

AI

,

,

가

1K

가

2/3

가

4K 가

10K

가

10K

. 18,000

. PC

200MB

가

가

가

2

가

가

가



2.

가

가

가

가

가

가

가 ,

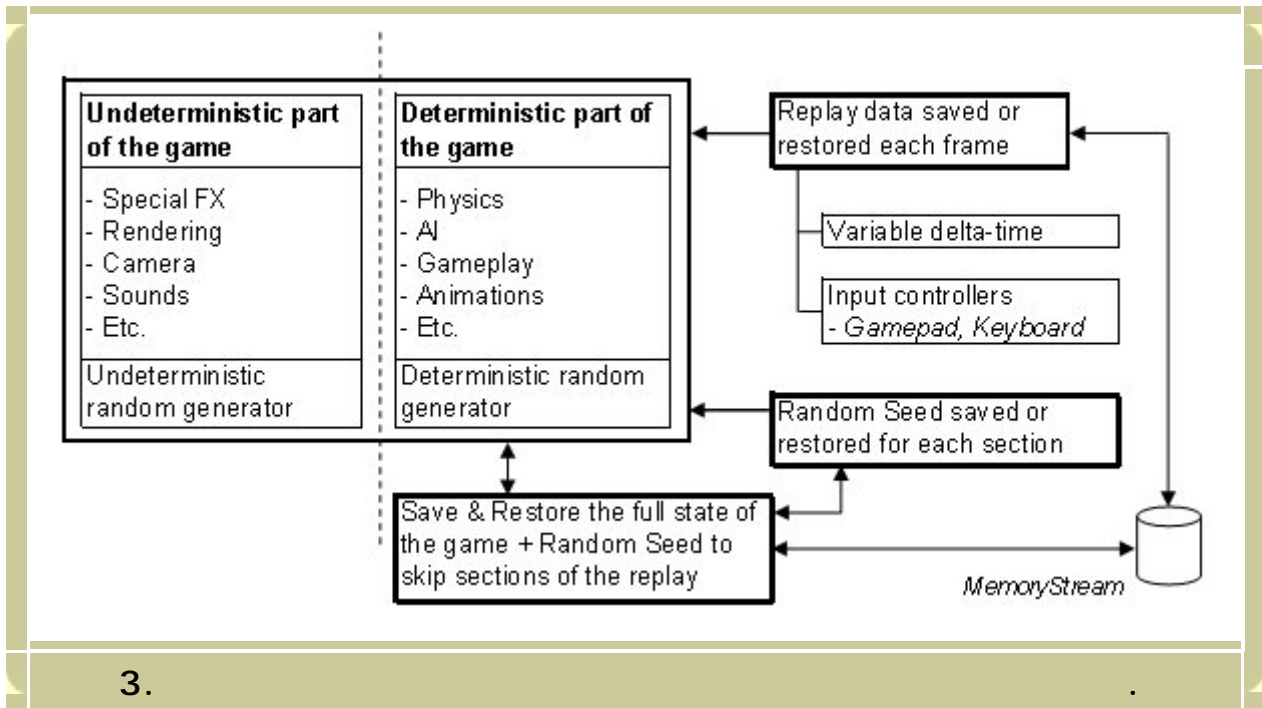
가

가 ,

가

가

( 3 )



3.

가

가

가

1MB

가

가



20ms

40ms

가

가

가

가

MPEG

( 1 ).

1

```

//
//          The
TMemoryStream
class uses a block of
memory instead of a
file
// but implement
IStreamReader and
IStreamWriter
interfaces.
//
class
TMemoryStream :
public
IStreamReader,
public IStreamWriter
{
public:

TMemoryStream();
    virtual
~TMemoryStream();

    // IStreamReader
/ IStreamWriter
    virtual      int
ReadBytes(void *
pBuffer,      int
numBytes);

```





가 . 가 가  
가? 60 가  
가 . 가  
- 가 - 가  
0 255  
3  
[-128, -48, -12, -4, +4, +12, +48, +128]

가 가 4  
가 가 , 가  
BitPacker  
1  
3  
( 2 )  
Write

가

,

.



```

//
// Container to store multiple bit information
in one or more bytes.
//
class TBitPacker
{
public:
    TBitPacker();
    virtual ~TBitPacker();

    // Clear all information to get an empty
container.
    void Clear();

    // Add a boolean information.
    void AddBit(bool bValue);

    // Add an integer stored on a given
number of bits only.
    void AddInteger(int nValue, int
nBitCount);

    // Get the next boolean information and
read the next byte
    // from the given stream if needed.
    bool GetBit(IStreamReader& stream);

    // Get the next integer and read the next
bytes from the
    // given stream if needed.
    int GetInteger(int nBitCount,
IStreamReader& stream);

    // Get the number of bytes needed to

```

## 2. BitPacker

3 가

1

가 . 가  
1 가 .







```

//
// To make the code as simple as possible, we
// only save two boolean and one analog
// values from the input controller. In the article
// however, explanations and memory size are
based
// on a real case with much more saved values.
//
void InputControllerUpdate(RwReal deltaTime)
{
    TMemoryStream*      pMemStream      =
GetReplayStream();

    // Store the previous state of all input values
    unsigned char nPreviousSteer = m_nSteer;

    bool          bPreviousJumpPressed      =
m_bJumpPressed;
    bool          bPreviousOpenWingsPressed      =
m_bOpenWingsPressed;

    // Replay system need to store and restore the
// exact same input controller state
    if (g_bRecord)
    {
        // Update all analog and boolean input
values// based on the input controller
        m_nSteer      =
FloatToUnsignedChar(m_pControllerBinding -
>Steer.GetValue());

        m_bJumpPressed      =
m_pControllerBinding - >Jump.BecamePressed();
        m_bOpenWingsPressed      =
m_pControllerBinding -

```

3.

가?

가

가

가

가

SaveState

REPLAY\_ASSERT

가

가

가

가 , 가 가  
가 가  
가 가  
가 4 AssertState  
가 가  
가 가



```

// This method saves the state of an object to
the memory stream.
void          SaveState(TMemoryStream&
MemStream)
{
    MemStream.WriteValue(m_vLastPosition);
MemStream.WriteValue(m_fLastProgression);
MemStream.WriteValue(m_fHeightInTrack);
    // Save AiVehicle
    m_pAiVehicle->SaveState(MemStream);}
// This method restores the state of an object
from the memory stream.
void          RestoreState(TMemoryStream&
MemStream)
{    MemStream.ReadValue(m_vLastPosition);
MemStream.ReadValue(m_fLastProgression);
MemStream.ReadValue(m_fHeightInTrack);
    // Restore AiVehicle
    m_pAiVehicle-
>RestoreState(MemStream);
}
// This method assert the state of an object is
the same as the one
// described in the memory stream and log
differences.
void          AssertState(TMemoryStream&
MemStream)
{    MemStream.AssertValue(m_vLastPosition);
MemStream.AssertValue(m_fLastProgression);
MemStream.AssertValue(m_fHeightInTrack);
    // Restore AiVehicle
    m_pAiVehicle->AssertState(MemStream);
}

```

4.

가 가

가

가

가

가

PS2

2 가

2 가

PS2

150K

가

[Kharkar01]

[Kharkar01] Sandeed V.Kharkar, "Camera AI for Replays", *AI Game Programming Wisdom*, pp.472-478, Charles River Media, 2002.

*Copyright 2003 CMP Media Inc. All rights reserved.*